

THE CHEMMACROS BUNDLE

V4.3 2014/01/24

packages **CHEMMACROS** (v4.3), **CHEMFORMULA** (v4.2a), **GHSYSTEM** (v4.2e) and
CHEMGREEK (v0.3)

documentation for the **CHEMMACROS** package

Clemens NIEDERBERGER

<https://bitbucket.org/cgnieder/chemmacros/>

contact@mychemistry.eu

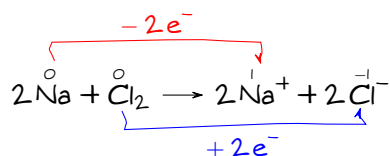


Table of Contents

I. Preliminaries	2	6.2. Supported Languages	7
1. Licence, Requirements and README	2	6.3. Specialties	9
2. Motivation and Background	3	6.3.1. German	9
3. News	4	6.3.2. Italian	9
3.1. Version 4.0	4	II. CHEMMACROS	9
3.2. Version 4.2	5	7. Particles, Ions and Symbols	9
3.3. Version 4.3	5	7.1. Predefined	9
4. Package Options	5	7.2. Own Particles	11
5. Setup	6	8. Nomenclature, Stereo Descriptors, Latin Phrases	12
6. Language Settings	7	8.1. IUPAC Names	12
6.1. How it Works	7	8.1.1. Predefined Commands	14
		8.1.2. Own Naming Commands	18
		8.2. Latin Phrases	18

9. Units for the Usage With siunitx	19	15.5. An Example	37
10. Acid/Base	20	15.5.1. Nearly Standard	38
		15.5.2. Formatted List	38
		15.5.3. Crazy	38
11. Oxidation Numbers, Real and Formal Charges	21	16. Reaction Environments	39
11.1. Ion Charges	22	16.1. Defined by CHEMMACROS . .	39
11.2. Oxidation Numbers	22	16.2. Own Reactions	42
11.3. Partial Charges and Similar Stuff	24	16.3. List of Reactions	43
12. Reaction Mechanisms	25	17. Phases	45
13. Redox Reactions	26	17.1. Basics	45
		17.2. Define Own Phases	45
		17.3. Language Dependencies . . .	46
14. (Standard) State, Thermodynamics	29	18. Newman Projections	47
14.1. Thermodynamic Variables . .	29	19. s, p, and Hybrid Orbitals	49
14.1.1. Create New Variables or Redefine Existing Ones	30	III. Other Packages of the Bundle	51
14.2. State	32	IV. Appendix	52
15. Spectroscopy and Experimental Data	33	Suggestions and Bug Reports	52
15.1. The \NMR Command	33	References	52
15.2. Short Cuts	34	Index	55
15.3. An Environment to Typeset Experimental Data	34		
15.4. Customization	35		

Part I.

Preliminaries

1. Licence, Requirements and README

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

The CHEMMACROS bundle needs the bundles l3kernel [The13a] and l3packages [The13b]. It

also needs the packages `siunitx`¹ [Wri13], `mathtools`² [MRW13], `bm`³ [CMo4], `nicefrac`⁴ [Rei98] and `environ`⁵ [Rob13] as well as `tikz`⁶ [Tan10] and the TikZ libraries `calc` and `arrows`. Language support is done with the help of the translations⁷ [Nie13]. The `CHEMMACROS` package also loads the other packages of this bundle.

The package option `xspace` also loads the package `xspace` [CHo9].

The `CHEMMACROS` bundle bundles four packages: `CHEMMACROS`, `CHEMFORMULA`, `GHSYSTEM` and `CHEMGREEK`. The package dependencies of the other packages are described in the respective manuals.

2. Motivation and Background

`CHEMMACROS` started some years ago as a growing list of custom macros that I frequently used. I cannot completely recall when and why I decided to release them as a package. Well – here we go and you might find it useful, too, I hope.

Both the macros and their functionality have changed over time and quite a lot have been added. Many things have been unified and what’s probably most important: many possibilities to customize have been added, too.

Probably every chemist using \LaTeX 2_ε is aware of the great `mhchem` package by Martin Hensel. There have always been some difficulties intertwining it with `CHEMMACROS`, though. Also, some other minor points in `mhchem` always bothered me, but they hardly seemed enough for a new package. They weren’t even enough for a feature request to the `mhchem` author. The challenge and the fun of creating a new package and the wish for a highly customizable alternative led to `CHEMFORMULA` after all.

`CHEMFORMULA` works very similar to `mhchem` but is more strict as to how compounds, stoichiometric factors and arrows are input. In the same time `CHEMFORMULA` offers possibilities to customize the output that `mhchem` does not. Although `CHEMFORMULA` is meant as an *alternative* to `mhchem` `CHEMMACROS` only loads `CHEMFORMULA` and uses it at various places internally, too.

As a chemist you are probably aware of the fact that the UNITED NATIONS have developed the GLOBALLY HARMONIZED SYSTEM OF CLASSIFICATION AND LABELLING OF CHEMICALS (GHS) as a global replacement for the various different systems in different countries. While it has not been implemented by all countries yet [Uni12], it is only a matter of time.

The package `GHSYSTEM` enables you to typeset all the hazard and precautionary statements and pictograms in a very easy way. The statements are taken from EU regulation 1272/2008 [Theo8].

There are four points I hope I have achieved with this bundle:

- intuitive usage as far as the syntax of the commands is concerned

-
1. on CTAN as `siunitx`: <http://mirrors.ctan.org/macros/latex/contrib/siunitx/>
 2. on CTAN as `mathtools`: <http://mirrors.ctan.org/macros/latex/contrib/mathtools/>
 3. on CTAN as `bm`: <http://mirrors.ctan.org/macros/latex/contrib/bm/>
 4. on CTAN as `nicefrac`: <http://mirrors.ctan.org/macros/latex/contrib/nicefrac/>
 5. on CTAN as `environ`: <http://mirrors.ctan.org/macros/latex/contrib/environ/>
 6. on CTAN as `pgf`: <http://mirrors.ctan.org/graphics/pgf/>
 7. on CTAN as `translations`: <http://mirrors.ctan.org/macros/latex/contrib/translations/>

3. News

- the commands shall not only make typesetting easier and faster but also the document source more readable with respect to semantics (`\ortho`-dichlorobenzene is easier to read and understand than `\textit{o}`-dichlorobenzene)
- as much customizability as I could think of so every user can adapt the commands to his or her own wishes
- default settings compliant with the recommendations of the INTERNATIONAL UNION OF PURE AND APPLIED CHEMISTRY (IUPAC).

Especially the last point needed some pushing from users to get things right in many places. If you find anything not compliant with IUPAC recommendations⁸ I would welcome an email very much!

3. News

3.1. Version 4.0

With version 4.0 some changes have been made:

- first of all the packages `CHEMFORMULA` and `GHSYSTEM` do not load `CHEMMACROS` any more which means they can be used independently.
- the option `bpchem` has been dropped.
- the commands `\mch` and `\pch` now match `CHEMFORMULA`'s charges.
- the option `method` has been dropped.
- the option `append` has deprecated.
- the option `greek` has been extended to support other uppercase greek letters, for example those provided by `kpfonts`. This is handled internally by the new package in the family: `CHEMGREEK`. This package is not really a package for usage at a user-level but could in principle be used to extend the `greek` option.
- language support is now done with the help of the translations. This means that with version 4.0 the document language is recognized automatically.
- the status of the commands `\Lfi` and `\Dfi` has been changed from *deprecated* to *dropped*.
- various other changes like bug fixes and improvements on the typographical appearance of `CHEMFORMULA`'s inline formulae with `\ch`.

8. This does not concern the `\ox` command. The IUPAC version is `\ox*`.

3.2. Version 4.2

- Changed particles with electron pairs such as `\ba` to use `CHEMFORMULA`'s new macro `\chlewis` for the Lewis electrons.
- Changed the implicit `\Delta` in the thermodynamic state variables into `\ChemDelta` to ensure that an upright symbol is used.
- Change in the syntax of `\DeclareChemState` and `\RenewChemState`. The old syntax is still supported but discouraged.

3.3. Version 4.3

- All one-letter IUPAC macros have been exchanged in favour of more meaningful macro names. The one-letter commands still exist for backward compatibility (and to some users no doubt also for convenience). They are no longer recommended though. One-letter commands seldomly have meaningful names and often they've also been defined by other packages. This means they make collaboration more difficult than it needs to be and are a source for package conflicts. `CHEMMACROS` used to solved the latter problem by only providing them inside the argument of `\iupac`. The one exception `CHEMMACROS` makes is the command `\p` (for things like pH) which is and will remain an official command.
- The environment `experimental` has got a number of new options, see section 15.4.
- The commands `\DeclareChem{...}` now don't give an error any more if the command already exists. This is more consistent with L^AT_EX's `\DeclareRobustCommand`. For all those commands a version `\NewChem{...}` is introduced that *does* give an error if the new command is already defined.
- The package option `strict` has been deprecated.
- The package option `cmversion` has been deprecated.
- The command `\mhName` has been dropped.

4. Package Options

`CHEMMACROS` has several package options. They all are used as key/value pairs like

```
\usepackage[option1 = <value1>, option2 = <value2>]{chemmacros}
```

Some also can be used without value (`\usepackage[option1]{chemmacros}`), which means that the underlined value is used.

Both `CHEMFORMULA` and `GHSYSTEM` don't have package options of their own. If you load them explicitly any given option will silently fail. Options can then only be set using the `setup` command.

5. Setup

- option** » `circled = formal|all|none` Default: `formal`
`CHEMMACROS` uses two different kinds of charges which indicate the usage of real (+/−) and formal (⊕/⊖) charges. The option `formal` distinguishes between them, option `none` displays them all without circle, option `all` circles all.
- option** » `circletype = chem|math` Default: `chem`
This option switches between two kinds of circled charge symbols: `\fplus` ⊕ and `\oplus` ⊕.
- option** » `ghsystem = true|false` Default: `true`
`ghsystem = {false}` disables the automatic loading of the `GHSYSTEM` package.
- option** » `greek = auto|upgreek|textgreek|mathdesign|kpfonts|newtx|fourier|textalpha` Default: `auto`
This option determines how the letters `\chemalpha` and friends are typeset. See pages 11 and 14 for more information. Please note that this option *does not load either upgreek, kpfonts or any other package!* It only determines which one to choose if available. The option `auto` will detect if any of the packages needed for one of the options has been loaded and use it if available. If more than one of the packages has been loaded the option will choose the one listed first in the above choice list. If you explicitly choose an option other than `auto` or `math` you also have to load the corresponding package. *This option can only be chosen in the preamble.*
- option** » `iupac = auto|restricted|strict` Default: `auto`
Take care of how IUPAC naming commands are defined, see page 13.
- option** » `language = american|british|english|french|german|italian|ngerman` (initially empty)
Load the language used by `CHEMMACROS`. *This option can only be chosen in the preamble.*
- option** » `Nu = chemmacros|mathspec` Default: `chemmacros`
The package `mathspec` also defines a macro `\Nu`. This option chooses which definition holds, see page 10. *This option can only be chosen in the preamble.*
- option** » `synchronize = true|false` Default: `false`
The setting `true` will tell `CHEMMACROS` to adapt the font settings of `CHEMFORMULA`.
- option** » `xspace = true|false` Default: `true`
With this option most commands are defined with a `\xspace`.

5. Setup

Various of `CHEMMACROS`’, `CHEMFORMULA`’s and `GHSYSTEM`’s commands have key/value pairs with which they can be customized. Most times they can be used as (optional) argument of the commands themselves. They also can most times be used with the `\chemsetup` command.

```
\chemsetup[⟨module⟩]{⟨key⟩ = ⟨value⟩}
  Set up the options for module ⟨module⟩ only or

\chemsetup{⟨module⟩/⟨key⟩ = ⟨value⟩}
  in combination with options from other modules.
```

6. Language Settings

The keys each belong to a module, which defines for which commands they are intended for. If a key is presented, you'll see the module to which it belongs in the left margin. You have two ways to use keys with the `\chemsetup`, as you can see above.

The package options can also be seen as keys belonging to the module `option`. This means they can also be used with the `\chemsetup` command (except for the option `version = 1|2|3`).

```
1 \chemsetup[option]{circled=none}
2 \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
3 \chemsetup[option]{circled=formal}
4 \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
5 \chemsetup[option]{circletype=math}
6 \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
7 \chemsetup{option/circletype=chem,option/circled=all}%
8 \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
9 \chemsetup{option/circletype=math}
10 \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt
```

$- + \ominus \oplus e^- p^+$
 $- + \ominus \oplus e^- p^+$
 $- + \ominus \oplus e^- p^+$
 $- + \ominus \oplus e^- p^+$
 $- + \ominus \oplus e^- p^+$

Keys *not* belonging to a module *cannot* be used with `\chemsetup`!

All options of `CHEMFORMULA` belong to the module `chemformula` and all of `GHSYSTEM`'s options belong to the module `ghsystem`.

6. Language Settings

6.1. How it Works

`CHEMMACROS` uses the translations package for a number of language dependent strings. That means that if a suitable translation to those strings is given the babel [Bra13] or polyglossia [Cha13] language will be picked up automatically. You can, however, overwrite this mechanism by explicitly choosing the language you want. This is done with the package option `language`.

Section 6.2 lists all language dependent strings and the provided translations.

6.2. Supported Languages

By choosing the option

```
\chemsetup[<option>]{language=<language>}
```

Selection of the language `<language>`.

6. Language Settings

TABLE 1: Language dependent strings.

translations key	English default
K-acid	A
K-base	B
K-water	W
phase-sld	l
phase-lqd	s
phase-gas	g
phase-aq	aq
list-of-reactions reaction	List of reactions

you can set the language that is used by **CHEMMACROS** if you want it to be a *different language than your main document language*.

There are some language definitions made by **CHEMMACROS**. They include

- the header of the list of reactions,
- the beginning of the entries in the list of reactions, and
- the H- and P-statements of the GHS statements.

CHEMMACROS uses the translations to get translated strings sensitive to babel or polyglossia settings. All pre-defined translations keys are listed in table 1. To some of those a few non-English translations are provided.

Currently this includes the following translations:

```
1 % subscript used in \Ka:
2 \DeclareTranslation{German}{K-acid}{S}
3 % the phases \sld and \lqd:
4 \DeclareTranslation{German}{phase-sld}{f}
5 \DeclareTranslation{German}{phase-lqd}{f}{l}
6 % heading of the list of reactions:
7 \DeclareTranslation{English}{list-of-reactions}{List of reactions}
8 \DeclareTranslation{German}{list-of-reactions}{Reaktionsverzeichnis}
9 \DeclareTranslation{Italian}{list-of-reactions}{Elenco delle reazioni}
10 \DeclareTranslation{French}{list-of-reactions}{Table des r\'eactions}
11 % name at the beginning of each entry in the list of reactions:
12 \DeclareTranslation{English}{reaction}{Reaction}
13 \DeclareTranslation{German}{reaction}{Reaktion}
14 \DeclareTranslation{Italian}{reaction}{Reazione}
15 \DeclareTranslation{French}{reaction}{R\'eaction}
```


All other languages will fall back to English. However, you can always add the translation you want. If you send me an email with translations you'd like to have added to **CHEMMACROS** I'll gladly add them.

6.3. Specialties

6.3.1. German

If you choose german/ngerman the phase commands `\sld` and `\lqd` and the command `\pKa` are translated.

6.3.2. Italian

Choosing the language italian defines two additional IUPAC commands:

`\ter`
ter

`\sin`
sin

Part II.

CHEMMACROS

7. Particles, Ions and Symbols

7.1. Predefined

CHEMMACROS defines some simple macros for displaying often needed particles and symbols. Please note, that they're displayed differently depending on the package options used, see section 4. These commands can be used in text as well as in math mode. Note that they are not meant to be used in **CHEMFORMULA**'s `\ch`.

`\Hpl`
 H^+ (proton)

`\Hyd`
 OH^- (hydroxide)

`\HtO`
 H_3O^+ (oxonium ion) (**H three O**)

`\water`
 H_2O

`\el` e^- (electron)`\prt` p^+ (proton)`\ntr` n^0 (neutron)`\Nu` Nu^- (nucleophile)

The package `mathspec` also defines a macro `\Nu`. If you chose package option `Nu = {mathspec}` `CHEMMACROS` defines `\Nuc` instead.

`\El` E^+ (electrophile)`\ba` ba^- (base)`\fplus` \oplus `\fminus` \ominus `\transitionstatesymbol` \ddagger `\standardstate` \ominus

This symbol is only provided by `CHEMMACROS`, if the package `chemstyle` is not loaded; the idea is borrowed from there.⁹

`\changestate` Δ

A math operator symbol for denoting the change in an extensive thermodynamic quantity for a process such as ΔH^\ominus . This symbol is used in the definitions presented in section 14.

`\chemalpha` α , `\chemAlpha` A

For each of the 24 greek letters a lowercase and uppercase `\Chem...` command is defined that maps to the upright greek letter as set with the option `greek`. More details on this can be found in the manual of the `CHEMGREEK` package.

The two particles `\Nu` and `\ba` can be modified. To do that you use the option

9. many thanks to the package author Joseph Wright.

7. Particles, Ions and Symbols

TABLE 2: Packages needed for the **greek** package option..

option	needed package
auto	—
math	—
textgreek	textgreek [Mic11]
upgreek	upgreek [Scho3]
newtx	newtxmath [Sha13]
kpfonts	kpfonts [Cai10]
mathdesign	mathdesign [Pic13]
fourier	fourier [Bov05]
textalpha	textalpha [Mil13]

particle » **elpair** = false|dots|dash

Default: false

Set how the electron pair of the particles **\Nu** and **\ba** are set.

```

1 \ba[elpair] \Nu[elpair=dash]
2
3 \chemsetup[particle]{elpair}
4 \ba\ \Nu

```

$\text{ba}^{\cdot-}$ $\text{Nu}^{\cdot-}$
 $\text{ba}^{\cdot-}$ $\text{Nu}^{\cdot-}$

The greek letters aren't newly defined symbols but are defined differently depending on the packages you've loaded. The default definition is the corresponding math letter. If you have loaded the textgreek package the letters are taken from there, and if you have loaded the package upgreek the macros of that package are used. This is also described in the description of the package option **greek**, other details can be found in the documentation of the **CHEMGREEK** package. Which package you have to load for a specific choice for the package option **greek** is listed in table 2. This documentation uses newtxmath and the setting **greek** = {newtx} for instance.

The reason why **CHEMMACROS** uses these macros in the first place is IUPAC compliance. IUPAC recommends to use upright greek letters in nomenclature.

Greek letters are used in systematic organic, inorganic, macromolecular and biochemical nomenclature. These should be roman (upright), since they are not symbols for physical quantities.

IUPAC Green Book [Coh+08, p. 9]

CHEMMACROS uses these commands now to define nomenclature commands, see page 14.

7.2. Own Particles

Surely sometimes it can be handy to have other particle macros defined such as **\positron** or **\photon**. This can easily be done with this command:

Introduced in
version 4.3

`\NewChemParticle{⟨cs⟩}{⟨definition⟩}`

Define a new particle command. Gives an error if `⟨cs⟩` already exists.

Changed in
version 4.3

`\DeclareChemParticle{⟨cs⟩}{⟨definition⟩}`

Define a new particle command.

`\RenewChemParticle{⟨cs⟩}{⟨definition⟩}`

Renew the definition of a particle command.

The particle defined this way behaves uses `CHEMFORMULA`'s `\ch` to typeset the particle which means that the `⟨definition⟩` should be a valid `CHEMFORMULA` compound. Please have a look at the `CHEMFORMULA` manual for details. The particle will obey the `circled` option.

```
1 \NewChemParticle\positron{\chembeta+}
2 \NewChemParticle\photon{\chemgamma}
3 \RenewChemParticle\el{\chembeta-}
4 \positron\ \photon\ \el
```



8. Nomenclature, Stereo Descriptors, Latin Phrases

8.1. IUPAC Names

Similar to the `bpchem` package `CHEMMACROS` provides a command¹⁰ to typeset IUPAC names. Why is that useful? IUPAC names can get very long. So long indeed that they span over more than two lines, especially in two-column documents. This means they must be allowed to be broken more than one time. This is what the following command does.

`\iupac{⟨IUPAC name⟩}`

Inside this command use `\|` and `\-` to indicate a breaking point or a breaking dash. Use `\^` as a shortcut for `\textsuperscript`.

```
1 \begin{minipage}{.4\linewidth}
2   \iupac{%
3     Tetra\|cyclo[2.2.2.1\^{1,4}]\-un\|decane-2\ -dodecyl\-%
4     5\-(hepta\|decyl\|iso\|dodecyl\|thio\|ester)%
5   }
6 \end{minipage}
```

¹⁰ The idea and the implementation is shamelessly borrowed from `bpchem` by Bjørn Pedersen.

Tetracyclo[2.2.2.1^{1,4}]-undecane-2-dodecyl-5-(heptadecylisododecylthioester)

The `\iupac` command is more of a semantic command. Most times you can achieve (nearly) the same thing by using `\-` instead of `\|`, `-` instead of `\-` and `\textsuperscript` instead of `\^`.

There are some subtleties: `\-` inserts a small space before the hyphen and removes a small space after it. The command `\|` not only prevents ligatures but also inserts a small space.

<code>1 \huge\iupac{2,4\text{-Di}\ chlor\ pentan} \par</code> <code>2 2,4-Dichloropentan</code>	<div style="font-size: 1.5em;">2,4-Dichloropentan</div> <div style="font-size: 1.5em;">2,4-Dichloropentan</div>
--	---

The spaces inserted by these commands can be customized.

`iupac` » `hyphen-pre-space = {<dim>}` Default: .01em

Set the space that is inserted before the hyphen set with `\-`.

`iupac` » `hyphen-post-space = {<dim>}` Default: - .03em

Set the space that is inserted after the hyphen set with `\-`.

`iupac` » `break-space = {<dim>}` Default: .01em

Set the space inserted by `\|`.

The command `\iupac` serves another purpose, too, however. Regardless of the setting of the `iupac` option all the commands presented in this section are always defined *inside* `\iupac`. Quite a number of the naming commands have very general names: `\meta`, `\D`, `\E`, `\L`, `\R`, `\S`, `\trans` and so forth.¹¹ This means they either are predefined already (`\L` `\L`) or are easily defined by another package or class (the `cool` package defines both `\D` and `\E`, for example). In order to give you control which commands are defined in which way, there is the package option `iupac`. It has three modes:

- `iupac = {auto}`: if the commands are *not* defined by any package or class you're using they are available generally, otherwise only *inside* `\iupac`.
- `iupac = {restricted}`: all naming commands are *only* defined inside `\iupac`. If the commands are defined by another package they of course have that meaning outside. They're not defined outside otherwise.
- `iupac = {strict}`: `CHEMMACROS` overwrites any other definition and makes the commands available throughout the document. Of course the commands can be redefined (but only in the document body). They will still be available inside `\iupac` then.

Table 3 demonstrates the different modes.

¹¹. Please read page 5 before you consider using the one-letter commands

TABLE 3: Demonstration of `iupac`'s modes.

	auto	restricted	strict
<code>\L</code>	L	L	L
<code>\iupac{\L}</code>	L	L	L
<code>\D</code>	D	—	D
<code>\iupac{\D}</code>	D	D	D

TABLE 4: IUPAC shortcuts for Greek letters.

macro	<code>\a</code>	<code>\b</code>	<code>\g</code>	<code>\d</code>	<code>\k</code>	<code>\m</code>	<code>\n</code>	<code>\w</code>
letter	α	β	γ	δ	κ	μ	η	ω

8.1.1. Predefined Commands

The macros in this section are intended to make the writing of IUPAC names more convenient.

Greek Letters Greek letters in compound names are typeset upright. For this there are for example the packages `upgreek` and `textgreek`. If you have loaded one of them¹² the following commands typeset upright Greek letters:

`\chemalpha` α
Upright lowercase alpha

`\chembeta` β
Upright lowercase alpha

`\chemgamma` γ
Upright lowercase alpha

`\chemdelta` δ
Upright lowercase alpha

There exist two commands for each of the twenty-four Greek letters: a lowercase and an uppercase version (`\chemalpha` and `\chemAlpha`). Those commands are actually provided by the `CHEMGREEK` package. For more details refer to its documentation.

There are a number of one-letter commands that some people may find convenient to use which use above mentioned commands to print Greek letters inside `\iupac`. They're listed in table 4. But please read page 5 first before you use them.

```

1 \iupac{5\chemalpha\androstano-3\chembeta-ol} \par
2 \iupac{\chemalpha-(tri\|chloro\|methyl)\-chemomega
3 \-chloro\|poly(1,4\|phenylene\|methylene)}
```

¹² There are other options, see the description of the `greek` option.

5 α -androstan-3 β -ol
 α -(trichloromethyl)- ω -chloropoly(1,4-phenylenemethylene)

Changed in
version 4.3

Hetero Atoms and added Hydrogen Attachments to hetero atoms and added hydrogen atoms are indicated by italic letters [Coh+08]. **CHEMMACROS** defines a few macros for the most common ones.

\hydrogen *H*

The italic H for hydrogen. (An alias for this command is **\H**. But please read page 5 first before you use it.)

\oxygen *O*

The italic O for oxygen. (An alias for this command is **\O**. But please read page 5 first before you use it.)

\nitrogen *N*

The italic N for nitrogen. (An alias for this command is **\N**. But please read page 5 first before you use it.)

\sulfur *S*

The italic S for sulfur. (An alias for this command is **\Sf**. But please read page 5 first before you use it.)

\phosphorus *P*

The italic P for phosphorus. (An alias for this command is **\P**. But please read page 5 first before you use it.)

1	<code>\iupac{\nitrogen\ -methyl\ benz\ amide}</code>	<i>N</i> -methylbenzamide
2		
3	<code>\iupac{3\hydrogen\ -pyrrole}</code>	3 <i>H</i> -pyrrole
4		<i>O</i> -ethyl hexanethioate
5	<code>\iupac{\oxygen\ -ethyl hexanethioate}</code>	

Cahn-Ingold-Prelog

\cip{*{conf}*}

Typeset Cahn-Ingol-Prelog descriptors, e. g.: **\cip**{R,S} (*R,S*)

\rectus (*R*)

Typeset rectus descriptor. (An alias for this command is **\R**. But please read page 5 first before you use it.)

\sinister (*S*)

Typeset sinister descriptor. (An alias for this command is **\S**. But please read page 5 first before you use it.)

Both these commands and the entgegen/zusammen descriptors get a small additional amount of kerning after the closing parenthesis. This amount can be changed through the following option:

iupac » `cip-kern = {⟨dim⟩}` Default: .075em

Set the amount of kerning after the closing parenthesis.

Fischer

`\dexter` *D*

Typeset dexter descriptor. (An alias for this command is `\D`. But please read page 5 first before you use it.)

`\laevus` *L*

Typeset laevus descriptor. (An alias for this command is `\L`. But please read page 5 first before you use it.)

cis/trans, zusammen/entgegen, syn/anti & tert

`\cis` *cis* `\trans` *trans* `\fac` *fac* `\mer` *mer* `\zusammen` (*Z*) `\entgegen` (*E*) `\syn`
`syn` `\anti` *anti* `\tert` *tert*

An alias for `\entgegen` is `\E` and an alias for `\zusammen` is `\Z`. But please read page 5 first before you use them.

ortho/meta/para



`\ortho` *o* `\meta` *m* `\para` *p*

Although these commands are provided I like to cite [PPRo4]:



The letters *o*, *m*, and *p* have been used in place of *ortho*, *meta*, and *para*, respectively, to designate the 1,2-, 1,3-, and 1,4- isomers of disubstituted benzene. This usage is strongly discouraged and is not used in preferred IUPAC names. *IUPAC Blue Book [PPRo4, p. 90]*

Absolute Configuration (uses TikZ)

`\Rconf[⟨letter⟩]`

`\Rconf:`  `\Rconf[]:` 

`\Sconf[⟨letter⟩]`

`\Sconf:`  `\Sconf[]:` 

Examples:


```

1 \iupac{\dexter\ -Wein\|s\"aure} =
2 \iupac{\cip{2S,3S}\ -Wein\|s\"aure} \par
3 \iupac{\dexter\ - ($-$)\ -Threose} =
4 \iupac{\cip{2S,3R}\ - ($-$)\ -2,3,4\ -Tri\|hydroxy\|butanal} \par
5 \iupac{\cis\ -2\ -Butene} =
6 \iupac{\zusammen\ -2\ -Butene}, \par
7 \iupac{\cip{2E,4Z}\ -Hexa\|diene} \par
8 \iupac{\meta\ -Xylol} =
9 \iupac{1,3\ -Di\|methyl\|benzene}

```

D-Weinsäure = (2*S*,3*S*)-Weinsäure
 D-(−)-Threose = (2*S*,3*R*)-(−)-2,3,4-Trihydroxybutanal
cis-2-Butene = (*Z*)-2-Butene,
 (2*E*,4*Z*)-Hexadiene
m-Xylol = 1,3-Dimethylbenzene

Coordination Chemistry **CHEMMACROS** provides a few commands useful with coordination chemistry:

`\bridge{<num>}` μ_3
 Denote bridging ligand connection.

`\hapto{<num>}` η^5
 Denote hapticity.

`\dento{<num>}` κ^2
 Denote denticity.

Introduced in
version 4.3

```

1 Ferrocene = \iupac{bis(\hapto{5}cyclo\|penta\|dienyl)iron} \par
2 \iupac{tetra\ - \bridge{3}iodido\ - tetrakis[tri\|methyl\|platinum(IV)]}

```

Ferrocene = bis(η^5 cyclopentadienyl)iron
 tetra- μ_3 iodido-tetrakis[trimethylplatinum(IV)]

Two options allow customization:

`iupac » bridge-number = sub|super` Default: sub
 Appends the number as a subscript or superscript. IUPAC recommendation is the subscript [Con+05].

`iupac » coord-use-hyphen = true|false` Default: true
 Append a hyphen to `\hapto`, `\dent` and `\bridge` or don't.

8.1.2. Own Naming Commands

If you find any commands missing you can define them using

Introduced in
version 4.3

`\NewChemIUPAC{<cs>}{<declaration>}`

Define a new IUPAC command that is in any case defined inside of `\iupac` regardless if `<cs>` is defined elsewhere already.

`\RenewChemIUPAC{<cs>}{<declaration>}`

Redefine an existing IUPAC command that is in any case defined inside of `\iupac` regardless if `<cs>` is defined elsewhere already.

Changed in
version 4.3

`\DeclareChemIUPAC{<cs>}{<declaration>}`

Define a new IUPAC command that is in any case defined inside of `\iupac` regardless if `<cs>` is defined elsewhere already. This silently overwrites an existing IUPAC definition.

A command defined in this way will obey the setting of the option `iupac`. This means any existing command is only overwritten with `iupac = {strict}`. However, `\NewChemIUPAC` will *not* change the definition of an existing IUPAC naming command but issue an error if the IUPAC naming command already exists. `\DeclareChemIUPAC` will overwrite an existing IUPAC command.

```

1 \NewChemIUPAC\endo{\textit{endo}}
2 \RenewChemIUPAC\anti{\textit{anti}}
3 \iupac{(2\-\endo,7\-\anti)\-2\-\bromo\-\7\-\fluoro\|bicyclo[2.2.1]heptane}

```

(2-endo,7-anti)-2-bromo-7-fluorobicyclo[2.2.1]heptane

`\RenewChemIUPAC` allows you to redefine the existing IUPAC naming commands.

1 <code>\iupac{\meta{-Xylol}} \par</code>	<i>m</i> -Xylol
2 <code>\RenewChemIUPAC\meta{\textup{m}}</code>	m-Xylol
3 <code>\iupac{\meta{-Xylol}}</code>	

8.2. Latin Phrases

The package `chemstyle` provides the command `\latin` to typeset common latin phrases in a consistent way. `CHEMMACROS` defines a similar `\latin` only if `chemstyle` has *not* been loaded and additionally provides these commands:

`\insitu` *in situ* `\abinitio` *ab initio* `\invacuo` *in vacuo*

If the package `chemstyle` has been loaded they are defined using `chemstyle`'s `\latin` command. This means that then the appearance depends on `chemstyle`'s option `abbremph`.

The commands are defined through

Introduced in
version 4.3

`\NewChemLatin{⟨cs⟩}{⟨phrase⟩}`

Define a new latin phrase. Gives an error if `⟨cs⟩` already exists.

Changed in
version 4.3

`\DeclareChemLatin{⟨cs⟩}{⟨phrase⟩}`

Define a new latin phrase.

`\RenewChemLatin{⟨cs⟩}{⟨phrase⟩}`

Redefine an existing latin phrase.

```
1 \NewChemLatin\ltn{latin text}\ltn      latin text
```

If you have *not* loaded chemstyle you can change the appearance with this option:

`latin» format = {⟨definition⟩}`

Set the format of the latin phrases.

Default: `\itshape`

9. Units for the Usage With siunitx

In chemistry some non-SI units are very common. siunitx provides the command `\DeclareSIUnit{⟨command⟩}{⟨unit⟩}` to add arbitrary units. **CHEMMACROS** uses that command to provide some units. Like all siunitx units they're only valid inside `\SI{⟨num⟩}{⟨unit⟩}` and `\si{⟨unit⟩}`.

`\atmosphere`
atm

`\atm`
atm

`\calory`
cal

`\cal`
cal

`\cmc`
cm³

The units `\cmc`, `\molar`, and `\Molar` are defined by the package chemstyle as well. **CHEMMACROS** only defines them, if chemstyle is not loaded.

`\molar`
mol dm⁻³

`\moLar`
mol L⁻¹

`\Molar`
M

`\MolMass`
g mol⁻¹

`\normal`
N

`\torr`
torr

By the way: `\mmHg` mmHg already is defined by siunitx and chemstyle.

10. Acid/Base

Easy representation of pH, p*K*_A ... (the command `\pKa` depends on the package option `language`).
The translations may be adapted, though, see section 6.

`\pH`
pH

`\pOH`
pOH

`\Ka`
*K*_A

`\Kb`
*K*_B

`\Kw`
*K*_W

`\pKa[⟨num⟩]`
`\pKa`: p*K*_A, `\pKa[1]`: p*K*_{A1}

`\pKb[⟨num⟩]`
`\pKb`: p*K*_B, `\pKb[1]`: p*K*_{B1}

`\p{⟨anything⟩}`
e. g. `\p{\Kw}` p*K*_W

<code>\Ka</code> <code>\Kb</code> <code>\pKa</code> <code>\pKa[1]</code> <code>\pKb</code> <code>\pKb[1]</code>	<i>K</i> _A <i>K</i> _B p <i>K</i> _A p <i>K</i> _{A1} p <i>K</i> _B p <i>K</i> _{B1}
---	---

The operator p [...] shall be printed in Roman type. *IUPAC Green Book [Coh+08, p. 103]*

There is one option which changes the style the p is typeset:

`acid-base` » `p-style` = `italics|slanted|upright` Default: upright
Set the style of the p operator.

`acid-base` » `K-acid` = $\{\langle text \rangle\}$ Default: A
The subscript to `\Ka` and `\pKa`.

`acid-base` » `K-base` = $\{\langle text \rangle\}$ Default: B
The subscript to `\Kb` and `\pKb`.

`acid-base` » `K-water` = $\{\langle text \rangle\}$ Default: W
The subscript to `\Kw`.

```
1 \pH, \pKa \par
2 \chemsetup{acid-base}{p-style=slanted} \pH, \pKa \par
3 \chemsetup{acid-base}{p-style=italics} \pH, \pKa
```

$\text{pH}, \text{p}K_A$
 $p\text{H}, pK_A$
 $p\text{H}, pK_A$

Introduced in
version 4.2d

As you can see the default subscripts of `\Kw`, `\Ka` and `\Kb` are uppercase letters. The literature is inconclusive about if this is the right way or if lowercase letters should be preferred. In textbooks the uppercase variant usually seems to be used while journals seem to prefer the lowercase variant. Since I like the uppercase version better this is the default. If you want to change this you have two possibilities:

```
1 % this works only in the preamble:
2 % \DeclareTranslation{English}{K-acid}{a}% use your language here
3 % alternative:
4 \chemsetup{acid-base/K-acid=a}% overwrites language dependent settings
5 \pKa
```

$\text{p}K_a$

11. Oxidation Numbers, Real and Formal Charges

CHEMMACROS distinguishes between real (+/−) and formal (⊕/⊖) charge symbols, also see section 4. All commands using formal charge symbols start with a f.

11.1. Ion Charges

Simple displaying of (real) charges. It is worth noting that these commands really are relicts from a time when **CHEMMACROS** tried hard to be compliant with mhchem and **CHEMFORMULA** didn't exist, yet. They are still provided for backwards compatibility but *my recommendation is to use `\ch`* (see the documentation of the **CHEMFORMULA** package) *and forget about these commands*:

`\pch[⟨number⟩]`
positive charge (**p**lus + **ch**arge)

`\mch[⟨number⟩]`
negative charge (**m**inus + **ch**arge)

```
1 \leavevmode
2 \pch, Na\pch, Ca\pch[2]\par
3 \leavevmode
4 \mch, F\mch, S\mch[2]
```

$$^+, \text{Na}^+, \text{Ca}^{2+}$$

$$^-, \text{F}^-, \text{S}^{2-}$$

The same for formal charges:

`\fpch[⟨number⟩]`
positive charge

`\fmch[⟨number⟩]`
negative charge

```
1 \leavevmode
2 \fpch\ \fmch\ \fpch[3] \fmch[3]
```

$$\oplus \ominus 3\oplus 3\ominus$$

11.2. Oxidation Numbers

Typesetting oxidation numbers:

`\ox[⟨options⟩]{⟨number⟩,⟨atom⟩}`

Places `⟨number⟩` above `⟨atom⟩`; `⟨number⟩` has to be a (rational) number!

```
1 \ox{+1,Na}, \ox{2,Ca}, \ox{-2,S}, \ox{-1,F}
```

$$\overset{\text{I}}{\text{Na}}, \overset{\text{II}}{\text{Ca}}, \overset{-\text{II}}{\text{S}}, \overset{-\text{I}}{\text{F}}$$

There are a number of keys, that can be used to modify the `\ox` command.

11. Oxidation Numbers, Real and Formal Charges

`ox » parse = true|false` Default: true
When false an arbitrary entry can be used for <number>.

`ox » roman = true|false` Default: false
Switches from roman to arabic numbers.

`ox » pos = top|super|side` Default: top
top places <number> above <atom>, super to the upper right as superscript and side to the right and inside brackets.

`ox » explicit-sign = true|false` Default: false
Shows the + for positiv numbers and the ± for 0.

`ox » decimal-marker = comma|point` Default: point
Choice for the decimal marker for formal oxidation numbers like $X^{1.2}$.

`ox » align = center|right` Default: center
Center the oxidation number relative to the atom or right-align it.

<code>1 \ox[roman=false]{2,Ca} \ox{2,Ca} \</code>	$\overset{2}{\text{Ca}} \overset{\text{II}}{\text{Ca}}$
<code>2 \ox[pos=super]{3,Fe}-Oxide \</code>	$\text{Fe}^{\text{III}}\text{-Oxide}$
<code>3 \ox[pos=side]{3,Fe}-Oxide \</code>	Fe(III)-Oxide
<code>4 \ox[parse=false]{?,Mn} \</code>	$\overset{?}{\text{Mn}}$
<code>5 \ox[align=right]{2,Ca}</code>	$\overset{\text{II}}{\text{Ca}}$

The `pos = {super}` variant also can be set with the shortcut `\ox*`:

<code>1 \ox{3,Fe} \ox*{3,Fe}</code>	$\overset{\text{III}}{\text{Fe}} \text{Fe}^{\text{III}}$
-------------------------------------	--

Using the `explicit-sign` key will always show the sign of the oxidation number:

```
1 \chemsetup[ox]{explicit-sign = true}
2 \ox{+1,Na}, \ox{2,Ca}, \ox{-2,S}, \ch{"\ox{0,F}" {2}}
```

$\overset{+1}{\text{Na}}, \overset{+II}{\text{Ca}}, \overset{-II}{\text{S}}, \overset{\pm 0}{\text{F}_2}$

```
1 Compare \ox{-1,\ch{O2^2-}} to \ch{"\ox{-1,0}" {2}^2-}
```

Compare $\overset{-1}{\text{O}_2^{2-}}$ to $\overset{-1}{\text{O}_2^{2-}}$

Sometimes one might want to use formal oxidation numbers like 0.5 or $\frac{1}{3}$:

```
1 \ox{.5,\ch{Br2}} \ch{"\ox{1/3,I}" {}3+} Br2 I30.5 1/3+
```

The fraction uses the `\sfrac` command of the `xfrac` package. For this purpose the instance `chemmacros-ox-frac` is defined.

```
1 \DeclareInstance{xfrac}{chemmacros-ox-frac}{text}{
2   scale-factor      = 1.2 ,
3   denominator-bot-sep = -.5ex ,
4   numerator-top-sep  = -.3ex ,
5   slash-left-kern    = -.2em ,
6   slash-right-kern   = -.2em ,
7   slash-symbol-font  = \mr
8 }
```

Of course you can redefine it so that it suits your needs as the output often strongly depends on the used font.

11.3. Partial Charges and Similar Stuff

The next ones probably are seldomly needed but nevertheless useful:

`\delp`

δ^+ (**d**elta + **p**lus)

`\delm`

δ^- (**d**elta + **m**inus)

`\fdelp`

δ^\oplus

`\fdelm`

δ^\ominus

These macros for example can be used with the `\ox` command or with the `chemfig` package:

```
1 \chemsetup{
2   option/circled = all,
3   ox/parse      = false
4 }
5 \ch{"\ox{\delp,H}" -{} "\ox{\delm,Cl}" } \hspace*{1cm}
6 \chemfig{\chemabove[3pt]{\lewis{246,Br}}{\delm}-\chemabove[3pt]{H}{\delp}}
```




The following macros are useful together with chemfig, too.

`\scrp`

+ (`\scriptstyle` + **plus**)

`\scrm`

- (`\scriptstyle` + **minus**)

`\fscrp`

\oplus

`\fscrm`

\ominus

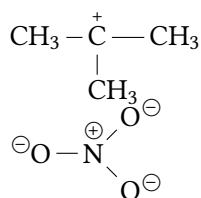
`\fsscrp`

\oplus (using `\scriptscriptstyle`)

`\fsscrm`

\ominus

```
1 \setatomsep{1.8em}\chemfig{CH_3-\chemabove{C}{\scrp}{-[6]C|H_3}-\vphantom{H_3}
   CH_3}
2
3 \chemfig{\fmch{}}|0-\chemabove{N}{\fscrp}{-[1]O|\fmch}-[7]O|\fmch}
```



12. Reaction Mechanisms

`\mech[⟨type⟩]`

Allows to specify the most common reaction mechanisms.

⟨type⟩ can have one of the following values:

`\mech`

(empty, no opt. argument) nucleophilic substitution S_N

`\mech[1]`

unimolecular nucleophilic substitution S_N1

`\mech[2]`
bimolecular nucleophilic substitution S_N2

`\mech[se]`
electrophilic substitution S_E

`\mech[1e]`
unimolecular electrophilic substitution S_{E1}

`\mech[2e]`
bimolecular electrophilic substitution S_{E2}

`\mech[ar]`
electrophilic aromatic substitution $Ar-S_E$

`\mech[e]`
elimination E

`\mech[e1]`
unimolecular elimination $E1$

`\mech[e2]`
bimolecular elimination $E2$

`\mech[cb]`
unimolecular elimination “conjugated base”, *i. e.*, via carbanion $E1_{cb}$

13. Redox Reactions

CHEMMACROS provides two commands to visualize the transfer of electrons in redox reactions. Both commands are using TikZ.

`\OX{<name>,<atom>}`
Label $\langle atom \rangle$ with the label $\langle name \rangle$.

`\redox(<name1>,<name2>)[<tikz>][<num>]{<text>}`
Connect two $\langle atom \rangle$ s previously labelled with `\OX`. Only the first argument ($\langle name1 \rangle, \langle name2 \rangle$) is required, the others are all optional.

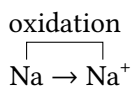
`\OX` places $\langle atom \rangle$ into a node, which is named with $\langle name \rangle$. If you have set two `\OX`, they can be connected with a line using `\redox`. To do so the names of the two nodes that are to be connected are written in the round braces. Since `\redox` draws a `tikzpicture` with options `remember picture, overlay`, the document needs to be *compiled at least two times*.

```

1 \vspace{7mm}
2 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b){oxidation}

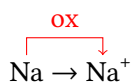
```

13. Redox Reactions



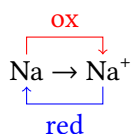
This line can be customized using TikZ keys in [*tikz*]:

```
1 \vspace{7mm}
2 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b)[->,red]{ox}
```



With the argument [*num*] the length of the vertical parts of the line can be adjusted. The default length is .6em. This length is multiplied with *num*. If you use a negative value the line is placed *below* the text.

```
1 \vspace{7mm}
2 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch
3 \redox(a,b)[->,red]{ox}
4 \redox(a,b)[->,blue][-1]{red}
5 \vspace{7mm}
```



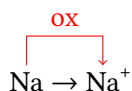
The default length of the vertical lines can be customized with the option

`redox » dist = {<dim>}`

Default: .6em

A TeX dimension.

```
1 \vspace{7mm}
2 \chemsetup{redox/dist=1em}
3 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b)[->,red]{ox}
```



`redox » sep = {<dim>}`

Default: .2em

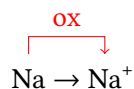
The option can be used to change the distance between the atom and the beginning of the line.

13. Redox Reactions

```

1 \vspace{7mm}
2 \chemsetup{redox/sep=.5em}
3 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b)[->,red]{ox}

```

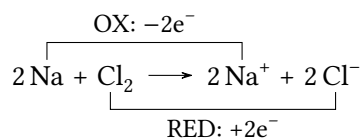


Examples:

```

1 \vspace{7mm}
2 \ch{
3   2 "\OX{o1,Na}" + "\OX{r1,Cl}" {}2
4   ->
5   2 "\OX{o2,Na}" {}+ + 2 "\OX{r2,Cl}" {}-
6 }
7 \redox(o1,o2){\small OX: $- 2\text{el}$}
8 \redox(r1,r2)[][-1]{\small RED: $+ 2\text{el}$}
9 \vspace{7mm}

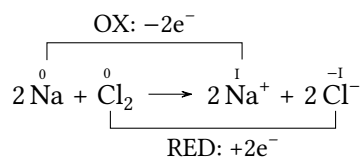
```



```

1 \vspace{7mm}
2 \ch{
3   2 "\OX{o1,\ox{0,Na}}" + "\OX{r1,\ox{0,Cl}}" {}2
4   ->
5   2 "\OX{o2,\ox{+1,Na}}" {}+ + 2 "\OX{r2,\ox{-1,Cl}}" {}-
6 }
7 \redox(o1,o2){\small OX: $- 2\text{el}$}
8 \redox(r1,r2)[][-1]{\small RED: $+ 2\text{el}$}
9 \vspace{7mm}

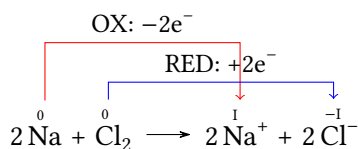
```



```

1 \vspace{14mm}
2 \ch{
3   2 "\OX{o1,\ox{0,Na}}" + "\OX{r1,\ox{0,Cl}}" {}2
4   ->
5   2 "\OX{o2,\ox{+1,Na}}" {}+ + 2 "\OX{r2,\ox{-1,Cl}}" {}-
6 }
7 \redox(o1,o2)[draw=red,->][3.33]{\small OX: $- 2\text{e}^-$}
8 \redox(r1,r2)[draw=blue,->]{\small RED: $+ 2\text{e}^-$}

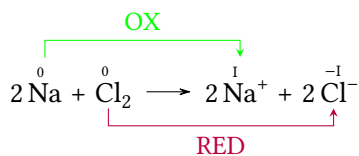
```



```

1 \vspace{7mm}
2 \ch{
3   2 "\OX{o1,\ox{0,Na}}" + "\OX{r1,\ox{0,Cl}}" {}2
4   -> 2 "\OX{o2,\ox{+1,Na}}" {}+ + 2 "\OX{r2,\ox{-1,Cl}}" {}-
5 }
6 \redox(o1,o2)[green,-stealth]{\small OX}
7 \redox(r1,r2)[purple,-stealth][ -1]{\small RED}
8 \vspace{7mm}

```



14. (Standard) State, Thermodynamics

14.1. Thermodynamic Variables

The following commands use siunitx:

`\Enthalpy[⟨options⟩](⟨subscript⟩){⟨value⟩}`

Typeset the amount of enthalpy.

`\Entropy[⟨options⟩](⟨subscript⟩){⟨value⟩}`

Typeset the amount of entropy.

`\Gibbs[⟨options⟩](⟨subscript⟩){⟨value⟩}`

Typeset the amount of Gibbs enthalpy.

Their usage is pretty much self-explaining:

1 <code>\Enthalpy{123} \par</code>	$\Delta H^\circ = 123 \text{ kJ mol}^{-1}$
2 <code>\Entropy{123} \par</code>	$S^\circ = 123 \text{ J K}^{-1} \text{ mol}^{-1}$
3 <code>\Gibbs{123}</code>	$\Delta G^\circ = 123 \text{ kJ mol}^{-1}$

The argument (*<subscript>*) adds a subscript for specification: `\Enthalpy(r){123}` $\Delta_r H^\circ = 123 \text{ kJ mol}^{-1}$.

There are several keys to customize the commands. They do not belong to a module and can only be used in the optional arguments of the commands.

exponent = {*<anything>*}

Choose *<anything>* as exponent.

delta = *<anything>* | false

Disable or choose a symbol in front of the main symbol. *<anything>* will be placed in math mode!

subscript = left | right

Choose if the subscript is placed to the left or the right of the main symbol.

unit = {*<unit>*}

Set the unit of the variable.

The default values depend on the command.

1 <code>\Enthalpy[unit=\kilo\joule]{-285} \par</code>	$\Delta H^\circ = -285 \text{ kJ}$
2 <code>\Gibbs[delta=false]{0} \par</code>	$G^\circ = 0 \text{ kJ mol}^{-1}$
3 <code>\Entropy[delta=\Delta,exponent=]{56.7}</code>	$\Delta S = 56.7 \text{ J K}^{-1} \text{ mol}^{-1}$

The unit is set corresponding to the rules of siunitx and depends on its settings:

1 <code>\Enthalpy{-1234.56e3} \par</code>	
2 <code>\sisetup{</code>	
3 <code>per-mode=symbol,</code>	$\Delta H^\circ = -1234.56 \times 10^3 \text{ kJ mol}^{-1}$
4 <code>exponent-product=\cdot,</code>	$\Delta H^\circ = -1\,234,56 \cdot 10^3 \text{ kJ/mol}$
5 <code>output-decimal-marker={,},</code>	
6 <code>group-four-digits=true</code>	
7 <code>}</code>	
8 <code>\Enthalpy{-1234.56e3}</code>	

14.1.1. Create New Variables or Redefine Existing Ones

`\NewChemState{<name>}[<options>]{<symbol>}{<unit>}`

Define new corresponding commands. *<name>* may either be a control sequence token or a

control sequence name without leading backslash. This means that `\DeclareChemState{name}` and `\DeclareChemState{\name}` are equivalent. The reason for this rather strange definition is a syntax change in `\DeclareChemState` while retaining backwards compaitibility. The latter version is recommended though and the former version may deprecate in the future. Gives an error if $\langle name \rangle$ already exists

`\RenewChemState{<name>}[<options>]{<symbol>}{<unit>}`

Changed in
version 4.2

Redefine existing state commands. $\langle name \rangle$ may either be a control sequence token or a control sequence name without leading backslash. This means that `\RenewChemState{name}` and `\RenewChemState{\name}` are equivalent. The reason for this rather strange definition is a syntax change in `\RenewChemState` while retaining backwards compaitibility. The latter version is recommended though and the former version may deprecate in the future.

`\DeclareChemState{<name>}[<options>]{<symbol>}{<unit>}`

Changed in
version 4.3

Like `\NewChemState` but gives now error if $\langle name \rangle$ already exists.

```
1 \NewChemState\Helmholtz{A}{\kilo\joule\per\mole}
2 \NewChemState\ElPot[subscript-left=false,exponent=]{E}{\volt}
3 \Helmholtz{123.4} \par
4 \ElPot{-1.1} \par
5 \ElPot[exponent=0]({\ch{Sn}}|\ch{Sn^2+}||\ch{Pb^2+}|\ch{Pb}){0.01}
```

$$\Delta A^\circ = 123.4 \text{ kJ mol}^{-1}$$

$$\Delta E = -1.1 \text{ V}$$

$$\Delta E_{\text{Sn}|\text{Sn}^{2+}||\text{Pb}^{2+}|\text{Pb}}^0 = 0.01 \text{ V}$$

The command has some keys with which the default behaviour of the new command can be set.

`exponent = {<anything>}`

Default: `\standardstate`

Set the default exponent.

`delta = <anything>|false`

Default: `\changestate`

Choose the default “delta” symbol that is placed in front of the main symbol. $\langle anything \rangle$ will be placed in math mode!

`subscript-left = true|false`

Default: true

Choose the default position of the sybscript.

`subscript = {<anything>}`

(initially empty)

Choose the default subscript symbol.

```

1 \RenewChemState\Enthalpy{h}{\joule}
2 \Enthalpy{f}{12.5}

```

$$\Delta_f h^\ominus = 12.5 \text{ J}$$

The existing commands have been defined like this:

```

1 \NewChemState\Enthalpy{H}{\kilo\joule\per\mole}
2 \NewChemState\Entropy[delta=false,subscript-left=false]{S}
3   {\joule\per\kelvin\per\mole}
4 \NewChemState\Gibbs{G}{\kilo\joule\per\mole}

```

So – for following thermodynamic conventions – one could define a molar and an absolute variable:

```

1 \NewChemState\enthalpy[exponent=]{h}{\kilo\joule\per\mole}% molar
2 \RenewChemState\Enthalpy[exponent=]{H}{\kilo\joule}% absolute
3 \enthalpy{-12.3} \Enthalpy{-12.3}

```

$$\Delta h = -12.3 \text{ kJ mol}^{-1} \quad \Delta H = -12.3 \text{ kJ}$$

14.2. State

The commands presented in section 14.1 internally all use the command¹³

`\State[⟨options⟩]{⟨symbol⟩}{⟨subscript⟩}`

Typeset a state variable.

It can be used to write the thermodynamic state variables without value and unit.

```

1 \State{A}, \State{G}{f},
2 \State[subscript-left=false]{E}{\ch{Na}},
3 \State[exponent=\SI{1000}]{\celsius}{H}

```

$$\Delta A^\ominus, \Delta_f G^\ominus, \Delta E_{\text{Na}}^\ominus, \Delta H^{1000^\circ\text{C}}$$

Again there are some keys to customize the command:

`state` » `exponent = {⟨anything⟩}`

Set the default exponent symbol.

13. Please note that `{⟨subscript⟩}` is an *optional* argument.

`state` » `subscript-left = true|false`

Set the default subscript position.

`state` » `delta = <anything>|false`

Set the default “delta” symbol. *<anything>* will be placed in math mode!

15. Spectroscopy and Experimental Data

15.1. The `\NMR` Command

When you’re trying to find out if a compound is the one you think it is often NMR spectroscopy is used. The experimental data are typeset similar to this:

$^1\text{H-NMR (400 MHz, CDCl}_3\text{): } \delta = 1.59$

CHEMMACROS provides a command which simplifies the input (uses `siunitx`).

`\NMR*{<num>,<element>}(<num>,<unit>)[<solvent>]`

Typeset nuclear magnetic resonance data.

All Argument are optional! Without arguments we get:

<code>1 \NMR \par</code>	$^1\text{H-NMR: } \delta$
<code>2 \NMR*</code>	$^1\text{H-NMR}$

The first argument specifies the kind of NMR:

<code>1 \NMR{13,C}</code>	$^{13}\text{C-NMR: } \delta$
---------------------------	------------------------------

The second argument sets the frequency (in MHz):

<code>1 \NMR(400)</code>	$^1\text{H-NMR (400 MHz): } \delta$
--------------------------	-------------------------------------

You can choose another unit:

<code>1 \NMR(4e8,\hertz)</code>	$^1\text{H-NMR (4} \times 10^8 \text{ Hz): } \delta$
---------------------------------	--

Please note that the setup of `siunitx` also affects this command:

<code>1 \sisetup{exponent-product=\cdot}</code>	
<code>2 \NMR(4e8,\hertz)</code>	$^1\text{H-NMR (4} \cdot 10^8 \text{ Hz): } \delta$

The third argument specifies the solvent:

<code>1 \NMR[CDCl3]</code>	$^1\text{H-NMR (CDCl}_3\text{): } \delta$
----------------------------	---

15.2. Short Cuts

It is possible to define short cut commands for specific nuclei.

Introduced in
version 4.3

`\NewChemNMR{<cs>}{<num>,<atom>}`

Define a new shortcut macro for typesetting a certain type of magnetic resonance data. Gives an error if <cs> already exists.

Changed in
version 4.3

`\DeclareChemNMR{<cs>}{<num>,<atom>}`

Define a new shortcut macro for typesetting a certain type of magnetic resonance data.

`\RenewChemNMR{<cs>}{<num>,<atom>}`

Redefine an existing shortcut macro for typesetting a certain type of magnetic resonance data.

This defines a command with the same arguments as `\NMR` except for `{<num>,<atom>}`:

1	<code>\NewChemNMR\HNMR{1,H}%</code>	^{13}C -NMR (100 MHz)
2	<code>\NewChemNMR\CNMR{13,C}%</code>	
3	<code>\CNMR*(100) \par</code>	^1H -NMR (400 MHz)
4	<code>\HNMR*(400)</code>	

15.3. An Environment to Typeset Experimental Data

`CHEMMACROS` provides an environment to ease the input of experimental data.

`\begin{experimental}`

Environment for the output of experimental data. Inside the environment the following commands are defined.

`\data{<type>}[<specification>]`

Type of data, e. g. IR, MS... The optional argument takes further specifications which are output in parentheses.

`\data*{<type>}[<specification>]`

Like `\data` but changes the = into a :, given that `use-equal = {true}` is used.

`\NMR{<num>,<elem>[<coupling core>]}(<num>,<unit>)[<solvent>]`

This command gets an additional argument: `\NMR{13,C[^1H]}` $^{13}\text{C}\{^1\text{H}\}$ -NMR: δ

`\J(<bonds>;<nuclei>)[<unit>]{<list of nums>}`

Coupling constant, values are input separated by ; (NMR). The argument `(<bonds>;<nuclei>)` and `[<unit>]` are optional and enable further specifications of the coupling.

`\#{<num>}`

Number of nuclei (NMR).

`\pos{<num>}`

Position of nuclues (NMR).

`\val{<num>}`

A number, an alias of siunitx' `\num{<num>}`.

`\val{<num1>--<num2>}`

An alias of siunitx' `\numrange{<num1>}{<num2>}`.

```

1 \begin{experimental}
2   \data{type1} Data.
3   \data{type2}[specifications] More data.
4   \data*{type3} Even more data.
5 \end{experimental}

```

type1 Data. type2 (specifications) More data. type3 Even more data.

15.4. Customization

The output of the environment and of the NMR commands can be customized by a number of options. For historical reasons they all belong to the module `nmr`.

`nmr` » `unit = {<unit>}`

The used default unit.

Default: `\mega\hertz`

`nmr` » `nucleus = {<num>,<atom>}`

The used default nucleus.

Default: `{1,H}`

`nmr` » `format = {<commands>}`

For example `\bfseries`.

(initially empty)

`nmr` » `pos-number = side|sub|super`

Position of the number next to the atom.

Default: `side`

Changed in
version 4.3

`nmr` » `coupling-symbol = {<code>}`

The symbol used for the coupling constant.

Default: `J`

Introduced in
version 4.3

`nmr` » `coupling-unit = {<unit>}`

A siunitx unit.

Default: `\hertz`

`nmr` » `coupling-pos = side|sub`

Placement of the coupling nuclei next to the symbol *J* (or rather the symbol specified with option `coupling-symbol`).

Default: `side`

Introduced in
version 4.3

`nmr` » `coupling-nuclei-pre = {<code>}`

Code inserted before the coupling nuclei when `coupling-pos = {side}`.

Default: `(`

Introduced in
version 4.3

`nmr` » `coupling-nuclei-post = {<code>}`

Code inserted after the coupling nuclei when `coupling-pos = {side}`.

Default: `)`

Introduced in
version 4.3

<code>nmr</code> » <code>coupling-bonds-pre</code> = {<code>}	(initially empty)
Introduced in version 4.3	Code inserted before the coupling bonds.
<code>nmr</code> » <code>coupling-bonds-post</code> = {<code>}	Default: \!
Introduced in version 4.3	Code inserted after the coupling bonds.
<code>nmr</code> » <code>parse</code> = <code>true</code> <code>false</code>	Default: <code>true</code>
	Treat the solvent as CHEMFORMULA formula or not.
<code>nmr</code> » <code>delta</code> = {<tokens>}	(initially empty)
	The <tokens> are added after δ .
<code>nmr</code> » <code>list</code> = <code>true</code> <code>false</code>	Default: <code>false</code>
	The environment <code>nmr</code> is formatted as a list
<code>nmr</code> » <code>list-setup</code> = {<setup>}	
	Setup of the list. See below for the default settings.
<code>nmr</code> » <code>use-equal</code> = <code>true</code> <code>false</code>	Default: <code>false</code>
	Add equal sign after <code>\NMR</code> and <code>\data</code> .

The default setup of the list:

```
1 \topsep\z@skip \partopsep\z@skip
2 \itemsep\z@ \parsep\z@ \itemindent\z@
3 \leftmargin\z@
```

```
1 \begin{experimental}[format=\bfseries]
2   \data{type1} Data.
3   \data{type2}[specifications] More data.
4   \data*{type3} Even more data.
5 \end{experimental}
```

type1 Data. **type2 (specifications)** More data. **type3** Even more data.

The command `\NMR` and all commands defined through `\NewChemNMR` can be used like `\data` for the NMR data.

```
1 \begin{experimental}[format=\bfseries,use-equal]
2   \data{type1} Data.
3   \data{type2}[specifications] More data.
4   \NMR Even more data.
5 \end{experimental}
```

type1 = Data. **type2 (specifications)** = More data. ¹H-NMR: δ = Even more data.

15.5. An Example

The code below is shown with different specifications for *<options>*. Of course options can also be chosen with `\chemsetup`.

```

1 \sisetup{separate-uncertainty,per-mode=symbol,detect-all,range-phrase=- -}
2 \begin{experimental}[<optionen>]
3   \data*{yield} \SI{17}{\milli\gram} yellow needles (\SI{0.04}{\milli\mole},
4     \SI{13}{\percent}).
5   %
6   \data{mp.} \SI{277}{\celsius} (DSC).
7   %
8   \NMR{600}[CDCl3] \val{2.01} (s, \#{24}, \pos{5}), \val{2.31} (s, \#{12},
9     \pos{1}), \val{6.72--6.74} (m, \#{2}, \pos{11}), \val{6.82} (s, \#{8},
10    \pos{3}), \val{7.05--7.07} (m, \#{2}, \pos{12}), \val{7.39--7.41} (m, \#{4},
11    \pos{9}), \val{7.48--7.49} (m, \#{4}, \pos{8}).
12   %
13   \NMR{13,C}{150}[CDCl3] \val{21.2} ($+$, \#{4}, \pos{1}), \val{23.4} ($+$,
14     \#{8}, \pos{5}), \val{126.0} ($+$, \#{4}, \pos{9}), \val{128.2} ($+$, \#{8},
15     \pos{3}), \val{130.8} ($+$, \#{2}, \pos{12}), \val{133.6} ($+$, \#{2},
16     \pos{11}), \val{137.0} ($+$, \#{4}, \pos{8}), \val{138.6} (q, \#{4},
17     \pos{2}), \val{140.6} (q, \#{2}, \pos{10}), \val{140.8} (q, \#{8}, \pos{4}),
18     \val{141.8} (q, \#{4}, \pos{6}), \val{145.6} (q, \#{2}, \pos{7}).
19   %
20   \data{MS}[DCP, EI, \SI{60}{\electronvolt}] \val{703} (2, \ch{M+}), \val{582}
21     (1), \val{462} (1), \val{249} (13), \val{120} (41), \val{105} (100).
22   %
23   \data{MS}[\ch{MeOH + H2O + KI}, ESI, \SI{10}{\electronvolt}] \val{720} (100,
24     \ch{M+ + OH-}), \val{368} (\ch{M+ + 2 OH-}).
25   %
26   \data{IR}[KBr] \val{3443} (w), \val{3061} (w), \val{2957} (m), \val{2918}
27     (m), \val{2856} (w), \val{2729} (w), \val{1725} (w), \val{1606} (s),
28     \val{1592} (s), \val{1545} (w), \val{1446} (m), \val{1421} (m), \val{1402}
29     (m), \val{1357} (w), \val{1278} (w), \val{1238} (s), \val{1214} (s),
30     \val{1172} (s), \val{1154} (m), \val{1101} (w), \val{1030} (w), \val{979}
31     (m), \val{874} (m), \val{846} (s), \val{818} (w), \val{798} (m), \val{744}
32     (w), \val{724} (m), \val{663} (w), \val{586} (w), \val{562} (w), \val{515}
33     (w).
34   %
35   \data*{UV-Vis} \SI{386}{\nano\metre} ($\varepsilon = \val{65984}$),
36     \SI{406}{\nano\metre} ($\varepsilon = \val{65378}$).
37   %
38   \data*{quantum yield} $\Phi = \val{0.74+-0.1}$\,,.
39 \end{experimental}

```

15.5.1. Nearly Standard

Output with these options:

```
1 delta=(ppm),pos-number=sub,use-equal
```

yield: 17 mg yellow needles (0.04 mmol, 13 %). mp. = 277 °C (DSC). ¹H-NMR (600 MHz, CDCl₃): δ (ppm) = 2.01 (s, 24 H, H₅), 2.31 (s, 12 H, H₁), 6.72–6.74 (m, 2 H, H₁₁), 6.82 (s, 8 H, H₃), 7.05–7.07 (m, 2 H, H₁₂), 7.39–7.41 (m, 4 H, H₉), 7.48–7.49 (m, 4 H, H₈). ¹³C-NMR (150 MHz, CDCl₃): δ (ppm) = 21.2 (+, 4 C, C₁), 23.4 (+, 8 C, C₅), 126.0 (+, 4 C, C₉), 128.2 (+, 8 C, C₃), 130.8 (+, 2 C, C₁₂), 133.6 (+, 2 C, C₁₁), 137.0 (+, 4 C, C₈), 138.6 (q, 4 C, C₂), 140.6 (q, 2 C, C₁₀), 140.8 (q, 8 C, C₄), 141.8 (q, 4 C, C₆), 145.6 (q, 2 C, C₇). MS (DCP, EI, 60 eV) = 703 (2, M⁺), 582 (1), 462 (1), 249 (13), 120 (41), 105 (100). MS (MeOH + H₂O + KI, ESI, 10 eV) = 720 (100, M⁺ + OH⁻), 368 (M⁺ + 2 OH⁻). IR (KBr) = 3443 (w), 3061 (w), 2957 (m), 2918 (m), 2856 (w), 2729 (w), 1725 (w), 1606 (s), 1592 (s), 1545 (w), 1446 (m), 1421 (m), 1402 (m), 1357 (w), 1278 (w), 1238 (s), 1214 (s), 1172 (s), 1154 (m), 1101 (w), 1030 (w), 979 (m), 874 (m), 846 (s), 818 (w), 798 (m), 744 (w), 724 (m), 663 (w), 586 (w), 562 (w), 515 (w). UV-Vis: 386 nm (ε = 65 984), 406 nm (ε = 65 378). quantum yield: Φ = 0.74 ± 0.10.

15.5.2. Formatted List

Output with these options:

```
1 format=\bfseries,delta=(ppm),list=true,use-equal
```

yield: 17 mg yellow needles (0.04 mmol, 13 %).

mp. = 277 °C (DSC).

¹H-NMR (600 MHz, CDCl₃): δ (ppm) = 2.01 (s, 24 H, H₅), 2.31 (s, 12 H, H₁), 6.72–6.74 (m, 2 H, H₁₁), 6.82 (s, 8 H, H₃), 7.05–7.07 (m, 2 H, H₁₂), 7.39–7.41 (m, 4 H, H₉), 7.48–7.49 (m, 4 H, H₈).

¹³C-NMR (150 MHz, CDCl₃): δ (ppm) = 21.2 (+, 4 C, C₁), 23.4 (+, 8 C, C₅), 126.0 (+, 4 C, C₉), 128.2 (+, 8 C, C₃), 130.8 (+, 2 C, C₁₂), 133.6 (+, 2 C, C₁₁), 137.0 (+, 4 C, C₈), 138.6 (q, 4 C, C₂), 140.6 (q, 2 C, C₁₀), 140.8 (q, 8 C, C₄), 141.8 (q, 4 C, C₆), 145.6 (q, 2 C, C₇).

MS (DCP, EI, 60 eV) = 703 (2, M⁺), 582 (1), 462 (1), 249 (13), 120 (41), 105 (100).

MS (MeOH + H₂O + KI, ESI, 10 eV) = 720 (100, M⁺ + OH⁻), 368 (M⁺ + 2 OH⁻).

IR (KBr) = 3443 (w), 3061 (w), 2957 (m), 2918 (m), 2856 (w), 2729 (w), 1725 (w), 1606 (s), 1592 (s), 1545 (w), 1446 (m), 1421 (m), 1402 (m), 1357 (w), 1278 (w), 1238 (s), 1214 (s), 1172 (s), 1154 (m), 1101 (w), 1030 (w), 979 (m), 874 (m), 846 (s), 818 (w), 798 (m), 744 (w), 724 (m), 663 (w), 586 (w), 562 (w), 515 (w).

UV-Vis: 386 nm (ε = 65 984), 406 nm (ε = 65 378).

quantum yield: Φ = 0.74 ± 0.10.

15.5.3. Crazy

Output for these options:

```

1 format=\color{red}\itshape,
2 list=true,
3 delta=\textcolor{green}{\ch{M+ + H2O}},
4 pos-number=side,
5 coupling-unit=\mega\gram\per\square\second,
6 list-setup=,
7 use-equal

```

yield: 17 mg yellow needles (0.04 mmol, 13 %).

mp. = 277 °C (DSC).

¹H-NMR (600 MHz, CDCl₃): δ $M^+ + H_2O$ = 2.01 (s, 24 H, H-5), 2.31 (s, 12 H, H-1), 6.72–6.74 (m, 2 H, H-11), 6.82 (s, 8 H, H-3), 7.05–7.07 (m, 2 H, H-12), 7.39–7.41 (m, 4 H, H-9), 7.48–7.49 (m, 4 H, H-8).

¹³C-NMR (150 MHz, CDCl₃): δ $M^+ + H_2O$ = 21.2 (+, 4 C, C-1), 23.4 (+, 8 C, C-5), 126.0 (+, 4 C, C-9), 128.2 (+, 8 C, C-3), 130.8 (+, 2 C, C-12), 133.6 (+, 2 C, C-11), 137.0 (+, 4 C, C-8), 138.6 (q, 4 C, C-2), 140.6 (q, 2 C, C-10), 140.8 (q, 8 C, C-4), 141.8 (q, 4 C, C-6), 145.6 (q, 2 C, C-7).

MS (DCP, EI, 60 eV) = 703 (2, M⁺), 582 (1), 462 (1), 249 (13), 120 (41), 105 (100).

MS (MeOH + H₂O + KI, ESI, 10 eV) = 720 (100, M⁺ + OH⁻), 368 (M⁺ + 2 OH⁻).

IR (KBr) = 3443 (w), 3061 (w), 2957 (m), 2918 (m), 2856 (w), 2729 (w), 1725 (w), 1606 (s), 1592 (s), 1545 (w), 1446 (m), 1421 (m), 1402 (m), 1357 (w), 1278 (w), 1238 (s), 1214 (s), 1172 (s), 1154 (m), 1101 (w), 1030 (w), 979 (m), 874 (m), 846 (s), 818 (w), 798 (m), 744 (w), 724 (m), 663 (w), 586 (w), 562 (w), 515 (w).

UV-Vis: 386 nm (ϵ = 65 984), 406 nm (ϵ = 65 378).

quantum yield: Φ = 0.74 ± 0.10.

16. Reaction Environments

16.1. Defined by CHEMMACROS

You can use these environments for numbered...

`\begin{reaction}`

A single reaction where **CHEMFORMULA** code is placed directly in the environment body. A wrapper around the equation environment.

`\begin{reactions}`

Several aligned reactions. A wrapper around amsmath's align environment.

...and their starred versions for unnumbered reactions.

`\begin{reaction*}`

A wrapper around the `equation*` environment.

`\begin{reactions*}`

A wrapper around `amsmath`'s `align*` environment.

With them you can create (un)numbered reaction equations similar to mathematical equations.

These environments use the `equation/equation*` environments or the `align/align*` environments, respectively, to display the reactions.

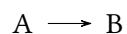
1 Reaction with counter:

2 `\begin{reaction}`

3 `A -> B`

4 `\end{reaction}`

Reaction with counter:



{1}

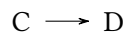
1 Reaction without counter:

2 `\begin{reaction*}`

3 `C -> D`

4 `\end{reaction*}`

Reaction without counter:



1 Several aligned reactions with counter:

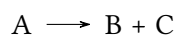
2 `\begin{reactions}`

3 `A &-> B + C \\\`

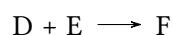
4 `D + E &-> F`

5 `\end{reactions}`

Several aligned reactions with counter:



{2}



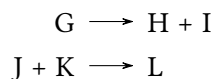
{3}


```

1 Several aligned reactions without counter:
2 \begin{reactions*}
3   G      &-> H + I \\
4   J + K &-> L
5 \end{reactions*}

```

Several aligned reactions without counter:



If you want to change the layout of the counter tags, you can use

`\renewtagform{<tagname>}[<format>]{<right delimiter>}{<left delimiter>}`.¹⁴

```

1 \renewtagform{reaction}[R \textbf{}]{[]{} }
2 \begin{reaction}
3   H2O + CO2 <=> H2CO3
4 \end{reaction}

```



With version 3.3 referencing and the use of \mathcal{M} Smith's `\intertext` also function properly:

```

1 \begin{reactions}
2   A + 2 B &-> 3 C + D \label{rxn:test}
3   \intertext{Some text in between aligned reactions}
4   3 E + F &=> G + 1/2 H
5 \end{reactions}
6 See reaction~\ref{rxn:test}.

```



Some text in between aligned reactions



¹⁴. Provided by the mathtools package

See reaction 5.

You should not use `\mch` and its relatives inside the reaction environments.

16.2. Own Reactions

You can create new types of reactions with the command:

Introduced in version 4.3 `\NewChemReaction` [*options*] {*name*} {*math name*}
name will be the name of the new environment. *math name* is the underlying math environment. Gives an error if *name* already exists.

Changed in version 4.3 `\DeclareChemReaction` [*options*] {*name*} {*math name*}
name will be the name of the new environment. *math name* is the underlying math environment.

Introduced in version 4.3 `\RenewChemReaction` [*options*] {*name*} {*math name*}
 Renew an existing definition.

The command has two options.

`star = true|false`

Also create a starred variant.

`arg = true|false`

Add a mandatory argument to the defined environment.

There is `star`, which will also define a starred version of the new environment, if the starred math environment exists. If it doesn't exist, this will cause an error.

Then there is `arg`, which is used to define an environment with a mandatory argument. Of course this only works, if the used math environment has a mandatory argument.

The predefined environments are defined via

```
1 \NewChemReaction[star]{reaction}{equation}
2 \NewChemReaction[star]{reactions}{align}.
```

Let's suppose, you'd like to have the alignment behaviour of the `alignat` environment for `CHEMFORMULA` reactions. You could do the following:

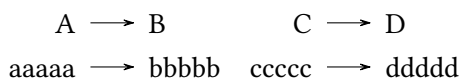
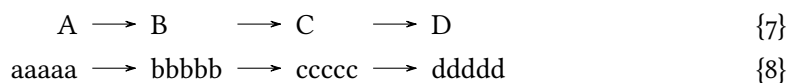
```
1 \NewChemReaction[star,arg]{reactionsat}{alignat}
```

With this the `reactionsat` environment is defined.

```

1 \NewChemReaction[star,arg]{reactionsat}{alignat}
2 \begin{reactionsat}{3}
3   A      &-> B      &&-> C      &&-> D \\
4   aaaaa &-> bbbbb &&-> ccccc &&-> ddddd
5 \end{reactionsat}
6 \begin{reactionsat*}{2}
7   A      &-> B      & C      &-> D \\
8   aaaaa &-> bbbbb &\quad{} &-> ddddd
9 \end{reactionsat*}

```



16.3. List of Reactions

CHEMMACROS also provides a command to display a list of the reactions created with the reaction environment.

`\listofreactions`

Print a list of reactions.

```

1 \listofreactions

```

List of reactions

Reaction {1}	40
Reaction {2}	40
Reaction {3}	40
Reaction [R 4]	41
Reaction {5}	41
Reaction {6}	41
Reaction {7}	43
Reaction {8}	43
Reaction {9}: Autoprotolyse	44

Reaction {10}: first step of chain	44
Reaction {11}: second step of chain	44

The output of this list can be modified by two options:

`reaction` » `list-name` = {<name of the list>} Default: List of reactions

Let's you set the name of the list manually. The default name is language dependent, see section 6.

`reaction` » `list-entry` = {<prefix to each entry>} Default: Reaction

Let's you set a prefix to each list entry. The default name is language dependent, see section 6.

Instead of using the option `list-name` you also could redefine `\reactionlistname`.

The list lists all reactions with a number and disregards reactions without number. All reaction environments without star have an optional argument which let's you add a description (or caption) for the entry in the list.

```

1 \begin{reaction}[Autoprotolyse]
2   2 H2O <=> H3O+ + OH-
3 \end{reaction}

```



If you use the reactions environment this will not work, though. In this case you can use

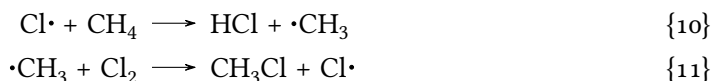
`\AddRxnDesc`{<description>}

Add a description to a reaction.

```

1 \begin{reactions}
2   Cl "\Lewis{0.,\vphantom{Cl}}" + CH4 &
3     -> HCl + "\Lewis{4.,\vphantom{CH}}" CH3 \AddRxnDesc{first-step-of-chain} \&
4     "\Lewis{4.,\vphantom{CH}}" CH3 + Cl2 &
5     -> CH3Cl + Cl "\Lewis{0.,\vphantom{Cl}}" \AddRxnDesc{second-step-of-chain}
6 \end{reactions}

```



Note: you don't have to use the phantom commands if you haven't changed the format of the atoms (see the documentation of the `CHEMFORMULA` package for information on how to do this).

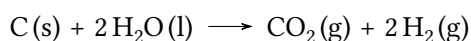
17. Phases

17.1. Basics

These commands are intended to indicate the phase of a compound.

`\sld` (s) `\lqd` (l) `\gas` (g) `\aq` (aq)

```
1 \ch{C\sld{}} + 2 H2O\lqd{} -> CO2\gas{} + 2 H2\gas{}\par
2 To make it complete: NaCl\aq.
```



To make it complete: NaCl(aq).

The IUPAC recommendation to indicate the state of aggregation is to put it in parentheses after the compound [Coh+08]. However, you might want to put it as a subscript which is also very common.

The [...] symbols are used to represent the states of aggregation of chemical species. The letters are appended to the formula in parentheses and should be printed in Roman (upright) type without a full stop (period). *IUPAC Green Book [Coh+08, p. 54]*

There are two options to customize the output:

`phases` » `pos = side|sub`

Default: side

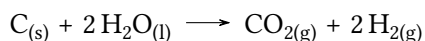
Switch the position of the phase indicator.

`phases` » `space = {<dim>}`

Default: .1333em

Change the default spacing between compound a phase indicator if `pos = {side}`. A $\text{T}_{\text{E}}\text{X}$ dimension.

```
1 \chemsetup{phases}{pos=sub}
2 \ch{C\sld{}} + 2 H2O\lqd{} -> CO2\gas{} + 2 H2\gas{}\par
3 To make it complete: NaCl\aq.
```



To make it complete: NaCl_(aq).

17.2. Define Own Phases

Depending on the subject of your document you might need to indicate other states of aggregation. You can easily define them.

Introduced in
version 4.3

`\NewChemPhase{⟨cs⟩}[⟨german⟩]{⟨english⟩}`

Define a new phase command. Actually the optional argument is an artefact of an earlier implementation of the command. It has no effect at all. See section 17.3 for a way to define language dependent settings. Gives an error if `⟨cs⟩` already exists.

`\DeclareChemPhase{⟨cs⟩}[⟨german⟩]{⟨english⟩}`

Define a new phase command. Actually the optional argument is an artefact of an earlier implementation of the command. It has no effect at all. See section 17.3 for a way to define language dependent settings.

`\RenewChemPhase{⟨cs⟩}[⟨german⟩]{⟨english⟩}`

Redefine an existing phase command. Actually the optional argument is an artefact of an earlier implementation of the command. It has no effect at all. See section 17.3 for a way to define language dependent settings.

`\phase{⟨phase⟩}`

If you need a phase indicator just once or twice.

`\NewChemPhase` only defines a phase if `⟨cs⟩` is not already used by any other command. If it is already used `CHEMMACROS` will either give an error `\RenewChemPhase` only defines a phase if `⟨cs⟩` is already used and issues an error otherwise. `\DeclareChemPhase` does not check if the phase command already exists but simply defines it with the new meaning. Unlike the other declaration commands of `CHEMMACROS` `\NewChemPhase`, `\RenewChemPhase` and `\DeclareChemPhase` can only be used in the preamble.

```
1 % preamble:
2 % \NewChemPhase{\aqi}{aq,$\infty$}% aqueous solution at infinite dilution
3 % \NewChemPhase{\cd}{cd}% condensed phase
4 % \NewChemPhase{\lc}{lc}% liquid crystal
5 NaOH\aqi\ \ch{H2O\cd} U\phase{cr} A\lc \par
6 \chemsetup[phases]{pos=sub}
7 NaOH\aqi\ \ch{H2O\cd} U\phase{cr} A\lc
```

NaOH(aq,∞) H₂O(cd) U(cr) A(lc)

NaOH_(aq,∞) H₂O_(cd) U_(cr) A_(lc)

17.3. Language Dependencies

For each phase command a translation into the custom language can be defined. If a phase is declared with `\NewChemPhase` no translation exists and for every babel language the literal string is used that was provided as a definition. Let's say you define the phase

```
1 \NewChemPhase\liquid{l}
```

and want to add the German translation “fl”. Then you could do

```
1 \DeclareTranslation{German}{phase-liquid}{f{l}}
```

This way, when you use it in a German document using the appropriate babel option using `\liquid` would correctly translate. For this the package translations is used. The ID always is `phase-⟨cname⟩` where `⟨cname⟩` is the name of the phase command you defined without leading backslash.

See section 6 for predefined translations.

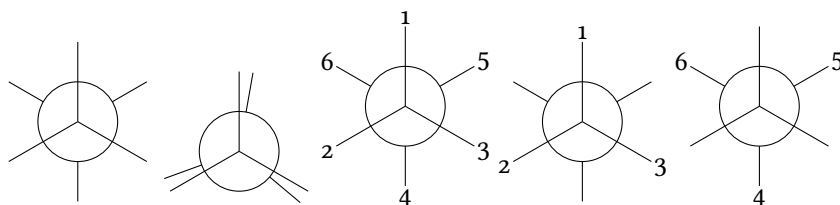
18. Newman Projections

CHEMMACROS provides a command to draw Newman projections.

```
\newman[⟨options⟩](⟨angle⟩){⟨1⟩,⟨2⟩,⟨3⟩,⟨4⟩,⟨5⟩,⟨6⟩}
```

Create Newman projections. This command uses TikZ internally. `⟨angle⟩` rotates the back atoms counter clockwise with respect to the front atoms and is an optional argument. `⟨1⟩` to `⟨6⟩` are the positions, the first three are the front atoms, the last three the back atoms.

```
1 \newman{} \newman(170){}
2 \newman{1,2,3,4,5,6} \newman{1,2,3} \newman{,,4,5,6}
```



Several options allow customization:

`newman` » `angle = {⟨angle⟩}` Default: 0
 Default angle.

`newman` » `scale = {⟨factor⟩}` Default: 1
 Scale the whole projection by factor `⟨factor⟩`.

`newman` » `ring = {⟨tikz⟩}` (initially empty)
 Customize the ring with TikZ keys.

18. Newman Projections

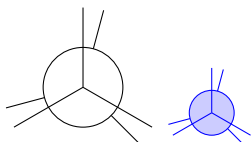
`newman` » `atoms = {\tikz}` (initially empty)

Customize the nodes within which the atoms are set with TikZ keys.

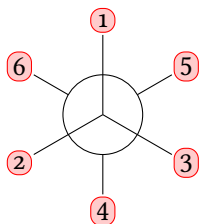
`newman` » `back-atoms = {\tikz}` (initially empty)

Explicitly customize the nodes of the back atoms with TikZ keys.

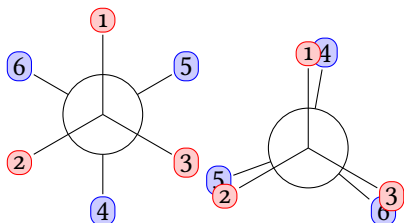
```
1 \chemsetup[newman]{angle=45} \newman{}
2 \newman[scale=.75,ring={draw=blue,fill=blue!20}]{}
```



```
1 \chemsetup[newman]{atoms={draw=red,fill=red!20,inner sep=2pt,rounded corners}}
2 \newman{1,2,3,4,5,6}
```



```
1 \chemsetup[newman]{
2   atoms = {draw=red,fill=red!20,inner sep=2pt,rounded corners},
3   back-atoms = {draw=blue,fill=blue!20,inner sep=2pt,rounded corners}
4 }
5 \newman{1,2,3,4,5,6} \newman(170){1,2,3,4,5,6}
```



19. s, p, and Hybrid Orbitals

CHEMMACROS provides the following command to create orbitals:

`\orbital[⟨options⟩]{⟨type⟩}`

Draw an orbital shape of type $\langle type \rangle$. This command uses TikZ internally.

There are the following types available for $\langle type \rangle$:

s p sp sp2 sp3

```
1 \orbital{s} \orbital{p} \orbital{sp} \orbital{sp2} \orbital{sp3}
```



Depending on the type you have different options to modify the orbitals:

`orbital` » `phase = ±| -` Default: +
changes the phase of the orbital (all types)

`orbital` » `scale = {⟨factor⟩}` Default: 1
changes the size of the orbital (all types)

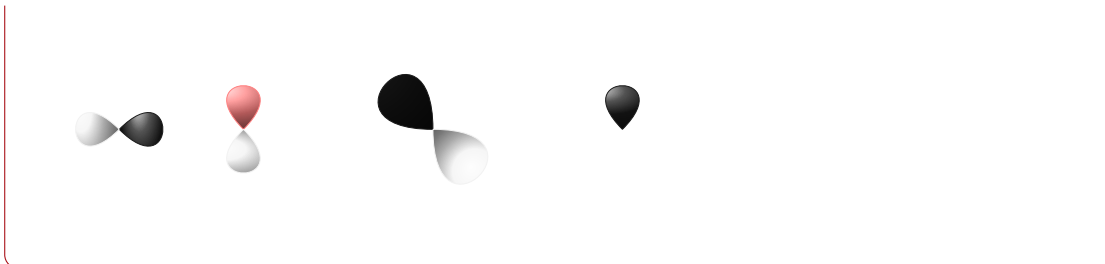
`orbital` » `color = {⟨color⟩}` Default: black
changes the color of the orbital (all types)

`orbital` » `angle = {⟨angle⟩}` Default: 0
rotates the orbitals with a p contribution counter clockwise (all types except s)

`orbital` » `half = true| false` Default: false
displays only half an orbital (only p)

```
1 \orbital{s} \orbital[phase=-]{s}
2 \orbital{p} \orbital[phase=-]{p}
3 \orbital{sp3} \orbital[phase=-]{sp3}
4
5 \orbital[angle=0]{p} \orbital[color=red!50]{p}
6 \orbital[angle=135,scale=1.5]{p} \orbital[half]{p}
```





Additionally there are two options, with which the TikZ behaviour can be changed.

`orbital` » `overlay = true|false`

The orbital “doesn’t need space”; it is displayed with the TikZ option `overlay`.

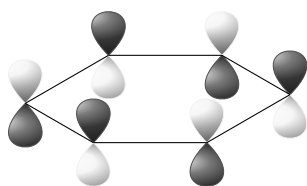
`orbital` » `opacity = {⟨num⟩}`

The orbital becomes transparent; `⟨value⟩` can have values between 1 (fully opaque) to 0 (invisible).

```

1 \vspace{7mm}
2 \chemsetup[orbital]{
3   overlay,
4   p/color = black!70
5 }
6 \setbondoffset{0pt}
7 \chemfig{
8   ?\orbital{p}
9   -[,1.3]{\orbital[phase=-]{p}}
10  -[:30,1.1]\orbital{p}
11  -[:150,.9]{\orbital[phase=-]{p}}
12  -[4,1.3]\orbital{p}
13  -[: -150,1.1]{\orbital[phase=-]{p}}?
14 }
15 \vspace{7mm}

```



```

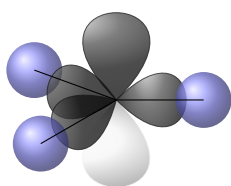
1 \vspace{7mm}
2 \setbondoffset{0pt}
3 \chemsetup[orbital]{
4   overlay ,
5   opacity = .75 ,
6   p/scale = 1.6 ,

```

```

7   s/color = blue!50 ,
8   s/scale = 1.6
9 }
10 \chemfig{
11   \orbital{s}
12   -[: -20]{\orbital[scale=2]{p}}
13           {\orbital[half,angle=0]{p}}
14           {\orbital[angle=170,half]{p}}
15           {\orbital[angle=-150,half]{p}}
16   (-[: -150]\orbital{s})-\orbital{s}
17 }
18 \vspace{1cm}

```



Part III.

Other Packages of the Bundle

The other three packages, **CHEMFORMULA**, **GHSYSTEM** and **CHEMGREEK**, all have their own documentation (if you click on the name the corresponding file will open):

- [chemformula_en.pdf](#)
- [ghsystem_en.pdf](#)
- [chemgreek_en.pdf](#)

Introduced in
version 4.0

This change has been made with version 4.0 since with this version every of those packages can be used independently from the **CHEMMACROS** package. It made sense to give each of them an own documentation file. You should be able to find them in the same folder as this document as well as via the `texdoc` program. You can also try and click on the names in the list above. They are links and *should* open the respective file.

Part IV.

Appendix

Suggestions and Bug Reports

Feedback on **CHEMMACROS**, **CHEMFORMULA**, **CHEMGREEK**, and **GHSYSTEM** is highly appreciated and welcome!

If you have suggestions for macros, missing features *etc.*, please don't hesitate to contact me. If you recognize any errors, be it chemical ones, wrong documentation and the like, I'd be grateful about a short email.¹⁵

If you find any bugs, it would be best, if you'd send me a minimal example, with which I can reproduce the bug. You can also submit an issue on <https://bitbucket.org/cgnieder/chemmacros/> instead.

Many thanks to all the people who already provided me with feedback, especially (in alphabetical order):

- Peter Cao
- Christina Lüdigg
- Dr. Paul King
- Jonas Rivetti (Special thanks for his translation of the hazard and precautionary statements into Italian!)
- Christoph Schäfer
- Timo Stein

References

- [Bov05] Michel Bovani. *fourier*. version 1.3, Jan. 30, 2005.
URL: <http://mirror.ctan.org/fonts/fourier-gut/>.
- [Bra13] Johannes Braams, current maintainer: Javier Bezos.
babel. version 3.9f, May 16, 2013.
URL: <http://mirror.ctan.org/macros/latex/required/babel/>.
- [Cai10] Christophe Caignaert. *kpfonts*. version 3.31, Aug. 20, 2010.
URL: <http://mirror.ctan.org/fonts/kpfonts>.
- [CH09] David Carlisle and Morten Høgholm. *xspace*. version 1.13, Oct. 20, 2009.
URL: <http://mirror.ctan.org/macros/latex/required/tools/>.

¹⁵. contact@mychemistry.eu

References

- [Cha13] François Charette, current maintainer: Arthur Reutenauer.
polyglossia. version 1.33.4, June 27, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/polyglossia/>.
- [CM04] David Carlisle and Frank Mittelbach. bm. version 1.1c, Feb. 26, 2004.
URL: <http://mirror.ctan.org/macros/latex/required/bm/>.
- [Coh+08] E. Richard Cohan et al.
“*Quantities, Symbols and Units in Physical Chemistry*”, *IUPAC Green Book*.
3rd Edition. 2nd Printing. IUPAC & RSC Publishing, Cambridge, 2008.
- [Con+05] Neil G. Connelly et al. “*Nomenclature of Inorganic Chemistry*”, *IUPAC Red Book*.
IUPAC & RSC Publishing, Cambridge, 2005. ISBN: 0-85404-438-8.
- [Mic11] Leonard Michlmayr. textgreek. version 0.7, Oct. 9, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/textgreek/>.
- [Mil13] Günter Milde. greek-fontenc. version 0.11.1, Dec. 1, 2013.
URL: <http://mirror.ctan.org/language/greek/greek-fontenc/>.
- [MRW13] Lars Madsen, Will Robertson, and Joseph Wright.
mathtools. version 1.13, Feb. 12, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/mh/>.
- [Nie13] Clemens Niederberger. translations. version 1.1a, Sept. 30, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/translations/>.
- [Pic13] Paul Pichaureau. mathdesign. version 2.31, Aug. 29, 2013.
URL: <http://mirror.ctan.org/fonts/mathdesign/>.
- [PPR04] R. Panico, W. H. Powell, and J-C. Richer. “*Nomenclature of Organic Chemistry, Sections A, B, C, D, E, F, and H*”, *IUPAC Blue Book*. DRAFT. Oct. 7, 2004.
URL: <http://old.iupac.org/reports/provisional/abstract04/BB-prs310305/CompleteDraft.pdf> (visited on 07/07/2013).
- [Rei98] Axel Reichert. nicefrac. version 0.9b, Aug. 4, 1998.
URL: <http://mirror.ctan.org/macros/latex/contrib/units/>.
- [Rob13] Will Robertson. environ. version 0.3, Apr. 1, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/environ/>.
- [Scho3] Walter Schmidt. upgreek. version 2.0, Feb. 12, 2003.
URL: <http://mirror.ctan.org/macros/latex/contrib/was/>.
- [Sha13] Michael Sharpe. newtx. version 1.151, Oct. 17, 2013.
URL: <http://mirror.ctan.org/fonts/newtx/>.
- [Tan10] Till Tantau. TikZ/pgf. version 2.10, Oct. 25, 2010.
URL: <http://mirror.ctan.org/graphics/pgf/>.
- [Theo8] The European Parliament and The Council of the European Union. *Regulation (EC) No 1272/2008 of the European Parliament and of the Council. on classification, labelling and packaging of substances and mixtures, amending and repealing Directives 67/548/EEC and 1999/45/EC, and amending Regulation (EC) No 1907/2006*. Dec. 16, 2008.

References

- [The13a] The L^AT_EX3 Project Team. l3kernel. version SVN 4582, July 28, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/l3kernel/>.
- [The13b] The L^AT_EX3 Project Team. l3packages. version SVN 4582, July 28, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/l3packages/>.
- [Uni12] United Nations Economic Commission for Europe. *GHS Implementation*.
Mar. 20, 2012. URL:
http://www.unece.org/trans/danger/publi/ghs/implementation_e.html
(visited on 03/20/2012).
- [Wri13] Joseph Wright. siunitx. version 2.5s, July 31, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/siunitx>.

Index

Symbols

<code>\-</code>	12 f.
<code>\#</code>	34
<code>\^</code>	12 f.
<code>\ </code>	12 f.
"Nomenclature of Organic Chemistry, Sections A, B, C, D, E, F, and H", IUPAC Blue Book	16

A

<code>\a</code>	14
<code>\abinitio</code>	18
<code>\AddRxnDesc</code>	44
<code>align</code>	23
<code>alignat</code> (environment)	42
<code>amsmath</code> (package)	39 f.
<code>angle</code>	47, 49
<code>\anti</code>	16, 18
<code>append</code>	4
<code>\aq</code>	45
<code>\aqi</code>	46
<code>arg</code>	42
<code>\atm</code>	19
<code>\atmosphere</code>	19
<code>atoms</code>	48

B

<code>\b</code>	14
<code>\ba</code>	5, 10 f.
<code>babel</code> (package)	7 f., 46 f.
<code>back-atoms</code>	48
Bezos, Javier	7
<code>bm</code> (package)	3
Bovani, Michel	11
<code>bpchem</code>	4
<code>bpchem</code> (package)	12
Braams, Johannes	7
<code>break-space</code>	13
<code>\bridge</code>	17
<code>bridge-number</code>	17

C

Caignaert, Christophe	11
<code>\cal</code>	19
<code>\calory</code>	19
Carlisle, David	3
<code>\cd</code>	46
<code>\ch</code>	4, 9, 12, 22 ff., 28 f., 31 f., 37, 39, 45 f.
<code>\changestate</code>	10, 31
Charette, François	7
<code>\chemabove</code>	24 f.
<code>\chemAlpha</code>	10, 14

<code>\chemalpha</code>	6, 10, 14
<code>\chembeta</code>	12, 14
<code>\chemdelta</code>	14
<code>chemfig</code> (package)	24 f.
<code>chemformula</code>	7
<code>\chemgamma</code>	12, 14
<code>\chemomega</code>	14
<code>\chemsetup</code>	6 f., 11, 21, 23 f., 27 f., 37, 45 f., 48, 50
<code>chemstyle</code> (package)	10, 18 ff.
<code>\chlewis</code>	5
<code>\cip</code>	15, 17
<code>cip-kern</code>	16
<code>circled</code>	6, 12
<code>circletype</code>	6
<code>\cis</code>	16 f.
<code>\cmc</code>	19
<code>cmversion</code>	5
<code>\CNMR</code>	34
Cohan, E. Richard	11, 15, 20, 45
<code>color</code>	49
Connelly, Neil G.	17
<code>cool</code> (package)	13
<code>coord-use-hyphen</code>	17
<code>coupling-bonds-post</code>	36
<code>coupling-bonds-pre</code>	36
<code>coupling-nuclei-post</code>	35
<code>coupling-nuclei-pre</code>	35
<code>coupling-pos</code>	35
<code>coupling-symbol</code>	35
<code>coupling-unit</code>	35
CTAN	3

D

<code>\D</code>	13 f., 16
<code>\d</code>	14
<code>\data</code>	34–37
<code>decimal-marker</code>	23
<code>\DeclareChemIUPAC</code>	18
<code>\DeclareChemLatin</code>	19
<code>\DeclareChemNMR</code>	34
<code>\DeclareChemParticle</code>	12
<code>\DeclareChemPhase</code>	46
<code>\DeclareChemReaction</code>	42
<code>\DeclareChemState</code>	5, 31
<code>\delm</code>	24
<code>\delp</code>	24
<code>\Delta</code>	30
<code>delta</code>	30 f., 33, 36
<code>\dent</code>	17
<code>\dento</code>	17
<code>\dexter</code>	16

INDEX

<code>\Dfi</code>	4	<code>\Ht0</code>	9
<code>dist</code>	27	<code>\Hyd</code>	9
E		<code>\hydrogen</code>	15
<code>\E</code>	13, 16	<code>hyphen-post-space</code>	13
<code>\El</code>	10	<code>hyphen-pre-space</code>	13
<code>\el</code>	7, 10, 12, 28 f.	I	
<code>elpair</code>	11	<code>\insitu</code>	18
<code>\ElPot</code>	31	<code>\intertext</code>	41
<code>\endo</code>	18	<code>\invacuo</code>	18
<code>\entgegen</code>	16	<code>iupac</code>	6, 13, 18
<code>\Enthalpy</code>	29 f., 32	<code>\iupac</code>	5, 12–15, 17 f.
<code>\enthalpy</code>	32	J	
<code>\Entropy</code>	29 f., 32	<code>\J</code>	34
<code>environ</code> (package).....	3	K	
<code>experimental</code> (environment).....	5, 34	<code>\k</code>	14
<code>explicit-sign</code>	23	<code>K-acid</code>	21
<code>exponent</code>	30 ff.	<code>K-base</code>	21
F		<code>K-water</code>	21
<code>\fac</code>	16	<code>\Ka</code>	20 f.
<code>\fdelm</code>	24	<code>\Kb</code>	20 f.
<code>\fdelp</code>	24	<code>kpfonts</code> (package).....	4, 6, 11
<code>\fmch</code>	7, 22, 25	<code>\Kw</code>	20 f.
<code>\fminus</code>	10	L	
<code>format</code>	19, 35	<code>\L</code>	13 f., 16
<code>fourier</code> (package).....	11	<code>l3kernel</code> (bundle).....	2
<code>\fpch</code>	7, 22	<code>l3packages</code> (bundle).....	2
<code>\fplus</code>	6, 10	<code>\laevus</code>	16
<code>\fscrm</code>	25	<code>language</code>	6 f., 20
<code>\fscrp</code>	25	<code>\latin</code>	18
<code>\fsscrm</code>	25	<code>\Lfi</code>	4
<code>\fsscrp</code>	25	<code>list</code>	36
G		<code>list-entry</code>	44
<code>\g</code>	14	<code>list-name</code>	44
<code>\gas</code>	45	<code>list-setup</code>	36
<code>GHS Implementation</code>	3	<code>\listofreactions</code>	43
<code>ghsystem</code>	7	<code>LPPL</code>	2
<code>ghsystem</code>	6	<code>\lqd</code>	8 f., 45
<code>\Gibbs</code>	29 f., 32	M	
<code>\gram</code>	37, 39	<code>\m</code>	14
<code>greek</code>	4, 6, 10 f., 14	<code>Madsen, Lars</code>	3
<code>greek-fontenc</code> (bundle).....	11	<code>mathdesign</code> (package).....	11
H		<code>mathspec</code> (package).....	6, 10
<code>\H</code>	15	<code>mathtools</code> (package).....	3, 41
<code>Høgholm, Morten</code>	3	<code>\mch</code>	4, 7, 22, 42
<code>half</code>	49	<code>\mech</code>	25 f.
<code>\hapto</code>	17	<code>\mega</code>	39
<code>\Helmholtz</code>	31	<code>\mer</code>	16
<code>Hensel, Martin</code>	3	<code>\meta</code>	13, 16 ff.
<code>\HNMR</code>	34	<code>method</code>	4
<code>\Hpl</code>	9		

INDEX

mhchem (package)	3, 22	parse	23, 36
\mhName	5	\pch	4, 7, 22, 26 ff.
Michlmayr, Leonard	11	Pedersen, Bjørn	12
Milde, Günter	11	\per	30 ff., 37, 39
Mittelbach, Frank	3	\pH	20 f.
\Molar	19	phase	49
\moLar	19	\phase	8, 46 f., 49 f.
\molar	19	\phosphorus	15
\MolMass	20	\photon	12
N		Pichaureau, Paul	11
\N	15	\pKa	9, 20 f.
\n	14	\pKb	20 f.
\NewChemIUPAC	18	\pOH	20
\NewChemLatin	19	polyglossia (package)	7 f.
\NewChemNMR	34, 36	pos	23, 45
\NewChemParticle	12	\pos	23, 34, 37 ff., 45 f.
\NewChemPhase	46 f.	pos-number	35
\NewChemReaction	42 f.	\positron	12
\NewChemState	30 ff.	Powell, W. H.	16
\newman	47 f.	\prt	7, 10
newtx (package)	11	Q	
newtxmath (package)	11	“Quantities, Symbols and Units in Physical Chemistry”, IUPAC Green Book	11, 15, 20, 45
nicefrac (package)	3	R	
Niederberger, Clemens	1, 3	\R	13, 15
\nitrogen	15	\Rconf	16
\NMR	33 f., 36 f.	reaction (environment)	39, 42 f.
nmr	35	reaction* (environment)	40
nmr (environment)<optionen>	36	\reactionlistname	44
“Nomenclature of Inorganic Chemistry”, IUPAC Red Book	17	reactions (environment)	39, 44
\normal	20	reactions* (environment)	40
\ntr	10	reactionsat (environment)	42
Nu	6, 10	\rectus	15
\Nu	6, 10 f.	\redox	26–29
\Nuc	10	Regulation (EC) No 1272/2008 of the European Parliament and of the Council	3
nucleus	35	Reichert, Axel	3
O		\RenewChemIUPAC	18
\O	15	\RenewChemLatin	19
opacity	50	\RenewChemNMR	34
option	7	\RenewChemParticle	12
\orbital	49 ff.	\RenewChemPhase	46
\ortho	4, 16	\RenewChemReaction	42
overlay	50	\RenewChemState	5, 31 f.
\OX	26–29	\renewtagform	41
\ox	4, 22 ff., 27 ff.	Reutenauer, Arthur	7
\oxygen	15	Richer, J-C.	16
P		ring	47
\P	15	Robertson, Will	3
\p	5, 20	roman	23
p-style	21	S	
Panico, R.	16	\S	13, 15
\para	16		

INDEX

<code>scale</code>	47, 49	The European Parliament and The Council of the European Union	3
Schmidt, Walter	11	<code>tikz</code> (package)	3
<code>\Sconf</code>	16	TikZ/pgf (package)	3
<code>\scrm</code>	25	<code>\torr</code>	20
<code>\scrp</code>	25	<code>\trans</code>	13, 16
<code>\second</code>	39, 44	<code>\transitionstatesymbol</code>	10
<code>sep</code>	27	translations (package)	3 f., 7 f., 47
<code>\Sf</code>	15	U	
Sharpe, Michael	11	<code>unit</code>	30, 35
<code>\sin</code>	9	United Nations Economic Commission for Europe . . .	3
<code>\sinister</code>	15	upgreek (package)	6, 11, 14
siunitx (package)	3, 19 f., 29 f., 33, 35	<code>use-equal</code>	34, 36
<code>\sld</code>	8 f., 45	V	
<code>space</code>	45	<code>\val</code>	35, 37
<code>\standardstate</code>	10, 31	<code>version</code>	7
<code>star</code>	42	W	
<code>\State</code>	32	<code>\w</code>	14
<code>strict</code>	5	<code>\water</code>	9
<code>subscript</code>	30 f.	Wright, Joseph	3
<code>subscript-left</code>	31, 33	X	
<code>\sulfur</code>	15	<code>xfrac</code> (package)	24
<code>\syn</code>	16	<code>xspace</code>	3, 6
<code>synchronize</code>	6	<code>xspace</code> (package)	3
T		Z	
Tantau, Till	3	<code>\Z</code>	16
<code>\ter</code>	9	<code>\zusammen</code>	16
<code>\tert</code>	16		
textalpha (package)	11		
textgreek (package)	11, 14		
The L ^A T _E X ₃ Project Team	2		