THE CHEMMACROS BUNDLE

v4.2b 2013/11/04

packages CHEMMACROS (v4.2b), CHEMFORMULA (v4.2a), GHSYSTEM (v4.0) and CHEMGREEK (v0.2)

documentation for the CHEMMACROS package

Clemens NIEDERBERGER

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

contact@mychemistry.eu

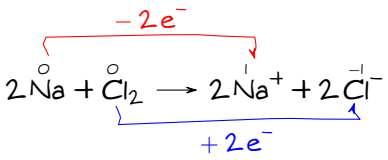


Table of Contents

Part I.
Preliminaries

1. Licence, Requirements and README

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

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

also needs the packages siunitx[1] [Wri13], mathtools[2] [MRW13], bm[3] [CM04], nicefrac[4] [Rei98] and environ[5] [Rob13] as well as tikz[6] [Tan10] and the Ti*k*Z libraries `calc` and `arrows`. Language support is done with the help of the translations[7] [Nie13]. The CHEMMACROS package also loads the other packages of this bundle.

The package option `xspace` also loads the package xspace [CH09].

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 2ε 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/`

- 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 recommendations[8] 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.

4. Package Options

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

```
1 \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.

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 » cmversion = 1|2|3|4|newest Default: newest

This option restores the old definitions of some commands and tries to ensure backwards compatibility as much as possible (default = 4). Actually 2 and 3 are aliases, as are – for now – 4 and newest. *This option can only be chosen in the preamble.*

option » ghsystem = true|false Default: true

ghsystem = false disables the automatic loading of the GHSYSTEM package.

option » greek = auto|kpfonts|math|newtx|textgreek|upgreek Default: auto

This option determines how the letters \Chemalpha and friends are typeset. See page 10 for more information. Please note that this option *does not load either upgreek, kpfonts, newtxmath nor textgreek!* It only determines which one to choose if available. The option auto will detect if either (in order of priority) upgreek, textgreek, kpfonts or newtxmath have been loaded and use them if available. If you explicitly choose upgreek, textgreek, kpfonts or newtxmath 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 12.

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 9. *This option can only be chosen in the preamble.*

option » strict = true|false Default: false
Setting strict = {⟨*true*⟩} will turn all warning messages into erros messages.

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.

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).

```
\chemsetup[option]{circled=none}
  \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
\chemsetup[option]{circled=formal}
  \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
\chemsetup[option]{circletype=math}
  \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
\chemsetup{option/circletype=chem,option/circled=all}%
  \leavevmode\mch\ \pch\ \fmch\ \fpch\ \el\ \prt \par
\chemsetup{option/circletype=math}
```

```
\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. Supported Languages

By choosing the option

`\chemsetup[`⟨*option*⟩`]{language=`⟨*language*⟩`}`
Selection of the language ⟨*language*⟩.

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.

Currently the following translations are provided:

```
% subscript used in \Ka:
\DeclareTranslation{German}{K-acid}{S}
% the phases \sld and \lqd:
\DeclareTranslation{German}{phase-sld}{f}
\DeclareTranslation{German}{phase-lqd}{f{}l}
% heading of the list of reactions:
\DeclareTranslation{English}{list-of-reactions}{List of reactions}
\DeclareTranslation{German} {list-of-reactions}{Reaktionsverzeichnis}
\DeclareTranslation{Italian}{list-of-reactions}{Elenco delle reazioni}
\DeclareTranslation{French} {list-of-reactions}{Table des r\'eactions}
% name at the beginning of each entry in the list of reactions:
\DeclareTranslation{English}{lor-reaction}{Reaction }
\DeclareTranslation{German} {lor-reaction}{Reaktion }
\DeclareTranslation{Italian}{lor-reaction}{Reazione }
```

```
\DeclareTranslation{French} {lor-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.

6.2. Specialties

6.2.1. German

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

6.2.2. 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`
$\doteq$

`\standardstate`
$\circ$

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°. 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.

9. many thanks to the package author Joseph Wright.

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

particle » elpair = false|dots|dash Default: false

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

```
\ba[elpair] \Nu[elpair=dash]

\chemsetup[particle]{elpair}
\ba\ \Nu
```

ba:⁻ Nu|⁻
ba:⁻ Nu:⁻

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. 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 12.

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:

\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 vaild CHEMFORMULA compound. Please have a look at the CHEMFORMULA manual for details. The particle will obey the circled option.

```
\DeclareChemParticle\positron{\Chembeta+}
\DeclareChemParticle\photon{\Chemgamma}
\RenewChemParticle\el{\Chembeta-}
\positron\ \photon\ \el
```

β^+ γ β^-

\DeclareChemParticle only defines a particle if ⟨*cs*⟩ is not already used by any other command. If it *is* already used CHEMMACROS will either give a warning or an error, depending on the option strict. \RenewChemParticle *only* defines a particle if ⟨*cs*⟩ *is* already used and issues a warning/error otherwise.

8. Nomenclature, Stereo Descriptors, Latin Phrases

8.1. IUPAC Names

Similar to the bpchem package CHEMMACROS provides a command[10] 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.

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

Tetracyclo[2.2.2.$1^{1,4}$]-undecane-2-dodecyl-5-(heptadecylisododecyl-thioester)

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.

```
\huge\iupac{2,4\-Di\|chlor\|pentan} \par
2,4-Dichlorpentan
```

2,4-Dichlorpentan
2,4-Dichlorpentan

The spaces inserted by these commands can be customized.

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

`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. This means they either are predefined already (`\L` Ł) 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 1 demonstrates the different modes.

	auto	restricted	strict
`\L`	Ł	Ł	L
`\iupac{\L}`	L	L	L
`\D`	D	–	D
`\iupac{\D}`	D	D	D

Table 1: Demonstration of `iupac`'s modes.

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 them[11] the following commands typeset upright Greek letters:

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

\a α \b β \g γ \d δ \k κ \m μ \n η \w ω

```
\iupac{5\a\-androstan\-3\b\-ol} \par
\iupac{\a\-(tri\|chloro\|methyl)\-\w\-chloro\|poly(1,4\-phenylene\|methylene)}
```

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

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

\H *H* \O *O* \N *N* \Sf *S* \P *P*

```
\iupac{\N\-methyl\|benz\|amide} \par
\iupac{3\H\-pyrrole} \par
\iupac{\O\-ethyl hexanethioate}
```

N-methylbenzamide
3*H*-pyrrole
O-ethyl hexanethioate

Cahn-Ingold-Prelog

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

`\R` (*R*)
Typeset recto descriptor.

`\S` (*S*)
Typeset sinister descriptor.

Since the command `\S` has another meaning already (§) it is only available inside `\iupac` in the default setting.

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

\D D \L L

Since the command `\L` has another meaning already (Ł) it is only available inside `\iupac` in the default setting.

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

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

The package cool defines the commands \E and \D, too. If you load it, the CHEMMACROS version will only be available inside \iupac in the default setting.

ortho/meta/para

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

Although these commands are provided I like to cite [PPR04]:

> 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 [PPR04, p. 90]*

Absolute Configuration (uses TikZ)

\Rconf[⟨*letter*⟩]
\Rconf: (R) \Rconf[]: (↻)

\Sconf[⟨*letter*⟩]
\Sconf: (S) \Sconf[]: (↺)

Examples:

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

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

Coordination Chemistry CHEMMACROS provides two commands useful with coordination chemistry:

`\bridge{⟨num⟩}` μ_3 `\hapto{⟨num⟩}` η^5

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

Ferrocene = bis(η^5cyclopentadienyl)iron
tetra-$\mu_3$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` and `\bridge` or don't.

8.1.2. Own Naming Commands

If you find any commands missing you can define them using

`\DeclareChemIUPAC{⟨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.

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, `\DeclareChemIUPAC` will *not* change the definition of an existing IUPAC naming command but issue a warning/an error (depending on the package option `strict`) if the IUPAC naming command already exists.

```
\DeclareChemIUPAC\endo{\textit{endo}}
\RenewChemIUPAC\anti{\textit{anti}}
\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.

```
\iupac{\meta\-Xylol} \par
\RenewChemIUPAC\meta{\textup{m}}
\iupac{\meta\-Xylol}
```

m-Xylol
m-Xylol

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

`\DeclareChemLatin{`⟨*cs*⟩`}{`⟨*phrase*⟩`}`
Define a new latin phrase.

`\RenewChemLatin{`⟨*cs*⟩`}{`⟨*phrase*⟩`}`
Redefine an existing latin phrase.

```
\DeclareChemLatin\ltn{latin text}\ltn
```

latin text

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

`latin » format = {`⟨*definition*⟩`}` Default: `\itshape`
Set the format of the latin phrases.

9. Units for the Usage With siunitx

In chemistry some non-SI units are very common. siunitx provides the command `\DeclareSIUnit{`⟨*command*⟩`}{`⟨*uni*
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`
cm^3

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, pK_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`: pK_A, `\pKa[1]`: pK_{A1}

`\pKb[`⟨*num*⟩`]`
`\pKb`: $\mathrm{p}K_B$, `\pKb[1]`: $\mathrm{p}K_{B1}$

`\p{`⟨*anything*⟩`}`
e. g. `\p{\Kw}` $\mathrm{p}K_W$

```
\Ka \Kb \pKa \pKa[1] \pKb \pKb[1]
```
$K_A\ K_B\ \mathrm{p}K_A\ \mathrm{p}K_{A1}\ \mathrm{p}K_B\ \mathrm{p}K_{B1}$

> 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.

```
\pH, \pKa \par
\chemsetup[acid-base]{p-style=slanted} \pH, \pKa \par
\chemsetup[acid-base]{p-style=italics} \pH, \pKa
```

---

pH, $\mathrm{p}K_A$
*p*H, pK_A
*p*H, pK_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)

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

$^{+}$, Na^{+}, Ca^{2+}
$^{-}$, F^{-}, S^{2-}

The same for formal charges:

`\fpch[`⟨*number*⟩`]`
positive charge

`\fmch[`⟨*number*⟩`]`
negative charge

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

⊕ ⊖ 3⊕ 3⊖

11.2. Oxidation Numbers

Typesetting oxidation numbers:

`\ox[`⟨*options*⟩`]{`⟨*number*⟩`,`⟨*atom*⟩`}`
Places ⟨*number*⟩ above ⟨*atom*⟩; ⟨*number*⟩ has to be a (rational) number!

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

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

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

`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 $\overset{1.2}{\mathrm{X}}$.

`ox` » `align` = `center|right` Default: `center`

Center the oxidation number relative to the atom or right-align it.

```
\ox[roman=false]{2,Ca} \ox{2,Ca} \\
\ox[pos=super]{3,Fe}-Oxide \\
\ox[pos=side]{3,Fe}-Oxide \\
\ox[parse=false]{?,Mn} \\
\ox[align=right]{2,Ca}
```

$\overset{2}{\mathrm{Ca}}$ $\overset{\mathrm{II}}{\mathrm{Ca}}$
$\mathrm{Fe}^{\mathrm{III}}$-Oxide
Fe(III)-Oxide
$\overset{?}{\mathrm{Mn}}$
$\overset{\mathrm{II}}{\mathrm{Ca}}$

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

```
\ox{3,Fe} \ox*{3,Fe}
```

$\overset{\mathrm{III}}{\mathrm{Fe}}$ $\mathrm{Fe}^{\mathrm{III}}$

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

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

$\overset{+\mathrm{I}}{\mathrm{Na}}$, $\overset{+\mathrm{II}}{\mathrm{Ca}}$, $\overset{-\mathrm{II}}{\mathrm{S}}$, $\overset{\pm 0}{\mathrm{F_2}}$

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

Compare $\overset{-\mathrm{I}}{\mathrm{O_2}}^{2-}$ to $\overset{-\mathrm{I}}{\mathrm{O_2}}^{2-}$

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

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

$\overset{0.5}{\mathrm{Br_2}}$ $\overset{⅓}{\mathrm{I_3}}^{+}$

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

```
\DeclareInstance{xfrac}{chemmacros-ox-frac}{text}{
  scale-factor        = 1.2 ,
  denominator-bot-sep = -.5ex ,
  numerator-top-sep   = -.3ex ,
  slash-left-kern     = -.2em ,
```

```
  slash-right-kern    = -.2em ,
  slash-symbol-font   = lmr
}
```

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`
$\delta+$ (**del**ta + **p**lus)

`\delm`
$\delta-$ (**del**ta + **m**inus)

`\fdelp`
$\delta\oplus$

`\fdelm`
$\delta\ominus$

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

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

The following macros are useful together with chemfig, too.

`\scrp`
$^+$ (**scr**iptstyle + **p**lus)

`\scrm`
$^-$ (**scr**iptstyle + **m**inus)

`\fscrp`
$\oplus$

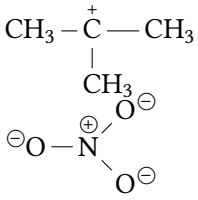
`\fscrm`
$\ominus$

`\fsscrp`
⊕ (using `\scriptscriptstyle`)

`\fsscrm`
⊖

```
\setatomsep{1.8em}\chemfig{CH_3-\chemabove{C}{\scrp}(-[6]C|H_3)-\vphantom{H_3}
  CH_3}

\chemfig{\fmch{}|O-\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 Ti*k*Z.

`\OX{`⟨*name*⟩`,`⟨*atom*⟩`}`
Label ⟨*atom*⟩ with the label ⟨*name*⟩.

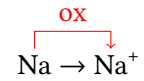
`\redox(`⟨*name1*⟩`,`⟨*name2*⟩`)[`⟨*tikz*⟩`][`⟨*num*⟩`]{`⟨*text*⟩`}`
Connect two ⟨*atom*⟩s previously labelled with `\OX`. Only the first argument `(`⟨*name1*⟩`,`⟨*name2*⟩`)` is required, the others are all optional.

`\OX` places ⟨*atom*⟩ into a node, which is named with ⟨*name*⟩. 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*.

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

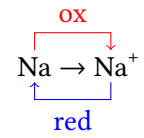
This line can be customized using Ti*k*Z keys in `[`⟨*tikz*⟩`]`:

```
\vspace{7mm}
\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.

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

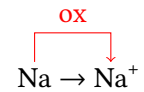


The default length of the vertical lines can be customized with the option

`redox` » `dist` = {⟨*dim*⟩} Default: `.6em`

A TEX dimension.

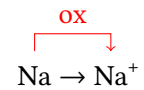
```
\vspace{7mm}
\chemsetup{redox/dist=1em}
\OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b)[->,red]{ox}
```



`redox` » `sep` = {⟨*dim*⟩} Default: `.2em`

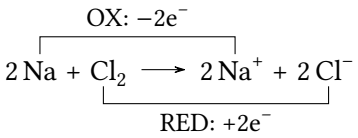
The option can be used to change the distance between the atom and the beginning of the line.

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

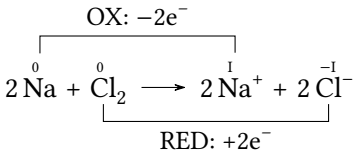


Examples:

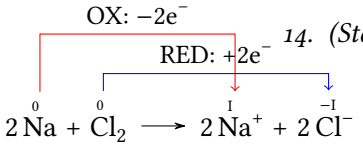
```
\vspace{7mm}
\ch{
  2 "\OX{o1,Na}" + "\OX{r1,Cl}" {}2
  ->
  2 "\OX{o2,Na}" {}+ + 2 "\OX{r2,Cl}" {}-
}
\redox(o1,o2){\small OX: $- 2\el$}
\redox(r1,r2)[][-1]{\small RED: $+ 2\el$}
\vspace{7mm}
```



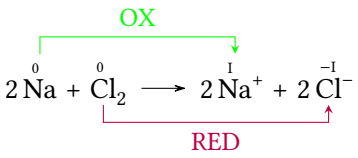

```
\vspace{7mm}
\ch{
  2 "\OX{o1,\ox{0,Na}}" + "\OX{r1,\ox{0,Cl}}" {}2
  ->
  2 "\OX{o2,\ox{+1,Na}}" {}+ + 2 "\OX{r2,\ox{-1,Cl}}" {}-
}
\redox(o1,o2){\small OX: $- 2\el$}
\redox(r1,r2)[][-1]{\small RED: $+ 2\el$}
\vspace{7mm}
```




```
\vspace{14mm}
\ch{
  2 "\OX{o1,\ox{0,Na}}" + "\OX{r1,\ox{0,Cl}}" {}2
  ->
  2 "\OX{o2,\ox{+1,Na}}" {}+ + 2 "\OX{r2,\ox{-1,Cl}}" {}-
}
\redox(o1,o2)[draw=red,->][3.33]{\small OX: $- 2\el$}
\redox(r1,r2)[draw=blue,->]{\small RED: $+ 2\el$}
```




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

```
\Enthalpy{123} \par
\Entropy{123} \par
\Gibbs{123}
```

$\Delta H^\ominus = 123\,\mathrm{kJ\,mol^{-1}}$
$S^\ominus = 123\,\mathrm{J\,K^{-1}\,mol^{-1}}$
$\Delta G^\ominus = 123\,\mathrm{kJ\,mol^{-1}}$

The argument (⟨*subscript*⟩) adds a subscript for specification: `\Enthalpy(r){123}` $\Delta_\mathrm{r} H^\ominus = 123\,\mathrm{kJ\,mol^{-1}}$.

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.

```
\Enthalpy[unit=\kilo\joule]{-285} \par
\Gibbs[delta=false]{0} \par
\Entropy[delta=\Delta,exponent=]{56.7}
```

$\Delta H^\ominus = -285\,\mathrm{kJ}$
$G^\ominus = 0\,\mathrm{kJ\,mol^{-1}}$
$\Delta S = 56.7\,\mathrm{J\,K^{-1}\,mol^{-1}}$

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

```
\Enthalpy{-1234.56e3} \par
\sisetup{
  per-mode=symbol,
  exponent-product=\cdot,
  output-decimal-marker={,},
  group-four-digits=true
}
\Enthalpy{-1234.56e3}
```

$\Delta H^\ominus = -1234.56 \times 10^3\,\mathrm{kJ\,mol^{-1}}$
$\Delta H^\ominus = -1\,234{,}56 \cdot 10^3\,\mathrm{kJ/mol}$

14.1.1. Create New Variables or Redefine Existing Ones

`\DeclareChemState{`⟨*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.

Changed in version 4.2

`\RenewChemState{`⟨*name*⟩`}[`⟨*options*⟩`]{`⟨*symbol*⟩`}{`⟨*unit*⟩`}`
Redefine existing state commands. ⟨*name*⟩ 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.

Changed in version 4.2

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

$\Delta A^{\ominus} = 123.4\,\mathrm{kJ\,mol^{-1}}$
$\Delta E = -1.1\,\mathrm{V}$
$\Delta E^{0}_{\mathrm{Sn|Sn^{2+}||Pb^{2+}|Pb}} = 0.01\,\mathrm{V}$

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. ⟨*anything*⟩ 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.

```
\RenewChemState\Enthalpy{h}{\joule}
\Enthalpy(f){12.5}
```

$\Delta_{\mathrm{f}} h^{\ominus} = 12.5\,\mathrm{J}$

The existing commands have been defined like this:

```
\DeclareChemState\Enthalpy{H}{\kilo\joule\per\mole}
\DeclareChemState\Entropy[delta=false,subscript-left=false]{S}
  {\joule\per\kelvin\per\mole}
\DeclareChemState\Gibbs{G}{\kilo\joule\per\mole}
```

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

```
\DeclareChemState\enthalpy[exponent=]{h}{\kilo\joule\per\mole}% molar
\RenewChemState\Enthalpy[exponent=]{H}{\kilo\joule}% absolute
\enthalpy{-12.3} \Enthalpy{-12.3}
```

$\Delta h = -12.3\,\mathrm{kJ\,mol^{-1}}$ $\Delta H = -12.3\,\mathrm{kJ}$

14.2. State

The commands presented in section 14.1 internally all use the command[12]

`\State[`⟨*options*⟩`]{`⟨*symbol*⟩`}{`⟨*subscript*⟩`}`
Typeset a state variable.

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

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

$\Delta A^{\ominus}$, $\Delta_\mathrm{f} G^{\ominus}$, $\Delta E^{\ominus}_\mathrm{Na}$, $\Delta H^{1000\,^\circ\mathrm{C}}$

Again there are some keys to customize the command:

`state` » `exponent` = `{`⟨*anything*⟩`}`
Set the default exponent symbol.

`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:

$$^1\text{H-NMR}\ (400\,\mathrm{MHz},\ \mathrm{CDCl_3})\colon\ \delta = 1.59$$

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:

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

```
\NMR \par
\NMR*
```
^{1}H-NMR: δ
^{1}H-NMR

The first argument specifies the kind of NMR:

```
\NMR{13,C}
```
^{13}C-NMR: δ

The second argument sets the frequency (in MHz):

```
\NMR(400)
```
^{1}H-NMR (400 MHz): δ

You can choose another unit:

```
\NMR(4e8,\hertz)
```
^{1}H-NMR (4×10^{8} Hz): δ

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

```
\sisetup{exponent-product=\cdot}
\NMR(4e8,\hertz)
```
^{1}H-NMR ($4 \cdot 10^{8}$ Hz): δ

The third argument specifies the solvent:

```
\NMR[CDCl3]
```
^{1}H-NMR ($CDCl_3$): δ

15.2. Short Cuts

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

`\DeclareChemNMR{`⟨*cs*⟩`}{`⟨*num*⟩`,`⟨*atom*⟩`}`
Define a new shortcut macro for typesetting a certain type of magnetic resonence data.

`\RenewChemNMR{`⟨*cs*⟩`}{`⟨*num*⟩`,`⟨*atom*⟩`}`
Redefine an existing shortcut macro for typesetting a certain type of magnetic resonence data.

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

```
\DeclareChemNMR\HNMR{1,H}%
\DeclareChemNMR\CNMR{13,C}%
\CNMR*(100) \par
\HNMR*(400)
```
^{13}C-NMR (100 MHz)
^{1}H-NMR (400 MHz)

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]}` $^{13}C\{^1H\}$-NMR: δ

`\J(`⟨*bonds*⟩`;`⟨*nuclei*⟩`)[`⟨*unit*⟩`]{`⟨*list of nums*⟩`}`
Coupling constant, values are input separated by `;` (NMR). The argument `(`⟨*bonds*⟩`;`⟨*nuclei*⟩`)` is optional and enables 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*⟩`}`.

```
\begin{experimental}
  \data{type1} Data.
  \data{type2}[specifications] More data.
  \data*{type3} Even more data.
\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 be a number of options. For historical reasons they all belong to the module `nmr`.

`nmr` » `unit` = {⟨*unit*⟩} Default: `\mega\hertz`
The used default unit.

`nmr` » `nucleus` = {⟨*num*⟩,⟨*atom*⟩} Default: `{1,H}`
The used default nucleus.

`nmr` » `format` = {⟨*commands*⟩} (initially empty)
For example `\bfseries`.

`nmr` » `pos-number` = `side|sub` Default: `side`
Position of the number next to the atom.

`nmr` » `coupling-unit` = {⟨*unit*⟩} Default: `\hertz`
A siunitx unit.

`nmr` » `parse` = `true|false` Default: `true`
Treat the solvent as CHEMFORMULA formula or not.

`nmr` » `delta` = {⟨*tokens*⟩} (initially empty)
The ⟨*tokens*⟩ are added after δ.

`nmr` » `list` = `true|false` Default: `false`
The environment `nmr` is formatted as a list

`nmr` » `list-setup` = {⟨*setup*⟩}
Setup of the list. See below for the default settings.

`nmr` » `use-equal` = `true|false` Default: `false`
Add egual sign after `\NMR` and `\data`.

The default setup of the list:

```
\topsep\z@skip \partopsep\z@skip
\itemsep\z@ \parsep\z@ \itemindent\z@
\leftmargin\z@
```

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

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

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

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

---

type1 = Data. **type2 (specifications)** = More data. **^{1}H-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.

```
\sisetup{separate-uncertainty,per-mode=symbol,detect-all,range-phrase=--}
\begin{experimental}[<optionen>]
  \data*{yield} \SI{17}{\milli\gram} yellow needles (\SI{0.04}{\milli\mole},
    \SI{13}{\percent}).
  %
  \data{mp.} \SI{277}{\celsius} (DSC).
  %
  \NMR(600)[CDCl3] \val{2.01} (s, \#{24}, \pos{5}), \val{2.31} (s, \#{12},
  \pos{1}), \val{6.72--6.74} (m, \#{2}, \pos{11}), \val{6.82} (s, \#{8},
  \pos{3}), \val{7.05--7.07} (m, \#{2}, \pos{12}), \val{7.39--7.41} (m, \#{4},
  \pos{9}), \val{7.48--7.49} (m, \#{4}, \pos{8}).
  %
  \NMR{13,C}(150)[CDCl3] \val{21.2} ($+$, \#{4}, \pos{1}), \val{23.4} ($+$,
  \#{8}, \pos{5}), \val{126.0} ($+$, \#{4}, \pos{9}), \val{128.2} ($+$, \#{8},
  \pos{3}), \val{130.8} ($+$, \#{2}, \pos{12}), \val{133.6} ($+$, \#{2},
  \pos{11}), \val{137.0} ($+$, \#{4}, \pos{8}), \val{138.6} (q, \#{4},
  \pos{2}), \val{140.6} (q, \#{2}, \pos{10}), \val{140.8} (q, \#{8}, \pos{4}),
  \val{141.8} (q, \#{4}, \pos{6}), \val{145.6} (q, \#{2}, \pos{7}).
  %
  \data{MS}[DCP, EI, \SI{60}{\electronvolt}] \val{703} (2, \ch{M+}), \val{582}
  (1), \val{462} (1), \val{249} (13), \val{120} (41), \val{105} (100).
  %
  \data{MS}[\ch{MeOH + H2O + KI}, ESI, \SI{10}{\electronvolt}] \val{720} (100,
  \ch{M+ + OH-}), \val{368} (\ch{M+ + 2 OH-}).
  %
  \data{IR}[KBr] \val{3443} (w), \val{3061} (w), \val{2957} (m), \val{2918}
  (m), \val{2856} (w), \val{2729} (w), \val{1725} (w), \val{1606} (s),
```

```
  \val{1592} (s), \val{1545} (w), \val{1446} (m), \val{1421} (m), \val{1402}
  (m), \val{1357} (w), \val{1278} (w), \val{1238} (s), \val{1214} (s),
  \val{1172} (s), \val{1154} (m), \val{1101} (w), \val{1030} (w), \val{979}
  (m), \val{874} (m), \val{846} (s), \val{818} (w), \val{798} (m), \val{744}
  (w), \val{724} (m), \val{663} (w), \val{586} (w), \val{562} (w), \val{515}
  (w).
  %
  \data*{UV-Vis} \SI{386}{\nano\metre} ($\varepsilon = \val{65984}$),
  \SI{406}{\nano\metre} ($\varepsilon = \val{65378}$).
  %
  \data*{quantum yield} $\Phi = \val{0.74+-0.1}$\,.
\end{experimental}
```

15.5.1. Nearly Standard

Output with these options:

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

yield: 17 mg yellow needles (0.04 mmol, 13 %). mp. = 277 °C (DSC). ^{1}H-NMR (600 MHz, $CDCl_3$): δ (ppm) = 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). ^{13}C-NMR (150 MHz, $CDCl_3$): δ (ppm) = 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 + H_2O + 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: $\Phi = 0.74 \pm 0.10$.

15.5.2. Formatted List

Output with these options:

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

yield: 17 mg yellow needles (0.04 mmol, 13 %).
mp. = 277 °C (DSC).
^{1}H-NMR (600 MHz, $CDCl_3$): δ (ppm) = 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).

^{13}C-NMR (150 MHz, $CDCl_3$): δ (ppm) = 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 + H_2O + 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: $\Phi = 0.74 \pm 0.10$.

15.5.3. Crazy

Output for these options:

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

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

mp. = 277 °C (DSC).

^{1}H-NMR (600 MHz, $CDCl_3$): 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).

^{13}C-NMR (150 MHz, $CDCl_3$): 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 + H_2O + 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: $\Phi = 0.74 \pm 0.10$.

16. Commands for mhchem

mhchem is not officially supported by CHEMMACROS as it used CHEMFORMULA instead. However, for historical reasons the following command is still provided.

`\mhName[`⟨*options*⟩`]{`⟨*formula*⟩`}{`⟨*text*⟩`}`
Writes ⟨*text*⟩ below ⟨*formula*⟩

For example:

```
\ce{
  4 C2H5Cl + Pb{/}Na
  ->
  \mhName{Pb(C2H5)4}{former antiknock additive} + NaCl
}
```

$$4C_2H_5Cl + Pb/Na \longrightarrow \underset{\substack{\text{former antiknock}\\ \text{additive}}}{Pb(C_2H_5)_4} + NaCl$$

There are several keys to customize `\mhName`.

`mhName » align = {`⟨*alignment command*⟩`}` Default: `\centering`
The alignment of the text in the box it is placed in.

`mhName » format = {`⟨*anything*⟩`}` (initially empty)
The format of the text.

`mhName » fontsize = {`⟨*font size command*⟩`}` Default: `\tiny`
The fontsize of the text.

`mhName » width =` ⟨*dim*⟩`|auto` Default: `auto`
The width of the box the text is placed in.

Since version 3.13 of mhchem you need to hide the command in braces if you want to use the optional argument,

```
\ce{
  4 C2H5Cl + Pb / Na
  ->
  {\mhName[fontsize=\footnotesize]{Pb(C2H5)4}{former antiknock additive}}
  + NaCl
}\par
```

```
\chemsetup[mhName]{
  align=\raggedright,
  fontsize=\small,
  format=\bfseries\color{red},
  width=3cm
}
\ce{
  4 C2H5Cl + Pb / Na
  ->
  \mhName{Pb(C2H5)4}{former antiknock additive} + NaCl
}
```

$$4C_2H_5Cl + Pb_7Na \longrightarrow \underset{\substack{\text{former}\\ \text{antiknock}\\ \text{additive}}}{Pb(C_2H_5)_4} + NaCl$$

$$4C_2H_5Cl + Pb_7Na \longrightarrow \underset{\textbf{former antiknock additive}}{Pb(C_2H_5)_4} + NaCl$$

17. Reaction Environments

17.1. 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.
Theses environments use the `equation`/`equation*` environments or the `align`/`align*` environments, respectively, to display the reactions.

```
Reaction with counter:
\begin{reaction}
```

```
  A -> B
\end{reaction}
```

Reaction with counter:

$$\mathrm{A} \longrightarrow \mathrm{B} \tag{1}$$

```
Reaction without counter:
\begin{reaction*}
  C -> D
\end{reaction*}
```

Reaction without counter:

$$\mathrm{C} \longrightarrow \mathrm{D}$$

```
Several aligned reactions with counter:
\begin{reactions}
  A     &-> B + C \\
  D + E &-> F
\end{reactions}
```

Several aligned reactions with counter:

$$\begin{aligned} \mathrm{A} &\longrightarrow \mathrm{B} + \mathrm{C} && (2)\\ \mathrm{D} + \mathrm{E} &\longrightarrow \mathrm{F} && (3) \end{aligned}$$

```
Several aligned reactions without counter:
\begin{reactions*}
  G     &-> H + I \\
  J + K &-> L
\end{reactions*}
```

Several aligned reactions without counter:

$$\begin{aligned} \mathrm{G} &\longrightarrow \mathrm{H} + \mathrm{I}\\ \mathrm{J} + \mathrm{K} &\longrightarrow \mathrm{L} \end{aligned}$$

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

`\renewtagform{`⟨*tagname*⟩`}[`⟨*format*⟩`]{`⟨*right delimiter*⟩`}{`⟨*left delimiter*⟩`}`.[13]

```
\renewtagform{reaction}[R \textbf]{[}{]}
\begin{reaction}
  H2O + CO2 <<=> H2CO3
\end{reaction}
```

$$H_2O + CO_2 \rightleftharpoons H_2CO_3 \qquad [\mathrm{R}\,\mathbf{4}]$$

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

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

$$\mathrm{A} + 2\,\mathrm{B} \longrightarrow 3\,\mathrm{C} + \mathrm{D} \qquad \{5\}$$

Some text in between aligned reactions

$$3\,\mathrm{E} + \mathrm{F} \rightleftharpoons \mathrm{G} + \frac{1}{2}\,\mathrm{H} \qquad \{6\}$$

See reaction 5.

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

17.2. Own Reactions

You can create new types of reactions with the command:

`\DeclareChemReaction[`⟨*options*⟩`]{`⟨*name*⟩`}{`⟨*math name*⟩`}`

⟨*name*⟩ will be the name of the new environment. ⟨*math name*⟩ is the underlying math environment.

The command has two options.

`star` = `true`|`false`

Also create a starred variant.

13. Provided by the mathtools package

`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

```
\DeclareChemReaction[star]{reaction}{equation}
\DeclareChemReaction[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:

```
\DeclareChemReaction[star,arg]{reactionsat}{alignat}
```

With this the `reactionsat` environment is defined.

```
\DeclareChemReaction[star,arg]{reactionsat}{alignat}
\begin{reactionsat}{3}
  A     &-> B     &&-> C     &&-> D \\
  aaaaa &-> bbbbb &&-> ccccc &&-> ddddd
\end{reactionsat}
\begin{reactionsat*}{2}
  A     &-> B     & C            &-> D  \\
  aaaaa &-> bbbbb &\quad{} ccccc &-> ddddd
\end{reactionsat*}
```

$$\begin{aligned} \mathrm{A} &\longrightarrow \mathrm{B} &&\longrightarrow \mathrm{C} &&\longrightarrow \mathrm{D} && \{7\} \\ \mathrm{aaaaa} &\longrightarrow \mathrm{bbbbb} &&\longrightarrow \mathrm{ccccc} &&\longrightarrow \mathrm{ddddd} && \{8\} \end{aligned}$$

$$\begin{aligned} \mathrm{A} &\longrightarrow \mathrm{B} & \qquad \mathrm{C} &\longrightarrow \mathrm{D} \\ \mathrm{aaaaa} &\longrightarrow \mathrm{bbbbb} & \mathrm{ccccc} &\longrightarrow \mathrm{ddddd} \end{aligned}$$

17.3. 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.

```
\listofreactions
```

List of reactions

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.

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

$$2\,H_2O \rightleftharpoons H_3O^+ + OH^- \qquad \{9\}$$

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.

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

$$Cl^\bullet + CH_4 \longrightarrow HCl + {}^\bullet CH_3 \qquad \{10\}$$
$$^\bullet CH_3 + Cl_2 \longrightarrow CH_3Cl + Cl^\bullet \qquad \{11\}$$

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).

18. Phases

18.1. Basics

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

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

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

$C\,(s) + 2\,H_2O\,(l) \longrightarrow CO_2\,(g) + 2\,H_2\,(g)$
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 TEX dimension.

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

$C_{(s)} + 2\,H_2O_{(l)} \longrightarrow CO_{2(g)} + 2\,H_{2(g)}$
To make it complete: $NaCl_{(aq)}$.

18.2. Define Own Phases

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

`\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 18.3 for a way to define language dependent settings.

`\RenewChemPhase`{⟨*cs*⟩}[⟨*german*⟩]{⟨*english*⟩}
Redefine an existing pahse command. Actually the optional argument is an artefact of an earlier implementation of the command. It has no effect at all. See section 18.3 for a way to define language dependent settings.

`\phase`{⟨*phase*⟩}
If you need a phase indicator just once or twice.

`\DeclareChemPhase` only defines a phase if ⟨*cs*⟩ is not already used by any other command. If it *is* already used CHEMMACROS will either give a warning or an error, depending on the option `strict`. `\RenewChemPhase` *only* defines a phase if ⟨*cs*⟩ *is* already used and issues a warning/error otherwise. Unlike the other declaration commands of CHEMMACROS `\DeclareChemPhase` and `\RenewChemPhase` *can only be used in the preamble.*

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

NaOH (aq,∞) H_2O (cd) U (cr) A (lc)
$NaOH_{(aq,\infty)}$ $H_2O_{(cd)}$ $U_{(cr)}$ $A_{(lc)}$

18.3. Language Dependencies

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

```
\DeclareChemPhase\liquid{l}
```

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

```
\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-`⟨*csname*⟩ where ⟨*csname*⟩ is the name of the phase command you defined without leading backslash.

See section 6 for predefined translations.

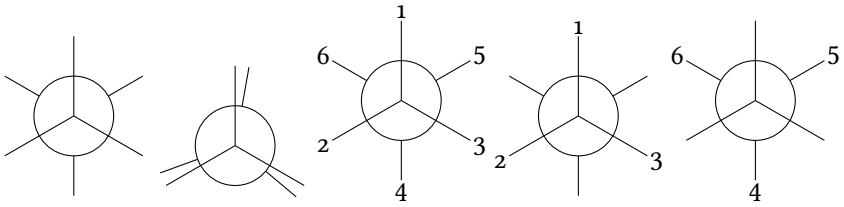
19. Newman Projections

CHEMMACROS provides a command to draw Newman projections.

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

Create Newman projections. This command uses Ti*k*Z 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.

```
\newman{} \newman(170){}
\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 Ti*k*Z keys.

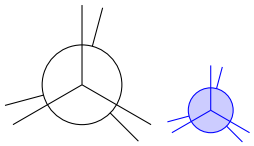
`newman` » `atoms` = {⟨*tikz*⟩} (initially empty)

Customize the nodes within which the atoms are set with Ti*k*Z keys.

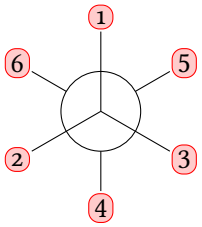
`newman` » `back-atoms` = {⟨*tikz*⟩} (initially empty)

Explicitly customize the nodes of the back atoms with Ti*k*Z keys.

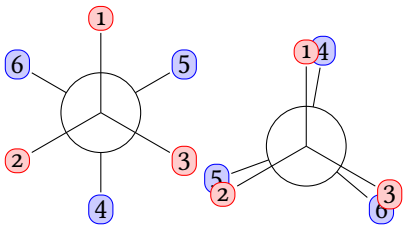
```
\chemsetup[newman]{angle=45} \newman{}
\newman[scale=.75,ring={draw=blue,fill=blue!20}]{}
```



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




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



20. s, p, and Hybrid Orbitals

CHEMMACROS provides the following command to create orbitals:

`\orbital`[⟨*options*⟩]{⟨*type*⟩}
Draw an orbital shape of type ⟨*type*⟩. This command uses Ti*k*Z internally.

There are the following types available for ⟨*type*⟩:

`s` `p` `sp` `sp2` `sp3`

```
\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`

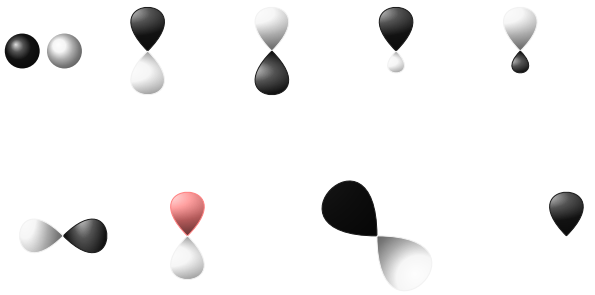
rotates the orbitals with a p contribution counter clockwise (all types except `s`)

`orbital` » `half` = `true`|`false` Default: `false`

displays only half an orbital (only `p`)

```
\orbital{s} \orbital[phase=-]{s}
\orbital{p} \orbital[phase=-]{p}
\orbital{sp3} \orbital[phase=-]{sp3}

\orbital[angle=0]{p} \orbital[color=red!50]{p}
\orbital[angle=135,scale=1.5]{p} \orbital[half]{p}
```



Additionally there are two options, with which the Ti*k*Z behaviour can be changed.

`orbital` » `overlay` = `true`|`false`

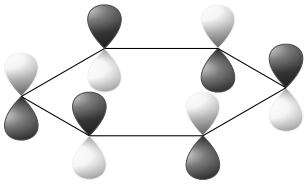
The orbital "doesn't need space"; it is displayed with the Ti*k*Z option `overlay`.

`orbital` » `opacity` = {⟨*num*⟩}

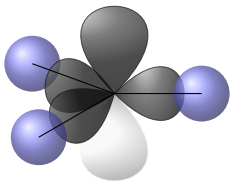
The orbital becomes transparent; ⟨*value*⟩ can have values between `1` (fully opaque) to `0` (invisible).

```
\vspace{7mm}
\chemsetup[orbital]{
  overlay,
  p/color = black!70
}
\setbondoffset{0pt}
```

```
\chemfig{
  ?\orbital{p}
  -[,1.3]{\orbital[phase=-]{p}}
  -[:30,1.1]\orbital{p}
  -[:150,.9]{\orbital[phase=-]{p}}
  -[4,1.3]\orbital{p}
  -[:-150,1.1]{\orbital[phase=-]{p}}?
}
\vspace{7mm}
```



```
\vspace{7mm}
\setbondoffset{0pt}
\chemsetup[orbital]{
  overlay ,
  opacity = .75 ,
  p/scale = 1.6 ,
  s/color = blue!50 ,
  s/scale = 1.6
}
\chemfig{
  \orbital{s}
  -[:-20]{\orbital[scale=2]{p}}
         {\orbital[half,angle=0]{p}}
         {\orbital[angle=170,half]{p}}
         {\orbital[angle=-150,half]{p}}
  (-[:-150]\orbital{s})-\orbital{s}
}
\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.[14]

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 Lüdigk
- Dr. Paul King

14. contact@mychemistry.eu

- Jonas Rivetti (Special thanks for his translation of the hazard and precautionary statements into Italian!)
- Christoph Schäfer
- Timo Stein

References

[CH09] David Carlisle and Morten Høgholm. xspace. version 1.13, Oct. 20, 2009. URL: http://mirror.ctan.org/macros/latex/required/tools/.

[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.

[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/.

[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/.

[Tan10] Till Tantau. TikZ/pgf. version 2.10, Oct. 25, 2010. URL: http://mirror.ctan.org/graphics/pgf/.

[The08] 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.

[The13a] The LaTeX3 Project Team. l3kernel. version SVN 4582, July 28, 2013. URL: http://mirror.ctan.org/macros/latex/contrib/l3kernel/.

[The13b] The LaTeX3 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