

CHEMFORMULA

v4.15j 2020/03/07

typeset chemical compounds and reactions

Clemens NIEDERBERGER

<http://www.mychemistry.eu/forums/forum/chemformula/>

contact@mychemistry.eu

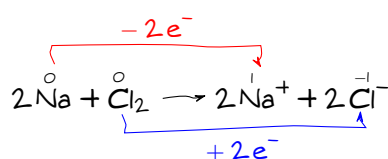


Table of Contents

1. Introduction	2	7. Special Input Types	17
2. Licence and Requirements	2	7.1. Single Token Groups	17
3. Setup	2	7.1.1. Addition Symbols	17
4. The Basic Principle	3	7.1.2. Symbols	18
5. Stoichiometric Factors	4	7.2. Option Input	19
6. Compounds	6	8. Escaped Input	19
6.1. Adducts	6	8.1. Text	19
6.2. Subscripts	7	8.2. Math	20
6.3. Commands	7	9. Arrows	21
6.4. Charges and Other Superscripts	8	9.1. Arrow types	21
6.5. Bonds	10	9.2. Labels	22
6.5.1. Native Bonds	10	9.3. Customization	23
6.5.2. Flexible Bonds	10	9.4. Modify Arrow Types	25
6.6. Customization	12	9.5. Standalone Arrows	27
6.7. Standalone Formulae	16	10. Names	27
6.8. Extend Compound Properties	16	10.1. Syntax	27
		10.2. Customization	28
		10.3. Standalone Names	28
		11. Format and Font	28

12. Usage In Math Equations	31	15. Kröger-Vink Notation	33
13. Usage with TikZ or pgfplots and externalization	31	A. History Since Version 4.0	34
14. Lewis Formulae	32	B. Index	37

1. Introduction

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 the CHEMMACROS package, 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 *many* possibilities to customize the output.

2. Licence and Requirements

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

The CHEMFORMULA package needs and thus loads the packages l3kernel [L3Pa], xparse, l3keys2e and xfrac (all three are part of the l3packages bundle [L3Pb]), tikz¹ [Tan19], ams-text [MSoo], nicefrac [Rei98] and scrfile (from the KOMA-Script² bundle [Koh19]).

3. Setup

If you're using CHEMFORMULA as a standalone package options are set up with the following command:

```
\setchemformula{<options>}
```

Set up CHEMFORMULA.

CHEMFORMULA is tightly intertwined with the CHEMMACROS package. If both packages are loaded together, CHEMFORMULA is integrated into the CHEMMACROS package. Then all of CHEMFORMULA's options belong to CHEMMACROS' module chemformula. This means if you load it via CHEMMACROS or in addition to CHEMMACROS they can be setup with

```
\chemsetup[chemformula]{<options>}
```

Set up options for CHEMFORMULA exclusively, or

1. on CTAN as pgf: <http://mirrors.ctan.org/graphics/pgf/>

2. on CTAN as koma-script: <http://mirrors.ctan.org/macros/latex/contrib/koma-script/>

4. The Basic Principle

`\chemsetup{chemformula/<option1>,chemformula/<option2>}`

Set up options for **CHEMFORMULA** together with others of **CHEMMACROS**' options.

4. The Basic Principle

CHEMFORMULA offers one main command.

`\ch[<options>]{<input>}`

CHEMFORMULA's main command.

The usage will seem very familiar to you if you're familiar with mhchem:

1 <code>\ch{H2O} \par</code>	H_2O
2 <code>\ch{Sb2O3} \par</code>	Sb_2O_3
3 <code>\ch{H+} \par</code>	H^+
4 <code>\ch{CrO4^2-} \par</code>	CrO_4^{2-}
5 <code>\ch{AgCl2-} \par</code>	AgCl_2^-
6 <code>\ch{[AgCl2]-} \par</code>	$[\text{AgCl}_2]^-$
7 <code>\ch{Y^{99+}} \par</code>	Y^{99+}
8 <code>\ch{Y^{99+}} \par</code>	Y^{99+}
9 <code>\ch{H2_{(aq)}} \par</code>	$\text{H}_{2(\text{aq})}$
10 <code>\ch{NO3-} \par</code>	NO_3^-
11 <code>\ch{(NH4)2S} \par</code>	$(\text{NH}_4)_2\text{S}$
12 <code>\ch{^{227}_{90}Th+} \par</code>	$^{227}_{90}\text{Th}^+$
13 <code>\$V_{\ch{H2O}}\$ \par</code>	$V_{\text{H}_2\text{O}}$
14 <code>\ch{Ce^{IV}} \par</code>	Ce^{IV}
15 <code>\ch{KCr(SO4)2 * 12 H2O}</code>	$\text{KCr}(\text{SO}_4)_2 \cdot 12 \text{H}_2\text{O}$

However, there are differences. The most notable one: **CHEMFORMULA** distinguishes between different types of input. These different parts *have* to be separated with blanks:

`\ch{part1 part2 part3 part4}`

A blank in the input *never* is a blank in the output. This role of the blank strictly holds and disregarding it can have unexpected results and even lead to errors.

The most visible differences regard spacing and the shapes of the default arrows:

1 <code>\ch{A + B ->[a] C} \par</code>	$\text{A} + \text{B} \xrightarrow{\text{a}} \text{C}$
2 <code>\ce{A + B ->[a] C}</code>	$\text{A} + \text{B} \longrightarrow \text{C}$

This means that `\ch{2H2O}` is recognized as a *single* part, which in this case is recognized as a compound.

5. Stoichiometric Factors

```
1 \ch{2H2O} \par          2H2O
2 \ch{2 H2O}             2 H2O
```

This also means, that a part cannot contain a blank since this will automatically divide it into two parts. If you need an extra blank in the output you need to use `~` or `\` . However, since commands in most cases gobble a space after them a input like `\ch{\command ABC}` will be treated as a single part. If you want or need to divide them you need to add an empty group: `\ch{\command{} ABC}`. The different input types are described in the following sections.

5. Stoichiometric Factors

A stoichiometric factor may only contain of numbers and the signs `.`, `-`/`()`

```
1 \ch{2} \par
2 \ch{12}
3
4 % decimals:
5 \ch{.5} \par
6 \ch{5,75}
7
8 % fractions:
9 \ch{3/2} \par
10 \ch{1_1/2}
11
12 % ``iupac``:
13 \ch{(1/2)}
```

As you can see if you input decimal numbers a missing leading zero is added. You have to be a little bit careful with the right syntax but I believe it is rather intuitive.

```
1 this won't work but will result in an error: \ch{1/1_1}
```

If stoichiometric factors are enclosed with parentheses the fractions are not recognized and missing leading zeros are not added. What's inside the parentheses is typeset as is.

```
1 \ch{(1/2) H2O} \ch{1/2 H2O} \ch{0.5 H2O}
```

(1/2)H₂O $\frac{1}{2}$ H₂O 0.5 H₂O

You can find many examples like the following for stoichiometric factors in parentheses in

5. Stoichiometric Factors

the IUPAC Green Book [Coh+08]:



There are a few possibilities to customize the output.

`decimal-marker = {<marker>}` Default: .

The symbol to indicate the decimal.

`frac-style = math|xfrac|nicefrac` Default: math

Determines how fractions are displayed.

`frac-math-cmd = {<command sequence>}` Default: `\frac`

Allows you to choose which command is used with `frac-style = {math}`. This needs to be a command sequence that takes two arguments that are set in math mode.

`stoich-space = {<skip>}` Default: .1667em plus .0333em minus .0117em

The space that is placed after the stoichiometric factor. A rubber length.

`stoich-paren-parse = true|false` Default: false

If set to true stoichiometric factors enclosed by parentheses also are parsed.

`stoich-print = {<cs>}` Default: `\chstoich`

This option allows to redefine the macro that prints the stoichiometric factors. `<cs>` should be a macro that takes one mandatory argument. *Please note that using this option will disable **CHEMFORMULA**'s stoichiometric parsing as that is done by the default command `\chstoich`.*

```
1 \ch[decimal-marker={,}]{3.5} \ch[decimal-marker={\cldot}]{3,5}
```

3,5 3·5

The option `frac-style = {xfrac}` uses the `\sfrac` command of the `xfrac` package. The output strongly depends on the font you use.

```
1 \ch[frac-style=xfrac]{3/2} \ch[frac-style=xfrac]{1_1/2}
```

¾ 1½

CHEMFORMULA defines the instance `chemformula-text-fraction` which you can redefine to your needs. See the `xfrac` documentation for further information. The default definition is this:

```
1 \DeclareInstance{xfrac}{chemformula-text-fraction}{text}
2 {
3   slash-left-kern = -.15em ,
```

Introduced
in version 4.1
(2013/08/24)

6. Compounds

```
4 slash-right-kern = -.15em
5 }
```

This document uses the font Linux Libertine O and the following definition:

```
1 \DeclareInstance{xfrac}{chemformula-text-frac}{text}
2 {
3   scale-factor      = 1 ,
4   denominator-bot-sep = -.2ex ,
5   denominator-format = \scriptsize #1 ,
6   numerator-top-sep  = -.2ex ,
7   numerator-format   = \scriptsize #1 ,
8   slash-right-kern   = .05em ,
9   slash-left-kern    = .05em
10 }
```

The option `frac-style = {nicefrac}` uses the `\nicefrac` command of the `nicefrac` package.

```
1 \ch[frac-style=nicefrac]{3/2} \ch[frac-style=nicefrac]{1_1/2}
```

 $\frac{3}{2}$ $1\frac{1}{2}$

The option `stoich-space` allows you to customize the space between stoichiometric factor and the group following after it.

```
1 \ch{2 H2O} \par                2 H2O
2 \ch[stoich-space=.3em]{2 H2O}  2 H2O
```

6. Compounds

CHEMFORMULA determines compounds as the type that “doesn’t fit in anywhere else.” This point will become more clear when you know what the other types are.

```
1 \ch{H2SO4} \par                H2SO4
2 \ch{[Cu(NH3)4]^2+}             [Cu(NH3)4]2+
```

6.1. Adducts

CHEMFORMULA has two identifiers which will create adducts.

6. Compounds

`\ch{A.B}`

A·B

`\ch{A*B}`

A·B

1 <code>\ch{CaSO4.H2O}</code> <code>\par</code>	CaSO ₄ ·H ₂ O
2 <code>\ch{CaSO4*H2O}</code>	CaSO ₄ ·H ₂ O

Since numbers in a compound always are treated as subscripts (see section 6.2) you sometimes need to introduce stoichiometric factors for the right output:

1 <code>\ch{Na3PO4*12H2O}</code> <code>\par</code>	Na ₃ PO ₄ · ₁₂ H ₂ O
2 <code>\ch{Na3PO4* 12 H2O}</code> <code>\par</code>	Na ₃ PO ₄ ·12 H ₂ O
3 <code>\ch{Na3PO4 * 12 H2O}</code>	Na ₃ PO ₄ ·12 H ₂ O

6.2. Subscripts

All numbers in a compound are treated as subscripts.

1 <code>\ch{H2SO4}</code>	H ₂ SO ₄
---------------------------	--------------------------------

If you want a letter to be a subscript you can use the math syntax:

1 <code>\ch{A_nB_m}</code>	A _n B _m
----------------------------	-------------------------------

The subscript recognizes groups. You can also use math inside it.

1 <code>\ch{A_{n\$}B_{m\$}}</code> <code>\par</code>	A _n B _m
2 <code>\ch{NaCl_{(aq)}}</code>	NaCl _(aq)

6.3. Commands

Commands are allowed in a compound:

1 <code>\ch{\textbf{A2}B3}</code> <code>\ch{A2\color{red}B3}</code>	A ₂ B ₃ A ₂ B ₃
---	---

6. Compounds

However, if the commands demand numbers as argument, *e. g.*, space commands or **CHEM-MACROS'** `\ox` command the direct use will fail. This is because the numbers are treated as subscripts *before* the command expands.

```
1 \ch{A\hspace{2mm}B} will raise an error because \hspace sees something like
2 this: \hspace{$_2$mm}. Actually not at all like this but equally bad\ldots
```

See section 8.1 for a way around this.

Please also note that formulas are placed inside a group!

```
1 \ch{A2\color{red}B3 C4}           A2B3C4
```

6.4. Charges and Other Superscripts

Basics If a compound *ends* with a plus or minus sign it will be treated as charge sign and typeset as superscript. In other places a plus is treated as a triple bond and a dash will be used as a single bond, see section 6.5.

```
1 \ch{A+B} \ch{AB+} \par           A≡B AB+
2 \ch{A-B} \ch{AB-} \par           A–B AB-
```

For longer charge groups or other superscripts you can use the math syntax. It recognizes groups and you can use math inside them. Inside these groups neither + nor - are treated as bonds. If a dot . is inside a superscript it is treated as indicator for a radical. A * gives the excited state.

```
1 \ch{A^{x-}} \par                 Ax-
2 \ch{A^x-} \par                   Ax-
3 \ch{A^{x}-} \par                 Ax-
4 \ch{A^{x-$}} \par                Ax-
5 \ch{RNO2^{-.}} \par              RNO2-•
6 \ch{^31H} \par                    31H
7 \ch{^14}6C \par                   146C
8 \ch{^58}_{26}Fe \par              5826Fe
9 \ch{NO^*} \par                    NO*
```

Changed in
version 4.5a ()

Actually a dot . is not always treated as indicator for a radical: if the dot in the superscript is followed by a number it is interpreted as a decimal sign. It is typeset according to the option `decimal-marker`. This may be a good place to mention that a comma , in a superscript is also typeset according to `decimal-marker`.

6. Compounds

```
1 \ch{^{22,98}_{11}Na}
2 \ch{^{22.98}_{11}Na}\par
3 \setchemformula{decimal-marker={,}}
4 \ch{^{22,98}_{11}Na}
5 \ch{^{22.98}_{11}Na}
```

$$\begin{array}{cc} {}^{22,98}_{11}\text{Na} & {}^{22.98}_{11}\text{Na} \\ {}^{22,98}_{11}\text{Na} & {}^{22,98}_{11}\text{Na} \end{array}$$

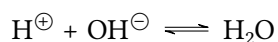
Ions and ion composites with more than one charge can be typeset quite as easy:

```
1 \ch{SO4^2-} \ch{Ca^2+ SO4^2-}
```

$$\text{SO}_4^{2-} \text{Ca}^{2+}\text{SO}_4^{2-}$$

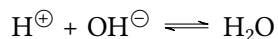
Charge Commands You don't need to use `\mch` and related commands inside `\ch`. Indeed, you *shouldn't* use them as they might mess with the subscript and superscript alignment. The `CHEMMACROS` option circled is obeyed by `\ch`.

```
1 \chemsetup[charges]{circled=all}
2 \ch{H+ + OH- <=> H2O}
```



`CHEMFORMULA` knows the options `circled` and `circletype` also on its own.

```
1 \setchemformula{circled=all}
2 \ch{H+ + OH- <=> H2O}
```



These options are coupled with `CHEMMACROS` options, *i. e.*, setting `CHEMMACROS`' options will also set `CHEMFORMULA`'s equivalents. The other way around the options act independently: setting `CHEMFORMULA`'s options will *not* set `CHEMMACROS`' options.

`circled` = `formal`|`all`|`none`

Default: `formal`

`CHEMFORMULA` uses two different kinds of charges which indicate the usage of real (+/−) and formal (⊕/⊖) charges. The choice `formal` distinguishes between them, choice `none` displays them all without circle, choice `all` circles all.

`circletype` = `chem`|`math`

Default: `chem`

This option switches between two kinds of circled charge symbols: `\fplus` ⊕ and `\oplus` ⊕.

Behaviour The superscripts behave differently depending on their position in a compound, if there are super- and subscripts following each other directly.

6. Compounds

```
1 \ch{^33B} \ch{{}^33B} \ch{3^3B} \ch{B^3} \ch{B3^3} \par
2 \ch{^{23}_{123}B} \ch{{}^{23}_{123}B} \ch{_{123}^{23}B}
3 \ch{B^{23}} \ch{B_{123}^{23}} \par
4 \ch{^{123}_{23}B} \ch{{}^{123}_{23}B} \ch{_{23}^{123}B}
5 \ch{B^{123}} \ch{B23^{123}}
```

```
3B 3B 3B B3 B3
23B 23B 23B B23 B23
123B 123B 123B B123 B123
```

- If a compound *starts* with a sub- or superscript both sub- and superscript are aligned to the *right* else to the *left*.
- If a compound *does not start* with a sub- or superscript and there is both a sub- and a superscript, the superscript is shifted additionally by a length determined from the option `charge-hshift = {<dim>}`, also see page 13f.

The second point follows IUPAC's recommendations:

In writing the formula for a complex ion, spacing for charge number can be added (staggered arrangement), as well as parentheses: SO_4^{2-} , $(\text{SO}_4)^{2-}$. The staggered arrangement is now recommended.

IUPAC Green Book [Coh+08, p. 51]

6.5. Bonds

6.5.1. Native Bonds

There are three kinds of what I will call "native bonds":

```
1 single: \ch{CH3-CH3} \par          single: CH3–CH3
2 double: \ch{CH2=CH2} \par        double: CH2=CH2
3 triple: \ch{CH+CH}               triple: CH≡CH
```

6.5.2. Flexible Bonds

Predefined Bonds In addition to the three native bonds there are a few more which can be called by

`\bond{<bond name>}`

Prints the bond type specified by `<bond name>`.

The predefined bond types are shown in table 1 on the following page.

6. Compounds

TABLE 1: Bonds available with `\bond`.

name	appearance	aliases
single	—	normal, sb
double	=	db
triple	≡	tp
dotted	⋯	semisingle
deloc	≡	semidouble
tdeloc	≡	semitriple
co>	→	coordright
<co	←	coordleft

```
\ch{\C\bond{sb}C\bond{db}C\bond{tp}C\bond{deloc}C\bond{tdeloc}C\bond{co>}C\bond{<co}C}
```

C—C=C≡C⋯C≡C→C←C

Own Bonds `CHEMFORMULA` offers commands to define own bond types:

`\NewChemBond{<name>}{<code>}`

Define the new bond type *<name>*. Issue an error if a bond *<name>* already exists.

`\DeclareChemBond{<name>}{<code>}`

Define the new bond type *<name>* or overwrite it if it already exists.

`\RenewChemBond{<name>}{<code>}`

Redefine the existing bond type *<name>*. Issue an error if a bond *<name>* doesn't exist.

`\ProvideChemBond{<name>}{<code>}`

Define the new bond type *<name>* only if it doesn't exist yet.

`\NewChemBondAlias{<new name>}{<old name>}`

Declare the bond type *<new name>* to be an alias of *<old name>*. Issue an error if a bond *<new name>* already exists.

`\DeclareChemBondAlias{<new name>}{<old name>}`

Declare the bond type *<new name>* to be an alias of *<old name>*.

`\ShowChemBond{<name>}`

Print the definition of bond type *<name>*.

The usage is best described with an example. So let's see how the single bond and the `co>` bond are defined:

Introduced in version 4.3 (2014/01/24)

Introduced in version 4.12a (2015/08/29)

Introduced in version 4.3

6. Compounds

```
1 \NewChemBond{single}
2 { \draw[chembond] (chemformula-bond-start) -- (chemformula-bond-end) ; }
3 \NewChemBond{coordright}
4 {
5   \draw[chembond,butt cap->]
6     (chemformula-bond-start) -- (chemformula-bond-end) ;
7 }
8 \NewChemBondAlias{co>}{coordright}
```

Two points are important: the names of the starting and the ending coordinates, `chemformula-bond-start` and `chemformula-bond-end`, and the TikZ style of the bonds `chembond`.

So, let's say you want to define a special kind of dashed bond. You could do this:

```
1 \usetikzlibrary{decorations.pathreplacing}
2 \makeatletter
3 \NewChemBond{dashed}
4 {
5   \draw[
6     chembond,
7     decorate,
8     decoration={
9       ticks,
10      segment length=\chemformula@bondlength/10,amplitude=1.5pt
11    }]
12   (chemformula-bond-start) -- (chemformula-bond-end) ;
13 }
14 \makeatother
15 \setchemformula{bond-length=2ex}
16 \ch{C\bond{dashed}C}
```



The last example showed you another macro: `\chemformula@bondlength`. It only exists so you can use it to access the bond length as set with `bond-length` directly.

6.6. Customization

These options allow you to customize the output of the compounds:

`subscript-vshift = {<dim>}` Default: 0pt

Extra vertical shift of the subscripts.

`subscript-style = text|math` Default: text

Style that is used to typeset the subscripts.

6. Compounds

	<code>charge-hshift = {⟨dim⟩}</code>	Default: .25em
	Shift of superscripts when following a subscript.	
	<code>charge-vshift = {⟨dim⟩}</code>	Default: 0pt
	Extra vertical shift of the superscripts.	
	<code>charge-style = text math</code>	Default: text
	Style that is used to typeset the superscripts.	
Introduced in version 4.6 (2014/06/30)	<code>circled = formal all none</code>	Default: formal
	Like <code>CHEMMACROS</code> ' package option but local to <code>CHEMFORMULA</code> 's <code>\ch</code> . That is: since <code>CHEMMACROS</code> ' macros use <code>CHEMFORMULA</code> 's mechanism this is effectively an alias.	
Introduced in version 4.6	<code>circletype = chem math</code>	Default: chem
	Like <code>CHEMMACROS</code> ' package option but local to <code>CHEMFORMULA</code> 's <code>\ch</code> . That is: since <code>CHEMMACROS</code> ' macros use <code>CHEMFORMULA</code> 's mechanism this is effectively an alias.	
	<code>adduct-space = {⟨dim⟩}</code>	Default: .1333em
	Space to the left and the right of the adduct point.	
Introduced in version 4.14 (2015/10/21)	<code>adduct-penalty = {⟨num⟩}</code>	Default: 300
	The penalty inserted after the adduct point for (dis-)allowing line breaks.	
	<code>bond-length = {⟨dim⟩}</code>	Default: .5833em
	The length of the bonds.	
	<code>bond-offset = {⟨dim⟩}</code>	Default: .07em
	Space between bond and atoms.	
	<code>bond-style = {⟨TikZ⟩}</code>	(initially empty)
	TikZ options for the bonds.	
Introduced in version 4.0a (2013/07/26)	<code>bond-penalty = {⟨num⟩}</code>	Default: 10000
	The penalty that is inserted after a bond for (dis-)allowing line breaks.	
	<code>radical-style = {⟨TikZ⟩}</code>	(initially empty)
	TikZ options for the radical point.	
	<code>radical-radius = {⟨dim⟩}</code>	Default: .2ex
	The radius of the radical point.	
	<code>radical-hshift = {⟨dim⟩}</code>	Default: .15em
	Horizontal shift before the radical point is drawn.	
	<code>radical-vshift = {⟨dim⟩}</code>	Default: .5ex
	Vertical shift relative to the current baseline.	
	<code>radical-space = {⟨dim⟩}</code>	Default: .15em
	Horizontal shift after the radical point is drawn.	

6. Compounds

Maybe you have noticed that charges of certain ions are shifted to the right.

```
1 \ch{S04^2-} \ch{NH4+} \ch{Na+}          SO42- NH4+ Na+
```

They are shifted if they *follow* a subscript which follows IUPAC recommendations [Coh+08, p. 51]. The amount of the shift can be set with the option `charge-hshift`.

```
1 \ch{S04^2-} \ch{NH4+} \ch{Na+} \par
2 \setchemformula{charge-hshift=.5ex}
3 \ch{S04^2-} \ch{NH4+} \ch{Na+} \par
4 \setchemformula{charge-hshift=.5pt}
5 \ch{S04^2-} \ch{NH4+} \ch{Na+}
```

SO₄²⁻ NH₄⁺ Na⁺
SO₄²⁻ NH₄⁺ Na⁺
SO₄²⁻ NH₄⁺ Na⁺

Despite IUPAC's recommendation `CHEMFORMULA` does not make fully staggered arrangements in the default setting as I find it hard to read in some cases and ugly in others. Since this is a subjective decision `CHEMFORMULA` not only let's you define the absolute amount of the shift but also provides a possibility for full staggered arrangements. For this you have to use `charge-hshift = {full}`.

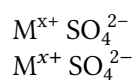
```
1 \ch[charge-hshift=0pt]{C5H11+} \ch[charge-hshift=0pt]{S04^2-} \par
2 \ch{C5H11+} \ch{S04^2-} \par
3 \ch[charge-hshift=1ex]{C5H11+} \ch[charge-hshift=1ex]{S04^2-} \par
4 \ch[charge-hshift=full]{C5H11+} \ch[charge-hshift=full]{S04^2-}
```

C₅H₁₁⁺ SO₄²⁻
C₅H₁₁⁺ SO₄²⁻
C₅H₁₁⁺ SO₄²⁻
C₅H₁₁⁺ SO₄²⁻

If you don't want the charges to be typeset in text mode you can switch to math mode:

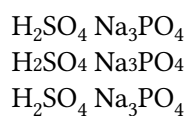
```
1 \ch{M^x+} \ch{S04^2-} \par
2 \setchemformula{charge-style = math}
3 \ch{M^x+} \ch{S04^2-}
```

6. Compounds



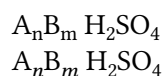
The option `subscript-vshift` can be used to adjust the vertical shift of the subscripts:

```
1 \ch{H2SO4} \ch{Na3PO4} \par
2 \setchemformula{subscript-vshift=.5ex}
3 \ch{H2SO4} \ch{Na3PO4} \par
4 \setchemformula{subscript-vshift=-.2ex}
5 \ch{H2SO4} \ch{Na3PO4}
```



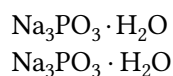
You can choose the mode subscripts are typeset in the same way as it is possible for the charges:

```
1 \ch{A_nB_m} \ch{H2SO4} \par
2 \setchemformula{subscript-style = math}
3 \ch{A_nB_m} \ch{H2SO4}
```



The option `adduct-space` sets the space left and right to the adduct symbol \cdot .

```
1 \ch{Na3PO3*H2O} \par
2 \setchemformula{adduct-space=.2em}
3 \ch{Na3PO3*H2O}
```



Changing the length of the bonds:

```
1 \setchemformula{bond-length=4mm}%
2 single: \ch{CH3-CH3} \par
3 double: \ch{CH2=CH2} \par
4 triple: \ch{CH+CH}
```

6. Compounds

single: CH₃—CH₃
double: CH₂=CH₂
triple: CH≡CH

You can change the distance between bond and atom, too:

```
1 \ch{H-H + N+N + O=O} \par
2 \ch[bond-offset=1pt]{H-H + N+N + O=O}
```

H—H + N≡N + O=O
H—H + N≡N + O=O

6.7. Standalone Formulae

Introduced in
version 4.0
(2013/07/06)

CHEMFORMULA offers a command that *only accepts* the “compound” input type:

`\chcpd[⟨options⟩]{⟨compound⟩}`
Typeset single compounds.

6.8. Extend Compound Properties

Introduced in
version 4.10
(2015/03/16)

It is possible to extend the range of special input symbols within compounds. In the default setting those are `*`, `-`, `=`, `+` and arabic numerals. Others can be added or the existing ones be changed with one of the following commands:

`\NewChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`
`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property is only added if `⟨token⟩` is not yet part the compounds’ property list. Otherwise an error is issued.

Introduced in
version 4.12a

`\ProvideChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`
`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property is only added if `⟨token⟩` is not yet part the compounds’ property list.

`\RenewChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`
`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property is only added if `⟨token⟩` is already part the compounds’ property list. Otherwise an error is issued.

`\DeclareChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`
`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property silently overwrites any previously set `⟨replacement⟩` for `⟨token⟩` if `⟨token⟩` is already part the compounds’ property list.

`\RemoveChemCompoundProperty{⟨token⟩}`
Removes `⟨token⟩` from the compounds’ property list.

For example you can use


```
\NewChemCompoundProperty{\}{\slash}
```

to allow line breaks after slashes in compounds.

7. Special Input Types

There are some “special type” input groups.

7.1. Single Token Groups

The first kind are groups which consist of only one token. They are again divided into two groups, “addition symbols” and “symbols”.

7.1.1. Addition Symbols

```
\ch{ + } +
```

Creates the plus sign between compounds with space around it:

```
\ch{2 Na + Cl2} 2 Na + Cl2
```

```
\ch{ - } -
```

Creates the minus sign between compounds with space around it:

```
\ch{M - H} M - H
```

Introduced in
version 4.3a ()

Addition symbols are surrounded with space which can be customized according to options explained in a bit. There is also some penalty prohibiting a line break after them which also can be customized with an option.

You can define/redefine your own addition symbols:

```
\NewChemAdditionSymbol{<name>}{<input>}{<output>}
```

Defines the addition symbol $\langle name \rangle$ with input symbol $\langle input \rangle$ and output $\langle output \rangle$.

Introduced in
version 4.11
(2015/06/30)

```
\ProvideChemAdditionSymbol{<name>}{<input>}{<output>}
```

Defines the addition symbol $\langle name \rangle$ with input symbol $\langle input \rangle$ and output $\langle output \rangle$ only no addition symbol with then name $\langle name \rangle$ doesn't exist.

Introduced in
version 4.12a

```
\RenewChemAdditionSymbol{<name>}{<input>}{<output>}
```

Redefines the addition symbol $\langle name \rangle$ with input symbol $\langle input \rangle$ and output $\langle output \rangle$.

Introduced in
version 4.11

```
\DeclareChemAdditionSymbol{<name>}{<input>}{<output>}
```

(Re-)Defines the addition symbol $\langle name \rangle$ with input symbol $\langle input \rangle$ and output $\langle output \rangle$ without checking if the symbol exists or not.

Introduced in
version 4.11

The space left and right of the plus and the minus sign and the signs themselves can be set with the following options:

7. Special Input Types

`plus-space = {<skip>}` Default: .3em plus .1em minus .1em
A rubber length.

Introduced in
version 4.0a

`plus-penalty = {<num>}` Default: 700
The penalty that is inserted after the plus sign for (dis-)allowing line breaks.

Introduced
in version 4.9
(2014/10/07)

`plus-output-symbol = {<code>}` Default: +
The `<code>` that is used for the plus sign.

Introduced in
version 4.9

`minus-space = {<skip>}` Default: .3em plus .1em minus .1em
A rubber length.

Introduced in
version 4.9

`minus-penalty = {<num>}` Default: 700
The penalty that is inserted after the minus sign for (dis-)allowing line breaks.

Introduced in
version 4.9

`minus-output-symbol = {<code>}` Default: \$-\$
The `<code>` that is used for the minus sign.

The corresponding three options are defined when `\NewChemAdditionSymbol` or one of the variants is used, `<name>-space` and `<name>-penalty` both with the same defaults as above, and `<name>-output-symbol`.

1	<code>\ch{A + B}\par</code>	A + B
2	<code>\ch[plus-space=4pt]{A + B}</code>	A + B

7.1.2. Symbols

`\ch{ v }↓`
Sign for precipitate: `\ch{BaSO4 v} BaSO4↓`

`\ch{ ^ }↑`
Sign for escaping gas³: `\ch{H2 ^} H2↑`

You can define/redefine your own symbols:

Introduced in
version 4.11

`\NewChemSymbol{<input>}{<output>}`
Defines the addition symbol with input `<input>` and output `<output>`.

Introduced in
version 4.12a

`\ProvideChemSymbol{<input>}{<output>}`
Defines the addition symbol with input `<input>` and output `<output>` only if no symbol with input `<input>` exists.

Introduced in
version 4.11

`\RenewChemSymbol{<input>}{<output>}`
Redefines the addition symbol with input `<input>` and output `<output>`.

Introduced in
version 4.11

`\DeclareChemSymbol{<input>}{<output>}`
(Re-)Defines the addition symbol with input `<input>` and output `<output>` without checking if the symbol exists or not.

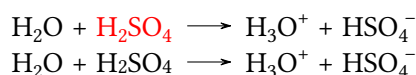
³. Is this the correct English term? Please correct me if it isn't.

8. Escaped Input

7.2. Option Input

Sometimes you might want to apply an option only to a part of a, say, reaction. Of course you have the possibility to use `\ch` several times.

```
1 \ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HS04-} \par
2 \ch{H2O +}\ch[subscript-vshift=2pt]{H2SO4}\ch{-> H3O+ + HS04-}
```

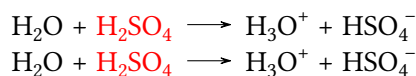


This, however, interrupts the input in your source and *may* mess with the spacing. That's why there is an alternative:

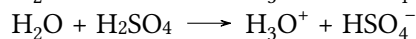
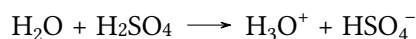
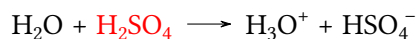
`\ch{ @{\options} }`

The options specified this way will be valid *only* until the next compound is set.

```
1 \ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HS04-} \par
2 \ch{H2O + @{\format=\color{red}} H2SO4 -> H3O+ + HS04-} \par
3 or of course:\par
4 \ch{H2O + \textcolor{red}{H2SO4} -> H3O+ + HS04-}\par\bigskip
5 \ch{H2O +}\ch[subscript-vshift=2pt]{H2SO4}\ch{-> H3O+ + HS04-} \par
6 \ch{H2O + @{\subscript-vshift=2pt} H2SO4 -> H3O+ + HS04-}
```



or of course:



8. Escaped Input

In some cases it may be desirable to prevent `CHEMFORMULA` from parsing the input. This can be done in two ways.

8.1. Text

If you put something between " " or ' ' then the input will be treated as normal text, except that spaces are not allowed and have to be input with `~`.

`\ch{ "escaped text" }`

One of two possibilities to *escape* `CHEMFORMULA`'s parsing.

8. Escaped Input

`\ch{ 'escaped text' }`

The second of two possibilities to *escape* CHEMFORMULA's parsing.

```
1 \ch{"\ox{2,Ca}" 0} \par
2 \ch{"\ldots\," Na + "\ldots\," Cl2 -> "\ldots\," NaCl} \par
3 \ch{'A~->~B'}
```

Ca⁰O
... Na + ... Cl₂ → ... NaCl
A -> B

In many cases you won't need to escape the input. But when you get into trouble when using a command inside `\ch` try hiding it.

8.2. Math

If you especially want to input math you just enclose it with `$ $`. This output is different from the escaped text as it is followed by a space. The reasoning behind this is that I assume math will mostly be used to replace stoichiometric factors.

`\ch{ escaped math $ }`

One of two possibilities to *escape* CHEMFORMULA's parsing into math mode.

`\ch{ \(\(escaped math)\)`

The second of two possibilities to *escape* CHEMFORMULA's parsing into math mode.

1 escaped text: <code>\ch{"\$x\$" H2O} \par</code>	escaped text: $x\text{H}_2\text{O}$
2 escaped math: <code>\ch{\$x\$ H2O} \par</code>	escaped math: $x\text{H}_2\text{O}$
3 also escaped math: <code>\ch{\(x\) H2O} \par</code>	also escaped math: $x\text{H}_2\text{O}$
4 <code>\ch{\$2n\$ Na + \$n\$ Cl2 -> \$2n\$ NaCl}</code>	$2n\text{Na} + n\text{Cl}_2 \longrightarrow 2n\text{NaCl}$

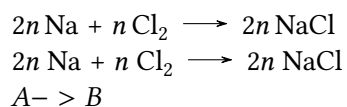
The space that is inserted after a math group can be edited:

`math-space = {skip}`

Default: .1667em plus .0333em minus .0117em

A rubber length.

```
1 \ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl} \par
2 \setchemformula{math-space=.25em}
3 \ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl} \par
4 \ch{$A->B$}
```



9. Arrows

9.1. Arrow types

Arrows are input in the same intuitive way they are with mhchem. There are various different types:

-> \longrightarrow
standard right arrow

<- \longleftarrow
standard left arrow

-/> \nrightarrow
does not react (right)

</- \nleftarrow
does not react (left)

<-> \longleftrightarrow
resonance arrow

<> \rightleftharpoons
reaction in both directions

== =
stoichiometric equation

<=> \rightleftharpoons
equilibrium arrow

>=< \rightleftharpoons
reversed equilibrium arrow

Introduced in version 4.5
(2014/04/07)







<=>> \rightleftharpoons
unbalanced equilibrium arrow to the right

Introduced in version 4.5

>=<< \rightleftharpoons
reversed unbalanced equilibrium arrow to the right

<<=> \rightleftharpoons
unbalanced equilibrium arrow to the left

9. Arrows

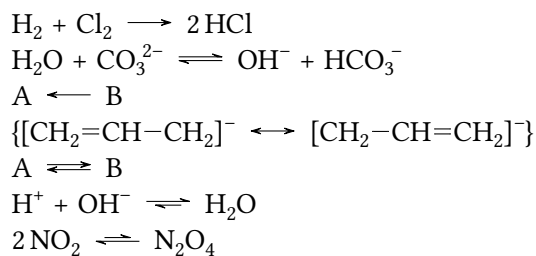
Introduced in version 4.5	$\gg<$ 	reversed unbalanced equilibrium arrow to the left
Introduced in version 4.15 (2016/05/11)	$<=0>$ 	quasi equilibrium arrow
Introduced in version 4.15	$<=0>>$ 	unbalanced quasi equilibrium arrow to the right
Introduced in version 4.15	$<<=0>$ 	unbalanced quasi equilibrium arrow to the left
	$<0>$ 	isolobal arrow
Introduced in version 4.5	$<==>$ 	I've seen this one used. I'm not sure it actually has a meaning in chemical equations. If you have some official reference for this arrow type please feel free to contact me.

All these arrows are drawn with TikZ.

```

1 \ch{H2 + Cl2 -> 2 HCl} \par
2 \ch{H2O + CO3^2- <=> OH- + HCO3-} \par
3 \ch{A <- B} \par
4 \ch{\{[CH2=CH-CH2]- <-> \}[CH2-CH=CH2]- \} } \par
5 \ch{A <> B} \par
6 \ch{H+ + OH- <==> H2O} \par
7 \ch{2 NO2 <<=> N2O4}

```



9.2. Labels

The arrows take two optional arguments to label them.

$\rightarrow[\langle above \rangle][\langle below \rangle]$

Add text above or under an arrow.

9. Arrows

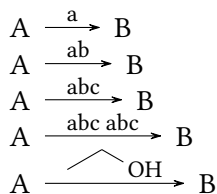
<pre>1 \ch{A ->[a] B} \par 2 \ch{A ->[a][b] B} \par 3 \ch{A ->[\SI{100}{\celsius}] B}</pre>	$A \xrightarrow{a} B$ $A \xrightarrow[a]{b} B$ $A \xrightarrow{100^\circ\text{C}} B$
--	--

The label text can be parsed separately from the arrow. The recipe is easy: leave blanks.

<pre>1 \ch{A ->[H2O] B} \par 2 \ch{A ->[H2O] B} \par 3 \ch{A ->["\ox{2,Ca}" F2] B} \par 4 \ch{A ->[\$\Delta\$,~ [H+]] B}</pre>	$A \xrightarrow{\text{H}_2\text{O}} B$ $A \xrightarrow{\text{H}_2\text{O}} B$ $A \xrightarrow{\text{Ca}^{\text{II}}\text{F}_2} B$ $A \xrightarrow{\Delta, [\text{H}^+]} B$
--	--

If you leave the blanks `CHEMFORMULA` treats the groups inside the square brackets as separated input types. The arrow reads its arguments *afterwards*. As you can see the arrows “grow” with the length of the labels. What stays constant is the part that protrudes the labels.

```
1 \ch{A ->[a] B} \par
2 \ch{A ->[ab] B} \par
3 \ch{A ->[abc] B} \par
4 \ch{A ->[abc~abc] B} \par
5 % needs the `chemfig' package:
6 \setchemfig{atom sep =15pt}
7 \ch{A ->[ "\chemfig{-[:30]-[: -30]OH}" ] B} \par
```



9.3. Customization

These are the options which enable you to customize the arrows:

`arrow-offset` = $\langle dim \rangle$ Default: .75em

This is the length that an arrow protrudes a label on both sides. This means an empty arrow's length is two times `arrow-offset`.

`arrow-min-length` = $\langle dim \rangle$ Default: 0pt

The minimal length an error must have unless two times `arrow-offset` plus the width of the label is larger.

`arrow-yshift` = $\langle dim \rangle$ Default: 0pt

Shifts an arrow up (positive value) or down (negative value).

Introduced in
version 3.6b
(2013/04/19)

9. Arrows

`arrow-ratio` = $\{\langle factor \rangle\}$ Default: .6

The ratio of the arrow lengths of the unbalanced equilibrium. .4 would mean that the length of the shorter arrow is 0.4× the length of the longer arrow.

`compound-sep` = $\{\langle dim \rangle\}$ Default: .5em

The space between compounds and the arrows.

`label-offset` = $\{\langle dim \rangle\}$ Default: 2pt

The space between the labels and the arrows.

`label-style` = $\{\langle font command \rangle\}$ Default: `\footnotesize`

The relative font size of the labels.

`arrow-penalty` = $\{\langle num \rangle\}$ Default: 0

The penalty that is inserted after an arrow for (dis-)allowing line breaks.

Introduced in
version 4.0a

`arrow-style` = $\{\langle TikZ \rangle\}$ (initially empty)

Additional TikZ keys for formatting the arrows.

Introduced in
version 4.1a ()

The following code shows the effect of the different options on the `<=>` arrow:

```

1 standard: \ch{A <=>[x][y] B} \par
2 longer: \ch[arrow-offset=12pt]{A <=>[x][y] B} \par
3 higher: \ch[arrow-yshift=2pt]{A <=>[x][y] B} \par
4 more balanced: \ch[arrow-ratio=.8]{A <=>[x][y] B} \par
5 labels further away: \ch[label-offset=4pt]{A <=>[x][y] B} \par
6 larger distance to compounds: \ch[compound-sep=2ex]{A <=>[x][y] B} \par
7 smaller labels: \ch[label-style=\tiny]{A <=>[x][y] B}

```

standard: A $\xrightarrow{\frac{x}{y}}$ B
longer: A $\xrightarrow{\frac{x}{y}}$ B
higher: A $\xrightarrow{\frac{x}{y}}$ B
more balanced: A $\xrightarrow{\frac{x}{y}}$ B
labels further away: A $\xrightarrow{\frac{x}{y}}$ B
larger distance to compounds: A $\xrightarrow{\frac{x}{y}}$ B
smaller labels: A $\xrightarrow{\frac{x}{y}}$ B

Introduced
in version 4.7
(2014/08/04)

If you want to have different arrow tips there is an easy way to use existing arrow tips (as defined by TikZ). `CHEMFORMULA` uses three different arrow tips: `cf`, `left cf` and `right cf`. If you want them to match those of chemfig [Tel19] for example you could do:

```

1 \pgfkeys{
2   cf /.tip = {CF@full} ,
3   left cf /.tip = {CF@half}

```



```
4 }
```

chemfig has no equivalent of `right cf`. This mechanism relies on TikZ version 3.0.0 and the new `arrows.meta` library.

9.4. Modify Arrow Types

The arrows are defined with the commands

```
\NewChemArrow{<type>}{<TikZ>}
```

Define the new arrow type `<type>`. Issue an error if an arrow type `<type>` already exists.

```
\ProvideChemArrow{<type>}{<TikZ>}
```

Define the new arrow type `<type>` only if it doesn't exist, yet.

```
\DeclareChemArrow{<type>}{<TikZ>}
```

Define the new arrow type `<type>` or overwrite it if it already exists.

```
\RenewChemArrow{<type>}{<TikZ>}
```

Redefine the arrow type `<type>`. Issue an error if an arrow type `<type>` doesn't exist.

```
\ShowChemArrow{<type>}
```

Print out the current definition of the arrow type `<type>`.

`<type>` is the sequence of tokens that is replaced with the actual arrow code. For example the basic arrow is defined via

```
1 \NewChemArrow{->}{
2   \draw[chemarrow,-cf] (cf_arrow_start) -- (cf_arrow_end) ;
3 }
```

In order to define arrows yourself you need to know the basics of TikZ.⁴ The predefined arrows use the arrow tips `cf`, `left cf` and `right cf`. They also all except the net reaction arrow `==` use the TikZ-style `chemarrow` that you should use, too, if you want the option `arrow-style` to have an effect.

There are some predefined coordinates you can and should use. For completeness' sake the arrow tips and the TikZ-style are also listed:

`(cf_arrow_start)`

The beginning of the arrow.

`(cf_arrow_end)`

The end of the arrow.

⁴. Please see the `pgfmanual` for details.

9. Arrows

(cf_arrow_mid)

The mid of the arrow.

(cf_arrow_mid_start)

The beginning of the shorter arrow in types like \Leftrightarrow .

(cf_arrow_mid_end)

The end of the shorter arrow in types like \Leftrightarrow .

cf

A double-sided arrow tip.

left cf

A left-sided arrow tip.

right cf

A right-sided arrow tip.

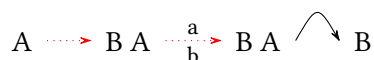
chemarrow

CHEMFORMULA's TikZ-style that is applied to the arrows and set with `arrow-style`

```

1 \NewChemArrow{.>}{
2   \draw[chemarrow,-cf,dotted,red] (cf_arrow_start) -- (cf_arrow_end);
3 }
4 \NewChemArrow{n>}{
5   \draw[chemarrow,-cf]
6     (cf_arrow_start)
7     .. controls ([lyshift=3ex]cf_arrow_mid) ..
8     (cf_arrow_end);
9 }
10 \ch{A .> B} \ch{A .>[a][b] B} \ch{A n> B}

```



```

1 \texttt{\ShowChemArrow{->}} \par
2 \RenewChemArrow{->}{\draw[chemarrow,->,red] (cf_arrow_start) -- (cf_arrow_end)
3 } ;}
4 \texttt{\ShowChemArrow{->}} \par
5 \ch{A -> B}

```

```

\draw [chemarrow,-cf](cf_arrow_start)--(cf_arrow_end);
\draw [chemarrow,->,red] (cf_arrow_start) -- (cf_arrow_end) ;
A \longrightarrow B

```

9.5. Standalone Arrows

Introduced in
version 4.0

CHEMFORMULA offers a command that *only accepts* the “arrow” input type:

`\charrow{<type>}[<above>][<below>]`
Print the arrow type *<type>*.

This command is internally used for the arrows, too, when `\ch` is parsed.

10. Names

10.1. Syntax

CHEMFORMULA has a built-in syntax to write text under a compound. In a way it works very similar to the arrows.

`\ch{ !(<text>)(<formula>) }`
Writes *<text>* below *<formula>*.

If an exclamation mark is followed by a pair of parentheses **CHEMFORMULA** will parse it this way:

```
1 \ch{!(ethanol)( CH3CH2OH )}
```

CH₃CH₂OH
ethanol

The same what’s true for the arrows arguments holds for these arguments: if you leave blanks the different parts will be treated according to their input type before the text is set below the formula.

```
1 \ch{!(water)(H2O)} \quad
2 \ch{!( "\textcolor{blue}{water}" )( H2O )} \quad
3 \ch{!( $2n-1$ )( H2O )} \quad
4 \ch{!( H2O )( H2O )} \quad
5 \ch{!(oxonium)( H3O+ )}
```

H₂O H₂O H₂O H₂O H₃O⁺
water water 2n – 1 H₂O oxonium

If for some reason you want to insert an exclamation mark *without* it creating a name you only have to make sure it isn’t followed by parentheses.

1 <code>\ch{H2O~(!)} \par</code>	H2O (!)
2 <code>\ch{A!{}}()</code>	A!()

10.2. Customization

CHEMFORMULA provides two options to customize the output of the names:

`name-format = {<commands>}` Default: `\scriptsize\centering`

The format of the name. This can be arbitrary input.

`name-width = <dim>|auto` Default: auto

The width of the box where the label is put into. auto will detect the width of the name and set the box to this width.

```

1 \ch{!(acid)( H2SO4 ) -> B} \par
2 \ch[name-format=\sffamily\small]{!(acid)( H2SO4 ) -> B} \par
3 \ch[name-format=\scriptsize N:~]{!(acid)( H2SO4 ) -> B} \par
4 \ch[name-width=3em,name-format=\scriptsize\raggedright]{!(acid)( H2SO4 ) -> B
  }

```

H₂SO₄ → B
acid

H₂SO₄ → B
acid

H₂SO₄ → B
N: acid

H₂SO₄ → B
acid

10.3. Standalone Names

Introduced in
version 4.0

CHEMFORMULA offers a command that allows the usage of the “name” syntax in normal text. This is the command that a bang is replaced with in **CHEMFORMULA**’s formulas, actually. Both arguments are mandatory.

`\chname(<text 1>)(<text 2>)`

The command that is used internally for placing *<text 1>* below of *<text 2>*.

11. Format and Font

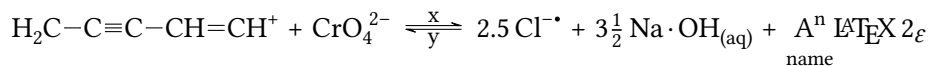
In the standard setting **CHEMFORMULA** doesn’t make any default changes to the font of the formula output. Let’s take a look at a nonsense input which shows all features:

11. Format and Font

```

1 \newcommand*\sample{%
2   \ch{H2C-C≡C-CH=CH+ + CrO4^2-
3     <=>[x][y]
4     2.5 Cl^{-.} + 3_1/2 Na*OH_{(aq)} + !(name)( A^n ) "\LaTeXe"}
5 }
6 \sample

```

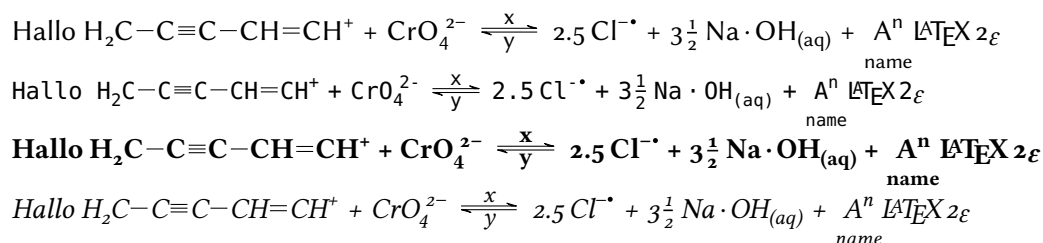


Now we're going to change different aspects of the font a look what happens:

```

1 \sffamily Hallo \sample \
2 \ttfamily Hallo \sample \normalfont \
3 \bfseries Hallo \sample \normalfont \
4 \itshape Hallo \sample

```



As you can see most features adapt to the surrounding font.

If you want to change the default format you need to use this option:

`format = {<code>}` (initially empty)

Adds <code> before the output of `\ch`.

`atom-format = {<code>}` (initially empty)

This adds <code> before each formula. This allows to specify a format for the chemical formulas only and have a different format for the rest of the chemical equation.

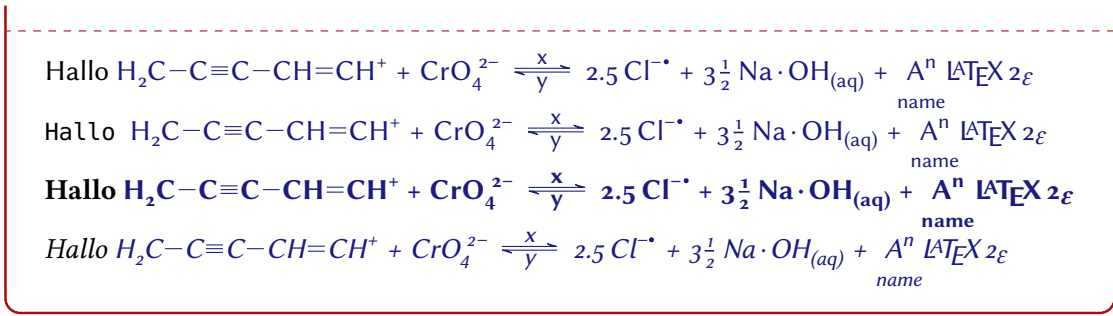
Introduced in
version 4.13
(2015/01/01)

```

1 \definecolor{newblue}{rgb}{.1,.1,.5}
2 \setchemformula{format=\color{newblue}\sffamily}
3 \sffamily Hallo \sample \
4 \ttfamily Hallo \sample \normalfont \
5 \bfseries Hallo \sample \normalfont \
6 \itshape Hallo \sample

```

11. Format and Font



You can also specifically change the fontfamily, fontseries and fontshape of the output.

font-family = {<family>} (initially empty)

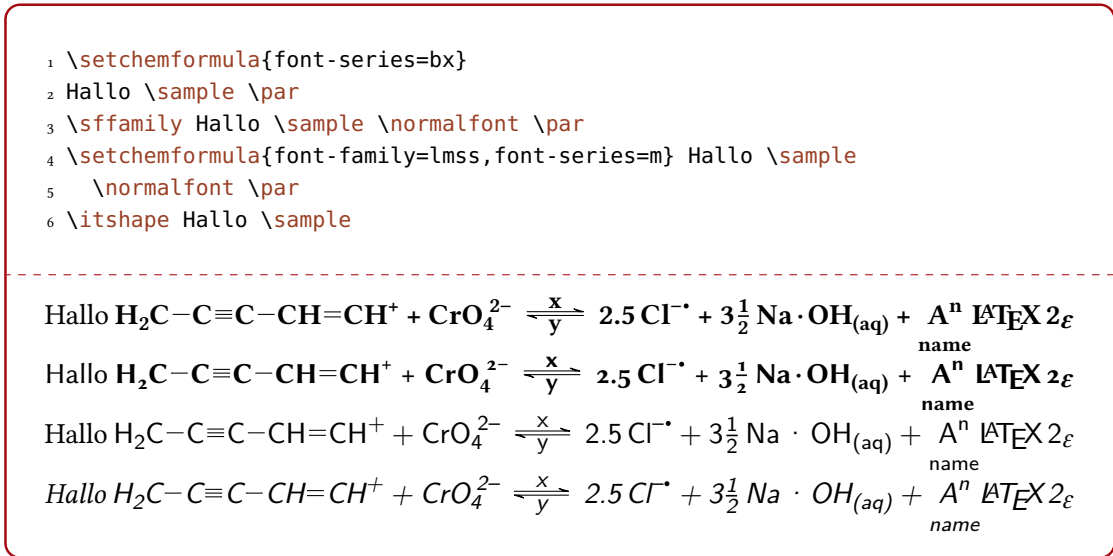
Changes the fontfamily of the output with `\fontfamily{<family>}`.

font-series = {<series>} (initially empty)

Changes the fontseries of the output with `\fontseries{<series>}`.

font-shape = {<shape>} (initially empty)

Changes the fontshape of the output with `\fontshape{<shape>}`.



If you're using $\text{X}_{\text{T}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ or $\text{L}^{\text{u}}\text{a}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and have loaded fontspec you have the possibility to set the font with it:

font-spec = {} (initially empty)

Use font for **CHEMFORMULA**'s formulas.

or with options

font-spec = {[<options>]}

Use font with options <options> for **CHEMFORMULA**'s formulas.

Since this document is typeset with $\text{p}^{\text{d}}\text{fL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ the option cannot be demonstrated here.

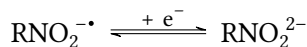
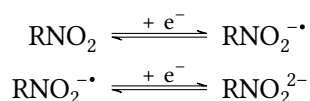
12. Usage In Math Equations

The `\ch` command can be used inside math equations. It recognizes `\` and `&` and passes them on. However, you can't use the optional arguments of `\` inside `\ch`.

```

1 \begin{align}
2 \ch{
3   H2O & \rightarrow[a] H2SO4 \\
4   Cl2 & \rightarrow[x][y] CH4
5 }
6 \end{align}
7 \begin{align*}
8 \ch{
9   RNO2 & \rightleftharpoons[ + e^- ] RNO2^{\cdot} \\
10  RNO2^{\cdot} & \rightleftharpoons[ + e^- ] RNO2^{2-}
11 }
12 \end{align*}

```



13. Usage with TikZ or pgfplots and externalization

Introduced in
version 4.1

Since `CHEMFORMULA` uses `TikZ` to draw reaction arrows and bonds they would be externalized, too, if you use that facility with `TikZ` or `pgfplots`⁵ [Feu18]. This may not be desirable since they are very small pictures maybe containing of a single line. This is why `CHEMFORMULA`'s default behaviour is to disable externalization for its bonds and arrows. This can be turned on and off through the following option:

`tikz-external-disable = true|false` Default: true
dis- or enable `TikZ`' externalization mechanism for `CHEMFORMULA`'s arrows and bonds.

If you should be using a formula that contains bonds or arrows inside of a `tikzpicture` that is externalized you should locally enable it for `CHEMFORMULA`, too:

5. on CTAN as `pgfplots`: <http://mirrors.ctan.org/macros/latex/contrib/pgfplots/>

```

1 \begin{tikzpicture}
2   \setchemformula{tikz-external-disable=false}
3   \begin{axis}[xlabel={\ch{2 H+ + 2 e- -> H2}}]
4     \addplot ... ;
5   \end{axis}
6 \end{tikzpicture}

```

14. Lewis Formulae

Introduced
in version 4.2
(2013/10/24)

CHEMFORMULA offers a command to typeset Lewis formulae. This does not mean Lewis structures! Those can be achieved using the chemfig package [Tel19]. **CHEMFORMULA** provides the possibility to draw electrons as dots and pairs of dots or a line around an atom.

`\chlewis`[*<options>*]{*<electron spec>*}{*<atom>*}

Draws electrons around the *<atom>* according to *<electron spec>*.

Electrons are specified by the angle to the horizontal in the counter-clockwise direction. The default appearance is a pair of electrons drawn as a pair of dots. Other specifications can be chosen. The specification follows the pattern *<angle>**<separator>*. *<angle>* is a positive or negative integer denoting the angle counter clockwise to the horizontal where the electrons should be drawn. *<separator>* is either a dot (., single electron), a colon (:, electron pair), a vertical line (|, electron pair), an o (o, empty pair), or a comma (, default spec).

`\chlewis`{*<angle1>**<type1>**<angle2>**<type2>*}{*<atom>*}

For example: `\chlewis{0,180}{0}` gives :O: and `\chlewis{0.90.180.270.}{C}` gives $\text{•}\overset{\cdot}{\underset{\cdot}{\text{C}}}\text{•}$.

The appearance can be influenced by a number of options:

`lewis-default` = .|:|o|single|pair|pair (dotted)|pair (line)|empty Default: pair
Sets the default type that is used when no type is given in *<electron spec>*.

`lewis-distance` = {*<dim>*} Default: 1ex
The distance of two electrons in a pair.

`lewis-line-length` = {*<dim>*} Default: 1.5ex
The length of the line representing an electron pair.

`lewis-line-width` = {*<dim>*} Default: 1pt
The thickness of a line representing an electron pair.


`lewis-offset` = {*<dim>*} Default: .5ex
The distance of the symbols from the atom.

15. Kröger-Vink Notation

The dots are drawn according to the `radical-radius` option mentioned in section 6.6.

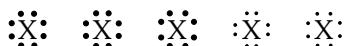
The basic usage should be more or less self-explaining:

```
1 \chlewis{0:90|180.270}{0}
2 \quad
3 \chlewis{45,135}{0}
4 \quad
5 \chlewis{0o}{Na}
```

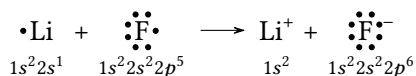


The next example shows the effect of some of the options:

```
1 \chlewis[lewis-default=.]{23,68,113,158,203,248,293,338}{X}
2 \quad
3 \chlewis{0,90,180,270}{X}
4 \quad
5 \chlewis[lewis-distance=1.25ex]{0,90,180,270}{X}
6 \quad
7 \chlewis[lewis-distance=.75ex,radical-radius=.5pt]{0,90,180,270}{X}
8 \quad
9 \chlewis[
10   radical-radius=.5pt,
11   lewis-default=.
12 ]{23,68,113,158,203,248,293,338}{X}
```



```
1 \ch{
2   !($1s^22s^1$)( "\chlewis{180.}{Li}" ) +
3   !($1s^22s^22p^5$)( "\chlewis{0.90,180,270}{F}" )
4   ->
5   !($1s^2$)( Li+ ) + !($1s^22s^22p^6$)( "\chlewis{0,90,180,270}{F}" {}- )
6 }
```



15. Kröger-Vink Notation

Introduced in
version 4.5

`CHEMFORMULA` also supports the Kröger-Vink notation.

`kroeger-vink = true|false`

Default: `false`

Enable the Kröger-Vink notation. As most options this can be enabled globally via the setup

A. History Since Version 4.0

command or locally as option to `\ch`.

With this option enabled several changes come into effect: ' produces a prime, a x in a superscript produces \times , and both a . and a * produce a little filled circle. In the Kröger-Vink notation a prime denotes a negative relative charge, the circle a positive relative charge, and the cross denotes a neutral relative charge.

```
1 \setchemformula{kroeger-vink=true}
2 \ch{Al_{Al}^{'}}
3 \ch{Al_{Al}^{'}}\par
4 \ch{Ni_{Cu}^{\times}}\par
5 \ch{V_{Cl}^{\bullet}}
6 \ch{V_{Cl}^{\bullet}}\par
7 \ch{Ca_i^{\bullet}}\par
8 \ch{e^{'}}\par
9 \ch{Cl_i^{'}}
10 \ch{Cl_i^{'}}\par
11 \ch{O_i^{'}}
12 \ch{O_i^{'}}
```

$$\begin{array}{l} Al'_{Al} \quad Al'_{Al} \\ Ni^{\times}_{Cu} \\ V^{\bullet}_{Cl} \quad V^{\bullet}_{Cl} \\ Ca^{\bullet}_i \\ e' \\ Cl'_i \quad Cl'_i \\ O''_i \quad O''_i \end{array}$$

There are a number of options for customizations:

`kv-positive-style = {<TikZ>}` (initially empty)
TikZ code for positive charge dot.

`kv-positive-radius = {<dim>}` Default: .3ex
Radius of positive charge dot

`kv-positive-hshift = {<dim>}` Default: .15em
Horizontal shift of positive charge dot

`kv-positive-vshift = {<dim>}` Default: .5ex
Vertical shift positive charge dot

`kv-positive-offset = {<dim>}` Default: .4em
The offset of two consecutive positive charge dots

`kv-neutral-symbol = {<TeX code>}` Default: \times
Symbol for neutral particles.

A. History Since Version 4.0

Version 4.0

Introduced in
version 4.0

- Since version 4.0, the **CHEMFORMULA** package is distributed independently from **CHEM-MACROS**.

Version 4.1

- New option `tikz-external-disable`.
- New option `frac-math-cmd`.

Version 4.2

- New option `arrow-style`.
- New command `\chlewis` that allows to add Lewis electrons to an atom, see section 14.

Version 4.3

- New option `stoich-print`.
- New command `\chstoich`.
- The commands `\DeclareChem{...}` now don't give an error any more if the command already exists. This is more consistent with \LaTeX 's `\DeclareRobustCommand`. For all those commands a version `\NewChem{...}` is introduced that *does* give an error if the new command is already defined.

Version 4.4

- A single dash - in `\ch` is now treated as a minus sign. This is consistent with the behaviour of a +.

Version 4.5

- New arrow types `>=<`, `>=<<`, `>>=<` and `<==>`.
- Internal changes to `\ch` allow usage of optional arguments of `\` and `\label` in `CHEM-MACROS`' reactions environment.

Version 4.6

- New options `circled` and `circletype`. this allows to set the behaviour as described on `CHEMMACROS`' manual for a specific usage of `\ch`.

Version 4.7

- Dependency change: `CHEMFORMULA` now requires the TikZ library `arrows.meta` instead of the library `arrows`. This requires TikZ version 3.0.0.

Version 4.8

- The `CHEMFORMULA` package now is no longer part of the `CHEMMACROS` bundle but is distributed as a package of it's own.

Version 4.9

- New options `minus-space` and `minus-penalty` equivalent to the existing `plus-space` and `plus-penalty`
- New options `plus-output-symbol` and `minus-output-symbol` for customizing the plus and minus signs in the output.

Version 4.10

- New macro set `\NewChemCompoundProperty`, see section 6.8 for a description.

Version 4.11

- New macro set `\NewChemAdditionSymbol`, see section 7.1.1.
- New macro set `\NewChemSymbol`, see section 7.1.2.

Version 4.12

- Change package requirement: `CHEMFORMULA` now not loads complete `amsmath` but only `amstext`.

Version 4.13

- Check for blank input parts and don't process them.
- Drop support for `\[` and `\]` as replacement for `[` and `]` inside arrow captions.
- New option `atom-format`.

Version 4.14

- New option `adduct-penalty`.

Version 4.15

- The order of arrow definitions doesn't matter any more.
- New quasi equilibria arrows.
- Require `amsmath`.
- Allow `name-format` to end with a macro that takes an argument.
- Various bug fixes.

B. Index

Symbols

-/> (arrow type) 21
-> (arrow type) 21
<- (arrow type) 21
<-> (arrow type) 21
</- (arrow type) 21
<==> (arrow type) 22, 35
<=> (arrow type) 21
<=>> (arrow type) 21
<=0> (arrow type) 22
<=0>> (arrow type) 22
<> (arrow type) 21
<0> (arrow type) 22
<<=> (arrow type) 21
<<=0> (arrow type) 22
== (arrow type) 21
><< (arrow type) 21, 35
><<< (arrow type) 21, 35
>>=< (arrow type) 22, 35

A

adduct-penalty 13, 36
adduct-space 13, 15
amsmath (package) 36
amstext 2
amstext (package) 2, 36
arrow-min-length 23
arrow-offset 23
arrow-penalty 24
arrow-ratio 24
arrow-style 24 ff., 35
arrow-yshift 23
atom-format 29, 36

B

\bond 10 ff., 15 f.
bond-length 12 f.
bond-offset 13
bond-penalty 13
bond-style 13

C

\ce 3
\ch 3–20, 22 ff., 26–29, 31–35
charge-hshift 10, 13 f.
charge-style 13
charge-vshift 13
\charrow 27
\chcpd 16
chemfig 24, 32
chemfig (package) 24 f., 32

chemformula 2
\chemformula@bondlength 12
\chemsetup 2 f., 9
\chlewis 32 f., 35
\chname 28
\chstoich 5, 35
circled 9, 13, 35
circletype 9, 13, 35
COHAN, E. Richard 5, 10, 14
compound-sep 24
CTAN 2, 31

D

decimal-marker 5, 8
\DeclareChemAdditionSymbol 17
\DeclareChemArrow 25
\DeclareChemBond 11
\DeclareChemBondAlias 11
\DeclareChemCompoundProperty 16
\DeclareChemSymbol 18

F

FEUERSÄNGER, Christian 31
font-family 30
font-series 30
font-shape 30
font-spec 30
fontspec (package) 30
format 29
\fplus 9
frac-math-cmd 5, 35
frac-style 5 f.

H

HENSEL, Martin 2

K

KOHM, Markus 2
KOMMA-Script 2
kroeger-vink 33
kv-neutral-symbol 34
kv-positive-hshift 34
kv-positive-offset 34
kv-positive-radius 34
kv-positive-style 34
kv-positive-vshift 34

L

l3kernel 2
l3kernel (bundle) 2
l3keys2e (package) 2

l3packages 2
l3packages (bundle) 2
label-offset 24
label-style 24
lewis-default 32
lewis-distance 32
lewis-line-length 32
lewis-line-width 32
lewis-offset 32
LPPL 2

M

math-space 20
\mch 9
mhchem (package) 2 f., 21
minus-output-symbol 18, 36
minus-penalty 18, 36
minus-space 18, 36
MITTELBACH, Frank 2

N

name-format 28, 36
name-width 28
\NewChemAdditionSymbol 17 f., 36
\NewChemArrow 25 f.
\NewChemBond 11 f.
\NewChemBondAlias 11 f.
\NewChemCompoundProperty 16 f., 36
\NewChemSymbol 18, 36
nicefrac 2
nicefrac (package) 2, 6

O

\ox 8, 20, 23

P

pgfmanual (manual) 25
pgfplots 31
pgfplots (package) 31
plus-output-symbol 18, 36
plus-penalty 18, 36
plus-space 18, 36
\ProvideChemAdditionSymbol 17
\ProvideChemArrow 25
\ProvideChemBond 11
\ProvideChemCompoundProperty 16
\ProvideChemSymbol 18

Q

“Quantities, Symbols and Units in
Physical Chemistry”

INDEX

<i>IUPAC Green Book</i> . 5, 10, 14	<code>\RenewChemCompoundProperty</code> . 16	<code>subscript-vshift</code> 12, 15
	<code>\RenewChemSymbol</code> 18	
R	S	T
<code>radical-hshift</code> 13	SCHÖPF, Rainer 2	TANTAU, Till 2
<code>radical-radius</code> 13, 33	scrfile (package) 2	TELECHEA, Christian 24, 32
<code>radical-space</code> 13	<code>\setchemformula</code> 2, 9, 12, 14 f., 20, 29 f., 32, 34	THE L ^A T _E X3 PROJECT TEAM 2
<code>radical-style</code> 13	<code>\ShowChemArrow</code> 25 f.	tikz (package) 2
<code>radical-vshift</code> 13	<code>\ShowChemBond</code> 11	<code>tikz-external-disable</code> 31, 35
REICHERT, Axel 2	<code>stoich-paren-parse</code> 5	<i>TikZ/pgf</i> 2
<code>\RemoveChemCompoundProperty</code> . 16	<code>stoich-print</code> 5, 35	
<code>\RenewChemAdditionSymbol</code> 17	<code>stoich-space</code> 5 f.	X
<code>\RenewChemArrow</code> 25 f.	<code>subscript-style</code> 12	xfrac (package) 2, 5
<code>\RenewChemBond</code> 11		xparse (package) 2