

CHEMNUM

v0.4a 2012/02/26

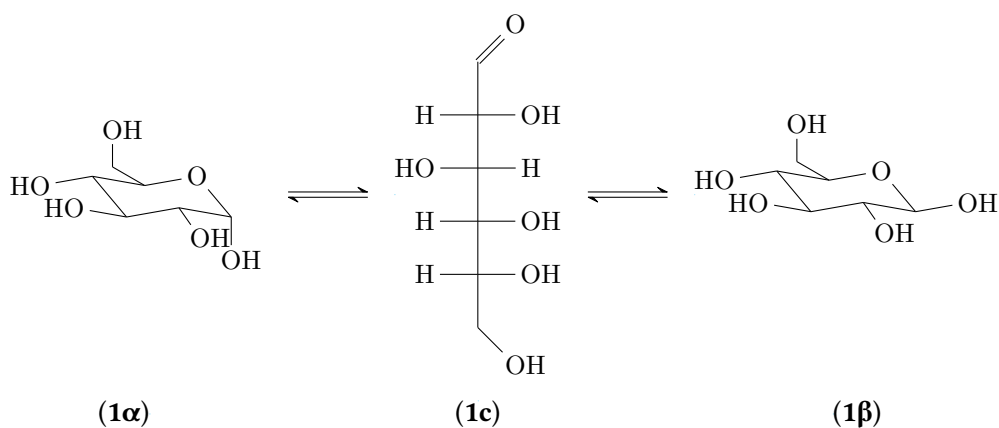
Clemens NIEDERBERGER

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

contact@mychemistry.eu

An approach for the numbering of chemical compounds with $\text{\LaTeX 2}_{\epsilon}$.

English documentation



Scheme 1: In a solution the α - and the β -anomer of D-glucose (**1 α** and **1 β**) are in equilibrium via the open-chain form (**1c**).

Contents		9 Usage with hyperref	16
1 A very short readme	2	9.1 Basics	16
2 News and a word of apology	3	9.2 Selected usage	17
3 Package options	4	10 Customization	17
4 Motivation	4	10.1 The principle	17
5 Numbering compounds	5	10.2 Font style	18
5.1 Main command	5	10.3 Own labels	19
5.2 Sublabel	6	10.4 Own markers in images	19
5.3 Lists	7	10.5 Counter	19
5.4 Lists and ranges of sublabels . .	7	10.6 Delimiters	20
6 Initialize labels at once	9	10.7 Suffix and prefix	21
6.1 Usage	9	10.8 Separators	22
6.1.1 Main labels	9	10.8.1 Labels – input	22
6.1.2 Sublabels	10	10.8.2 Labels – output	23
6.2 Error instead of warning	10	10.8.3 Sublabels – input	23
6.3 Customized input	11	10.8.4 Sublabels – output	24
6.4 Why initiate labels?	11	10.8.5 Lists – input	24
7 Resetting the counter	11	10.8.6 Lists – output	25
8 Special usage	12	10.8.7 Lists – last separator . .	25
8.1 Moving arguments – section titles	12	10.8.8 \cnpdinit – input	26
8.1.1 Test section title with compound label (1a) . .	12	10.8.9 An example	26
8.1.2 Test section title with compound label (1b) . .	12	11 Language support	27
8.2 Floating environments	13	11.1 Cooperation with babel	27
8.3 Schemes created by external programs	13	11.2 Harvard comma	27
		12 Overview	28
		12.1 Keys	28
		12.2 Commands	29
		13 Credits	30
		Index	31

1 A very short readme

CHEMNUM v0.4a underlies the L^AT_EX Project Public License Version 1.3 or later.
(<http://www.latex-project.org/lppl.txt>)

CHEMNUM internally loads the packages expl3 and xparse¹. xparse is part of the l3packages² bundle, expl3 is part of the l3kernel³ bundle. This means, CHEMNUM needs L^AT_EX3 Support.

¹CTAN: xparse

²CTAN: l3packages

³CTAN: l3kernel

`CHEMNUM` also needs the packages `etoolbox`⁴, `psfrag`⁵ and either `textgreek`⁶ or `bm`⁷, see section 3.

In this command you can easily distinguish between a `\command`, a `key` belonging to a key/value pair and other code.

2 News and a word of apology

In the update to v0.4 there have been some changes made with the keys. There are a few new ones and some of the old ones got new names. They now have consistent names which in my opinion better reflect their function. This means that your documents set with an older version of `CHEMNUM` might not compile any more without errors. I am sorry for that. `CHEMNUM` still is actively developed. Once it reaches v1.0, such changes won't be made any more⁸.

These keys have new names:

old name	new name
<code>compd-sep</code>	<code>sub-output-sep</code>
<code>sub-marker</code>	<code>sub-input-sep</code>
<code>sub-range-sep</code>	<code>subrange-output-sep</code>
<code>sub-range-marker</code>	<code>subrange-input-sep</code>
<code>strict</code>	<code>init-strict</code>
<code>sub-init</code>	<code>init-sub</code>

These ones are new:

key	short info
<code>sublist-output-sep</code>	list separation in the output of labels like 1a,b , see page 25.
<code>sublist-input-sep</code>	list separation in the input of labels like 1a,b , see page 24.
<code>init-input-sep</code>	input in <code>\compdinit</code> , see page 26.
<code>hyper-use</code>	usage of hyperlinks, see page 16.
<code>hyper-target</code>	
<code>hyper-link</code>	
<code>ref-style</code>	Output of labels with <code>\compdref</code> , see pages 13 and 18.

Also new is the way how lists and ranges of sublabels are input. The key `sub-list` no longer exists. More on this in section 5.4.

Then there is the possibility of creating hyperlinked labels together with the package `hyperref`⁹, see section 9.

With v0.4 this documentation got an index.

⁴CTAN: `etoolbox`

⁵CTAN: `psfrag`

⁶CTAN: `textgreek`

⁷CTAN: `bm`

⁸Or only in absolutely necessary cases.

⁹CTAN: `hyperref`

3 Package options

`CHEMNUM` has one package option.

`\usepackage[textgreek = <value>]{chemnum}`

With this you can both choose between the three different styles of the `textgreek` package¹⁰ or load `CHEMNUM` without the `textgreek` package. Available values are `artemisia`, `cbgreek`, `euler` and `false`. Using `\usepackage[textgreek]{chemnum}` is the same as `\usepackage[textgreek = cbgreek]{chemnum}`. If you use `CHEMNUM` without option `textgreek` is not used. This corresponds to `\usepackage[textgreek = false]{chemnum}`, the default setting.

This document was set with `\usepackage[textgreek]{chemnum}`.

4 Motivation

As far as I know there are three packages that are meant to help with numbering chemical compounds. All of them have their weaknesses.

The first one – `chemcono`¹¹ – redefined bibliography commands for that purpose. Compounds have to be declared in what is called `\theffbibliography`. Then one can reference them with `\fcite`. However, it produces a list of compounds in the text. So the package author suggests:

After compilation and printout, discard the last page.

Stefan Schulz

Obviously that's not a perfect solution.

The second one – `chemcompounds`¹² – was written, because the author didn't want to work with the weaknesses of `chemcono` any more. When he wrote the package he basically used the same mechanism to create the labels as `chemcono` did.

When taking a closer look at the `chemcono` package, I realised that the only thing one has to do is to get rid of everything which produces text. Thus, as a basis I used the mechanism of `\bibitem` and `\cite` in pretty much the same way as `chemcono` does by extracting the corresponding code from `article.cls` and `latex.ltx` but deleting any unnecessary commands producing output. I also introduced several lines of code to make the printing of the compound names more customisable.

Stephan Schenk

Some points still left me unsatisfied, though:

1. Compounds usually need to be declared with `\declarecompound`. They need to be declared in any case if you need a label like **1a**. Then, one even needs to choose the label by hand, what somehow undermines the automatic numbering principle.
2. The layout can't be changed for a single label but only for all.

¹⁰See the `textgreek` documentation for details.

¹¹CTAN: `chemcono`

¹²CTAN: `chemcompounds`

3. The numbers can't be reset. *Although in most cases this is neither necessary nor can it be recommended*, there can be individual cases where this would be useful.
4. A list of several compounds `\compound{a,b,c}` can only be customized with more effort than what would be convenient.

Then there is `bpchem`¹³, which provides commands similar to `\label \ref: \CNlabel{}`, `\CNlabelnoref{}` and `\CNref{}`. It provides commands for sublabels, too: `\CNlabelsub{}`, `\CNlabelsubnoref{}` and `\CNrefsub{}`. This makes it more flexible than the others regarding sublabels. However, it barely provides possibilities to customize the labels, lists are not possible and the fact that there are different commands for labels and sublabels isn't the best solution, either.

`CHEMNUM` is intended to fill these gaps. For this all commands have been written from scratch. Some of the ideas of `chemcompounds` e. g. regarding delimiters and layout have been picked up, though.

If you notice any feature missing, please let me know by sending me an email.

5 Numbering compounds

5.1 Main command

The main command of this package is this one:

```
\cmpd{<label name>}
```

When `<label name>` is used the first time, the label is created, saved (= declared) and printed. Each further use just prints the label.

```
1  Compounds \cmpd{a} and \cmpd{b} are declared and can be used any time: \cmpd{a}. No
    pre-declaring is necessary. Compounds like \cmpd{c} are numbered in the order
    they appear in the text.\par
2  Once again: \cmpd{b}, \cmpd{a}, \cmpd{c}.
```

Compounds 2 and 3 are declared and can be used any time: 2. No pre-declaring is necessary.
Compounds like 4 are numbered in the order they appear in the text.
Once again: 3, 2, 4.

If it is necessary to declare a compound without printing the label, this is possible with

```
\cmpd*{<label name>}
```

This declares the label but doesn't print anything. **This command is not needed when `\cmpdinit{}` is used, see section 6 (pages 9ff).**

¹³CTAN: `bpchem`

1 The hidden version `\cmpd*{d}` declares the label but doesn't print anything. The next `\cmpd{e}` continues to count with the next number. With `\cmpd{d}` the label can be used, of course.

The hidden version declares the label but doesn't print anything. The next **6** continues to count with the next number. With **5** the label can be used, of course.

You can pretty much use what you like for a label name except for the separator symbols (see sections 5.2, 5.3, 5.4 and 10.8). You shouldn't leave blanks in the name, though. This may not cause an error at all, but *could* declare different compounds with the same name.

1 `\cmpd{aa_}, \cmpd{a_a_}, \cmpd{a_aa}, \cmpd{_a_a}, \cmpd{__aa}` and `\cmpd{aa_}` all have the same label. As well as `\cmpd{aa_}, \cmpd{a_a}, \cmpd{aa}` and `\cmpd{aa}`.

7, 7, 7, 7, 7 and **7** all have the same label. As well as **7, 7, 7** and **7**.

5.2 Sublabel

If you want a label like **2a**, you need to use the following syntax:

```
\cmpd{<label name>.<subname>}
```

<label name> is the main name which stays the same, <subname> varies. This syntax means that the point . *cannot* be a part of <label name> or <subname>. Instead of the point you also can use another symbol, see section 10.8.1.

1 `\cmpd{f.one}` and `\cmpd{f.two}` are related, as are `\cmpd{g.one}` and `\cmpd{g.two}`. Of course, these labels can be used again: `\cmpd{g.two}` and `\cmpd{f.one}`.

1a and **1b** are related, as are **8a** and **8b**. Of course, these labels can be used again: **8b** and **1a**.

This also works, if the main name has already been used.

1 `\cmpd{a}` and its variants `\cmpd{a.one}` and `\cmpd{a.two}`

2 and its variants **2a** and **2b**

The same way the main name of combined labels can be used solely.

1 `\cmpd{f}` and `\cmpd{g}`

1 and **8**

How you can create a label like **1a,b** is explained in section 5.4.

5.3 Lists

There is actually more to the `\cmpd` command. It also prints lists of labels.

The right description would be something like:

`\cmpd{<(possibly comma separated list of) label name(s)>}`

This means that with default settings the comma can't be part of the label name unless hidden in braces. You can change that, see section 10.8.5 on page 24. Section 10.8.9 shows an example how a changed input could look like.

```
1 More than one label can be put inside \cmd{cmpd}, separated by commas. Then a list
  like \cmpd{a, b, c, e, g.two} is printed.
```

More than one label can be put inside `\cmpd`, separated by commas. Then a list like **2, 3, 4, 6, and 8b** is printed.

The Harvard comma (see 11.2) in , and between **6** and **8b** suggests that there are options to customize the list, see sections 10.8 and 11.

5.4 Lists and ranges of sublabels

Sometimes it can be useful to display a label with a list or a range of sublabels. Suppose you have compounds **9a**, **9b**, **9c**, **9d**, and **9e** which for example differ in their substituents. It can be useful to refer to them all at once: **9a–e**.

The syntax is rather intuitive – you just input a list of sublabels:

```
1 list of labels: \cmpd{q.one, q.two, q.three, q.four, q.five}\\
2 label with list of sublabels: \cmpd{q.{one,two,three,four,five}}
```

list of labels: **9a, 9b, 9c, 9d, and 9e**
label with list of sublabels: **9a,b,c,d,e**

Since the sublist is input with a comma in the default setting, you have to put them into braces.

Using this syntax you also can create ranges of sublabels. As a marker to indicate the range `CHEMNUM` uses two points ...

1	<code>\cmpd{q.{two..four}}\\</code>	9b–d
2	<code>\cmpd{q.{one,three..five}}\\</code>	9a,c–e
3	<code>\cmpd{q.{one..three,five}}</code>	9a–c,e

This means you can neither use the comma nor the two points as part of a sublabel name. How you can use other markers for the input you can read in sections 10.8.3 and 10.8.5.

`CHEMNUM` provides a key (see section 10 for more details), with which all sublabels of a main label can be printed as range.

`compd-all = true/false`

type: boolean

```
1 \compd[compd-all]{q}
```

9a-e

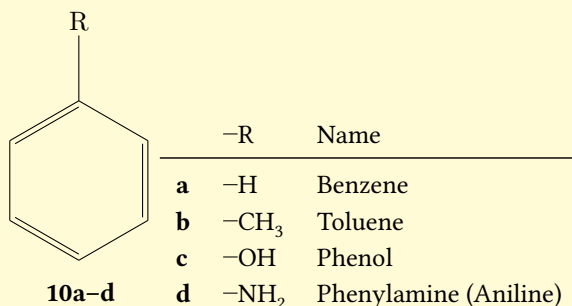
In order to get the right sublabels the command needs to know, which sublabels have been declared. This means that they're read from the <jobname>.compd file (supposing your source file is called <jobname>.tex). So you may need to compile twice (or more) until the labels are printed right.

If one uses label with sublabel ranges, one might want to use *only* the sublabels in a different place. This is possible with this key:

`sub-only = true/false`

type: boolean

```
1 % uses packages 'chemfig', 'chemmacros' and 'booktabs'
2 \compd*{benzene.H,benzene.Me,benzene.OH,benzene.NH2}
3 \chemname{\chemfig{*6(=--(-R)=)}}{\compd[compd-all]{benzene}}
4 \begin{tabular}{lll}
5
6 & & & & \ch{-R} & & Name \\
6 \compd[sub-only]{benzene.H} & & \ch{-H} & & Benzene \\
7 \compd[sub-only]{benzene.Me} & & \ch{-CH3} & & Toluene \\
8 \compd[sub-only]{benzene.OH} & & \ch{-OH} & & Phenol \\
9 \compd[sub-only]{benzene.NH2} & & \ch{-NH2} & & Phenylamine (Aniline)
10 \end{tabular}
```



10a-d

Ranges of sublabels require some care. For one thing one has to know which sublabel is the lower one and which is the higher one. Then one has to know which labels are in between them and what do they represent. As **CHEMNUM** can't take care for you, you need to do that for yourself if you want to avoid labels like these:

```
1 \compd{q.{five..three}} or \compd{q.{three,one,four}}.
```

9e-c or 9c,a,d.

6 Initialize labels at once

The commands described in this section are not essential for the usage of **CHEMNUM**, but provide additional functionality that some users might find useful.

6.1 Usage

6.1.1 Main labels

Perhaps you're missing chemcompounds' command `\declarecompound` already. For one thing one has a summary of all used label names. And one may be warned if one uses an uninitialized label name¹⁴. This behaviour can be realized with these commands:

```
\cmpdinit{<comma separated list of label names>} \cmpdinit*{<comma separated  
list of label names>}
```

All compounds set inside `\cmpdinit{}` are initialized and declared *in the given order*. If one uses a label name that's not in the list, **CHEMNUM** produces a warning message.

```
1 \cmpdinit{A, B, C} \cmpd{B} \cmpd{A.a} \cmpd{C} \cmpd{D}
```

2 1a 3 4

```
*****  
* chemnum warning: "cmpd-init"  
*  
* You used \cmpdinit but didn't initiate compound "D" on line 1.  
*****
```

Please note that for labels with sublabels only the *main name* can and should be initialized. This initializes all compounds with the same main name.

`\cmpdinit*{}` also produces a warning but doesn't declare the labels. They are declared when you first use them in the order of appearance in the text.

```
1 \cmpdinit*{X, Y, Z} \cmpd{Y} \cmpd{X.x} \cmpd{Z} \cmpd{W}
```

1 2a 3 4

```
*****  
* chemnum warning: "cmpd-init"  
*  
* You used \cmpdinit but didn't initiate compound "W" on line 1.  
*****
```

¹⁴for instance due to a typing error

If you need to use own labels (see page 19) but still want to initialize all labels at once, you should use `\cmpdinit*`, because then no labels are declared when they're initialized.

This command can be used several times, with or without *. A useful way would be a single usage, though, proposedly in the preamble of your document after the `CHEMNUM` setup (see section 10).

6.1.2 Sublabels

If you'd like a warning for label-sublabel combinations as well, or want to preset their order, you can use

```
\cmpdinit[init-sub = true]{<comma separated list of label names>}
```

This will give you a warning in these cases, too.

```
1 \cmpdinit[init-sub = true]{E.{e,f,g}}      1a and 1d
2 \cmpd{E.e, E.h}
```

```
*****
* chemnum warning: "cmpd-init-sub"
*
* You used \cmpdinit and "init-sub = true" but didn't initiate sub-compound
* "E.h" on line 1.
*****
```

6.2 Error instead of warning

If you want `CHEMNUM` to produce an error instead of a warning when you use an uninitialized label name, you can use `\cmpdinit` like this:

```
\cmpdinit[init-strict = true]{<comma separated list of label names>}
```

A usage of `init-strict = false` would be the usage without optional argument. Of course the *-version has the same option.

With `init-strict = true` the \LaTeX run will be aborted with a corresponding error message, if a label name is used that hasn't been initialized before.

Of course you can use `\cmpdinit` with both keys

```
\cmpdinit[init-strict = true, init-sub = true]{<comma separated list of label
names>}
```

or use `\cmpdsetup`, see section 10.

6.3 Customized input

When you're using other tokens for `sub-input-sep` (page 22) or `sublist-input-sep` (page 24), you have to use them with `\cmpdinit` as well.

A change of `list-input-sep` (page 24) doesn't affect `\cmpdinit`. If you want to change the input of `\cmpdinit` you need to use `init-input-sep`, see section 10.8.8 on page 26.

6.4 Why initiate labels?

As mentioned in the beginning of this section the initiation of labels is not required. There are two possible reasons, why one would like to use them, anyway.

1. The use of `\cmpdinit*` is a way to keep track of which labels you have used and gives you a warning/error message if you use a new one or just have misspelled an existing one. Depending on the number of labels you (have to) use, this can be lots of additional work, though.
2. The use of `\cmpdinit` does the same thing as `\cmpdinit*` and declares all the labels in the given order. So if you use it, you won't have to use `\compd*` (section 5.1) and `\compd+` (section 8.1). This gives you direct control, which number is given to which compound.

7 Resetting the counter

With the command

`\cmpdreset[<number>]`

it is possible to reset the counter. Without argument it is reset to 1. So `\cmpdreset` and `\cmpdreset[1]` are the same. Unlike most other `CHEMNUM` commands its impact is not only inside the local \TeX group but resets the counter globally.

```
1 \cmpdreset The numbering starts with 1 again: \compd{h, i, j}
```

The numbering starts with 1 again: **1**, **2**, and **3**

You should be careful with how you use this command! Usually it cannot be recommended to reset the counter. Different compounds can get the same label!

```
1 Same label: \compd{a}, \compd{h}
```

Same label: **2**, **1**

8 Special usage

8.1 Moving arguments – section titles

Using `\cmpd` inside a section title in the first instant doesn't seem to be a problem. This doesn't give any errors and the output looks like expected:

```
1 \section{Compound \cmpd[compd-delim]{b}}
2 \ldots
```

However, if you use `\tableofcontents` – which is rather probable, I guess – something unwanted happens: because the table of contents is at the beginning of the document, the label is declared in the table of contents instead of at the first appearance in the text. There are two ways to circumvent this:

1. You declare your labels in the order you want with `\cmpdinit{}` = (see page 9) in the preamble or the beginning of the document.
2. You use `\cmpd+` to set the label. It will read the label from the file `<jobname>.cmpd` (supposing your main file is called `<jobname>.tex`). This means *at least two*, maybe more \LaTeX runs are necessary until all labels are set.

The same is true for any other list of moving arguments that is created *before* the actual appearance of the label in the text.

The command

`\cmpd+{<label name>}`

reads the label name that is to be printed from `<jobname>.cmpd`. In a way this is the opposite of `\cmpd*` since it doesn't declare a label but only references and prints it. So in order to use `\cmpd+{<label>}`, `<label>` must have been declared, either with `\cmpd{<label>}` or with `\cmpd*{<label>}`. It doesn't matter, if this declaration is done before or after the usage of `\cmpd+`. This command is useful, if the label is used inside a moving argument like in `\section{}`. **It is not needed when `\cmpdinit{}` is used, see section 6 (pages 9 ff).**

```
1 \subsubsection{Test section title with compound label \cmpd+[compd-delim]{f.one}}
2 % with hyperref:
3 \subsubsection{Test section title with compound label \texorpdfstring{\cmpd+[compd-
  delim]{f.two}}{(7b)}}
```

8.1.1 Test section title with compound label (1a)

8.1.2 Test section title with compound label (1b)

8.2 Floating environments

The usage of `\cmpd` inside a float shouldn't be a problem: labels are declared in the order the floating environment appears in the code. However, to be on the safe side you can of course use `\cmpd+` here as well.

```
1 % preamble:
2 % \usepackage{chemscheme}
3 % document:
4 \cmpdreset\cmpd{float1, float2}. Now inside of a scheme:
5 \begin{scheme}[h]
6   \centering
7   \caption{This label should be a 2: \cmpd{float2}}
8   If 2 = \cmpd{float2} is true, everything's fine.
9 \end{scheme}
```

1 and 2. Now inside of a scheme:

If 2 = 2 is true, everything's fine.

Scheme 2: This label should be a 2: 2

This also works, if the floating environment is placed before the code, for instance by using the option `[t]`.

8.3 Schemes created by external programs

If you create your schemes by external programs like for e.g. CHEMDRAW or MARVINSKETCH, and want to use `CHEMNUM` with them as well, you can use this command:

`\cmpdref[<keyval>][<tag>]{<label name>}`

This command is inspired by the `\schemeref` command from the package `chemscheme`¹⁵. It works in a very similar way. You create the scheme and save it with temporary labels as an eps file.

```
1 \begin{scheme}[ht]
2   \centering
3   \includegraphics{scheme-tmp}
4   \caption{Scheme with temporary markers.}
5 \end{scheme}
6 \begin{scheme}[ht]
7   \centering
8   \cmpdref{Alc} % replaces TMP1
9   \cmpdref{EtherBr} % replaces TMP2
```

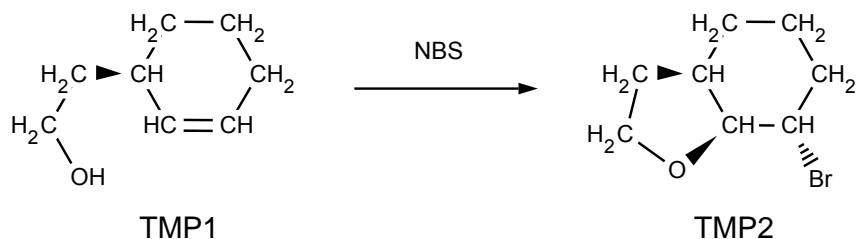
¹⁵CTAN: `chemscheme`

```

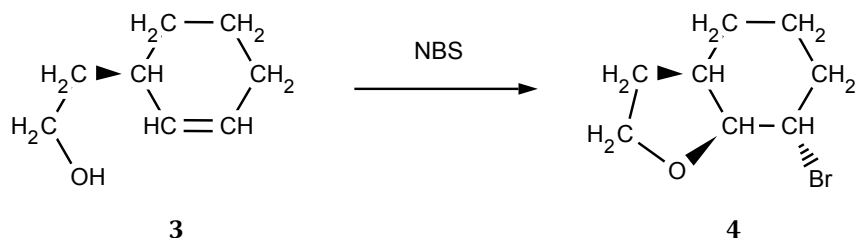
10 % \cmpdref{third} would replace TMP3
11 \includegraphics{scheme-tmp}
12 \caption{Scheme with automated labels.}
13 \end{scheme}

```

So you nummerate the compounds with TMP1, TMP2 etc. These markers will then be replaced



Scheme 3: Scheme with temporary markers.



Scheme 4: Scheme with automated labels.

with the corresponding labels. Like chemscheme [CHEMNUM](#) uses the `\psfrag` command of the `psfrag` package to do that. This means you have to either compile via L^AT_EX, D_VI_PS, P_S2P_DF, or for instance use the package `auto-pst-pdf`¹⁶ to be able to use P_DF_LA_TE_X.

In each case it is important that the marker is saved as *text* in the eps file.

You can customize the labels with the keys presented in section 10.

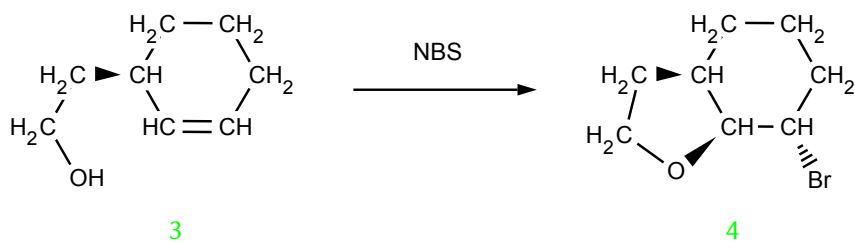
```

1 \begin{scheme}[ht]
2 \centering
3 \cmpdref[compd-style=\bf\sf\textcolor{green}]{Alc} % replaces TMP1
4 \cmpdref[compd-style=\bf\sf\textcolor{green}]{EtherBr} % replaces TMP2
5 \includegraphics{scheme-tmp}
6 \caption{Scheme with automated labels.}
7 \end{scheme}

```

You also can replace arbitrary text by using the second optional argument.

¹⁶CTAN: `auto-pst-pdf`

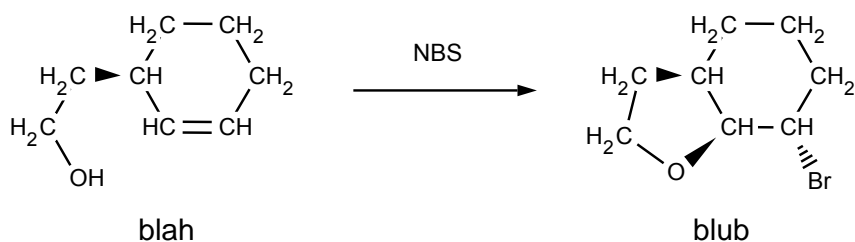


Scheme 5: Scheme with automated labels.

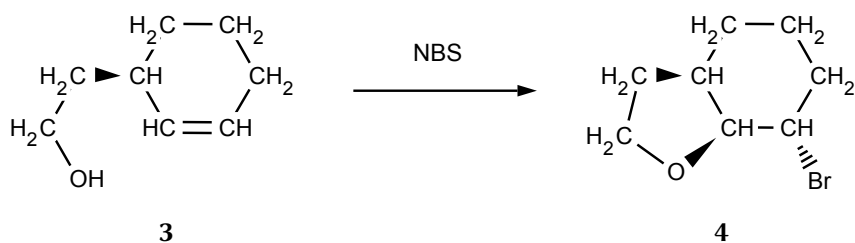
```

1 \begin{scheme}[ht]
2 \centering
3 \includegraphics{scheme-bla}
4 \caption{Scheme with arbitrary markers.}
5 \end{scheme}
6 \begin{scheme}[ht]
7 \centering
8 \cmpdref[][blah]{Alc}
9 \cmpdref[][blub]{EtherBr}
10 % \cmpdref{third} would replace TMP1
11 \includegraphics{scheme-bla}
12 \caption{Scheme with explicitly placed labels.}
13 \end{scheme}

```



Scheme 6: Scheme with arbitrary markers.



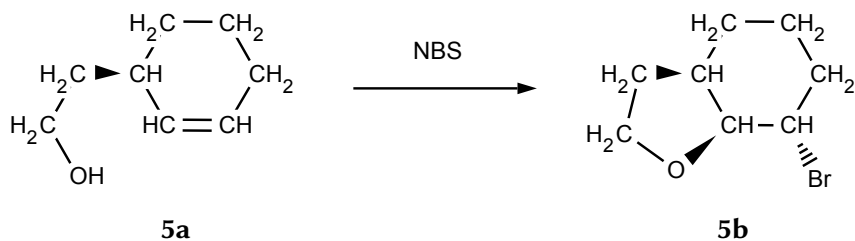
Scheme 7: Scheme with explicitly placed labels.

Of course sublabels work in the known way:

```

1 \begin{scheme}[ht]
2   \centering
3   \cmpdref{cpd.red}
4   \cmpdref{cpd.ox}
5   \includegraphics{scheme-tmp}
6   \caption{Scheme with automated labels and sublabels.}
7 \end{scheme}

```



Scheme 8: Scheme with automated labels and sublabels.

9 Usage with hyperref

9.1 Basics

If you are using the package `hyperref` you can make hyperlinked labels in your PDF.

<code>hyper-use = true/false</code>	type: boolean
-------------------------------------	---------------

Setting this key (see section 10 for more information) to true, the next usage of a label will create a link target and the ones after that will create links to this target. This only works if the label does *not* contain a list or range of sublabels. Also a label set with `\cmpd*` can't be marked as target.

```

1 \cmpdsetup{hyper-use}
2 The first time targets are created: \cmpd{a.two,b,c,d,e,f}. Then links are created:
   \cmpd{a.two,d,e}.
3
4 This doesn't work with labels containing a list of sublabels: \cmpd{a,f.{one,two},c
   }. Since \cmpd{a.two} and \cmpd{a} are different labels they have different
   link targets.

```

The first time targets are created: **2b**, **3**, **4**, **5**, **6**, and **1**. Then links are created: **2b**, **5**, and **6**. This doesn't work with labels containing a list of sublabels: **2**, **1a,b**, and **4**. Since **2b** and **2** are different labels they have different link targets.

9.2 Selected usage

Maybe the automated use isn't what you want or need. Maybe it is enough to create a hyperlink in one or two places. This can be achieved with two keys:

<code>hyper-target = <link name></code>	type: literal
<code>hyper-link = <link name></code>	type: literal

With `hyper-target` you can set the target. **Caution:** if you use this key with a list *every* label of the list will get the same link name and `hyperref` will warn you:

destination with the same identifier (name<link name>) has been already used,
duplicate ignored

`hyper-link` creates a link for the denominated target. **Caution:** if you use this key with a list *every* label of the list will refer to the same target.

```
1 Create a link \cmpd[hyper-link=mylink,cmpd-all]{f} but don't forget the
  corresponding target: \cmpd[hyper-target=mylink,cmpd-all]{f}.
```

Create a link **1a-4** but don't forget the corresponding target: **1a-4**.

If you use `hyper-link` without the corresponding `hyper-target` `hyperref` will issue a warning:

name<link name> has been referenced but does not exist, replaced by a fixed
one

10 Customization

10.1 The principle

`CHEMNUM` is customized with a key-value system. They can either be used for all following commands with

```
\cmpdsetup{key1 = value1, key2 = value2}
```

or only for a single command directly with

```
\cmpd[key1 = value1, key2 = value2]{<label name>}
```

There are different kinds of keys. Depending on it they can have different kinds of values.

macro This kind of keys expects one or more macros as value. The last of them may have a mandatory argument, e.g. `\bfseries` or `\textbf`. `CHEMNUM` has only one such key, see section 10.2.

literal These keys use the given value "as it is", i.e. literally. Most of `CHEMNUM`'s keys are literal keys.

choice For these keys there is a list of options from which one must be chosen as a value.

boolean These keys can either have the value true or false.

If you use keys without values like `\cmpdsetup{key1,key2}`, default values are used. These are *not* necessarily empty are *not* necessarily the default setting of `CHEMNUM`. All of `CHEMNUM`'s keys have default values. They are listed in section 12.1 on page 28.

The keys are divided in to different classes, which indicate the scope they're used for. They all follow the form `<class>-key = <value>`:

cmpd These keys change the whole label or the main label.

sub, sublist, subref These keys have an impact on sublabels.

list Keys which change the output of lists.

ref Keys which have an impact on `\cmpdref`.

hyper Keys which have an impact on the usage with `hyperref`.

init Keys which have an impact in the initialization of the labels.

10.2 Font style

With the keys

<code>cmpd-style = <style></code>	type: macro
<code>cmpd-weight = bold/normal</code>	type: choice

you can change the style with which the labels are printed.

```
1 % preamble:
2 % \usepackage[normalem]{ulem}
3 % document:
4 \cmpd{a, b, f.two, k}\ % default
5 \cmpd[cmpd-style = \uline]{a, b, f.two, k}\ % underlined
6 \cmpd{a, b, f.two, k}\ % default again
7 \cmpdsetup{cmpd-style = \itshape\uline}
8 \cmpd{a, b, f.two, k}\ % underlined and in italics
9 \cmpd[cmpd-weight=normal]{a, b, f.two, k}
```

2, 3, **1b**, and **10**
2, 3, **1b**, and **10**
 2, 3, **1b**, and **10**
2, 3, **1b**, and **10**
2, 3, *1b*, and *10*

The separate handling of the bold face is because of the greek labels (see section 10.5). Depending on the package option `textgreek`, the bold face must realized differently.

10.3 Own labels

With the key

<code>cmpd-label = <label></code>	type: literal
---	---------------

one can choose the label associated with the label name.

```
1 Own label: \cmpd[cmpd-label = XYZ]{1}; in a list \emph{all new compounds} get this
  label! \cmpd[cmpd-label = XYZ]{1, a, m}
```

Own label: XYZ; in a list *all new compounds* get this label! XYZ, 2, and XYZ

10.4 Own markers in images

If you don't want to use TMP as temporary markers for `\cmpdref` (see page 13), you can change this. For example into `tmp`:

```
1 \cmpdsetup{ref-marker = tmp}
```

10.5 Counter

You can change the format of the counters, too, if you like.

<code>cmpd-counter = <counter></code>	type: choice
<code>sub-counter = <counter></code>	type: choice

You can choose between arabic, alph, Alph, greek, Greek, roman, Roman and Symbol. Please note: a change of format only affects labels *which are not yet declared*!

```

1 \cmpdsetup{cmpd-counter = Alph, sub-counter = arabic}
2 \cmpd{a, b, f.two, k}\\ % no affect with existing labels
3 \cmpd{n.one,n.two,o.one}\\
4 \cmpd{f.three,f.four}\\ % be careful: the main label already exists and isn't
   changed!
5 \cmpd[cmpd-counter,sub-counter = greek]{p.one,p.two}

```

2, 3, 1b, and 10

K1, K2, and L1

13 and 14

13 α and 13 β

10.6 Delimiters

Both single labels and whole lists can get delimiters. This is done with the keys

<code>cmpd-delim = <odelim><cdelim></code>	type: literal
<code>list-delim = <odelim><cdelim></code>	type: literal
values need <i>two</i> tokens!	

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[cmpd-delim = ()]{a, b, f.two, k}\\ % in braces
3 \cmpd{a, b, f.two, k}\\ % default again
4 \cmpdsetup{cmpd-delim = ()}
5 \cmpd{a, b, f.two, k} % in braces

```

2, 3, 1b, and 10

(2), (3), (1b), and (10)

2, 3, 1b, and 10

(2), (3), (1b), and (10)

Please be aware that the default values (see pages 18 and 28) of `cmpd-delim` and `list-delim` and the default setting of `CHEMNUM` are *not* the same. By default `CHEMNUM` doesn't use delimiters. Also please note that the list delimiters only are used if `\cmpd` contains at least two label names.

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[list-delim = {[}{]}]{a, b, f.two, k}\\ % in braces
3 \cmpd{a, b, f.two, k}\\ % default again
4 \cmpdsetup{list-delim = []}
5 \cmpd{a, b, f.two, k}\\ % in braces
6 \cmpd{a} % NOT a list!

```

2, 3, 1b, and 10

[2, 3, 1b, and 10]

2, 3, 1b, and 10

[2, 3, 1b, and 10]

2

If you generally use labels in braces but want to use a single one without braces, you have several possibilities:

```
1 \cmpdsetup{cmpd-delim = ()}
2 normally: \cmpd{b}, \cmpd{c}, \cmpd{d.one}, but sometimes so \cmpd[cmpd-delim = ]{e}
   or so \cmpd-{e}.
```

normally: (3), (4), (5a), but sometimes so 6 or so 6.

With

```
\cmpd-{\<label name>}
\cmpdref-{\<label name>}
```

the braces of single labels can easily be removed (but not the ones of a list).

```
1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpdsetup{cmpd-delim = (), list-delim = []}
3 \cmpd{a, b, f.two, k}\\ % braced twice
4 \cmpd-{a, b, f.two, k} % braced once
```

2, 3, **1b**, and 10

[(2), (3), (**1b**), and (10)]

[2, 3, **1b**, and 10]

Using `\cmpdsetup{cmpd-delim = }` and `\cmpdsetup{list-delim = }` will restore the default behaviour.

10.7 Suffix and prefix

If you want to, you can also add a prefix and/or a suffix to a compound and/or a list.

<code>cmpd-prefix = <prefix></code>	type: literal
<code>cmpd-suffix = <prefix></code>	type: literal
<code>list-prefix = <prefix></code>	type: literal
<code>list-suffix = <prefix></code>	type: literal

The same rule as for the delimiters applies: list attributes are only used with lists, i. e. at least with two items.

An example for the label attributes:

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[cmpd-prefix = Nr.]{a, b, f.two, k}\\
3 \cmpdsetup{cmpd-prefix = \(\rightarrow\)}
4 \cmpd{a, b, f.two, k}\\
5 \cmpd{a} % NOT a list!

```

2, 3, 1b, and 10

Nr. 2, Nr. 3, Nr. 1b, and Nr. 10

→ 2, → 3, → 1b, and → 10

→ 2

An example for the list attributes:

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[list-prefix = list:]{a, b, f.two, k}\\
3 \cmpdsetup{list-prefix = collection:}
4 \cmpd{a, b, f.two, k}\\
5 \cmpd{a} % NOT a list!

```

2, 3, 1b, and 10

list: 2, 3, 1b, and 10

collection: 2, 3, 1b, and 10

2

Between prefix, label and suffix there is a space inserted as default behaviour:

```

1 \cmpd[cmpd-suffix=sx,cmpd-prefix=px]{k} px 10 sx

```

It can be changed with

`cmpd-space = <space>`

type: literal

```

1 \cmpd[cmpd-suffix=sx,cmpd-prefix=px,cmpd-space={}]{k}
px10sx

```

10.8 Separators

10.8.1 Labels – input

As a default setting **CHEMNUM** use the point . as a marker to separate main labels from sublabels. You can change that as you like. Tokens you *can't* use are , , % # and you *shouldn't* use @. Furthermore you should be careful to use different symbols for the other separators (see the next sections).

`sub-input-sep = <separator>`

type: literal

```
1 \cmpdsetup{sub-input-sep = !}          1a and 8b
2 \cmpd{f!one, g!two}\\                  1a and 8b
3 \cmpd[sub-input-sep= +]{f+one, g+two}\\ 1a and 8b
4 \cmpd[sub-input-sep= ~]{f~one, g~two}\\ 1a and 8b
5 \cmpd[sub-input-sep= &]{f&one, g&two}\\ 1a and 8b
6 \cmpd[sub-input-sep= *]{f*one, g*two}
```

You should – not only for the sake of consistency – decide only *once* in the beginning of your document *before* the `\tableofcontents` or in the preamble, which marker you use. Otherwise, the `\tableofcontents` won't know the marker anymore and your labels might be at least displayed wrong in the table of contents. Also you need to setup the marker *before* you use it with `\cmpd` or `\cmpdinit` etc.

10.8.2 Labels – output

You can separate the labels from the sublabels if you like to.

`sub-output-sep = <separator>`

type: literal

```
1 \cmpd{a, b, f.two, f.three, k}\\ % default
2 \cmpd[sub-output-sep = -]{a, b, f.two, f.three, k}\\
3 \cmpdsetup{sub-output-sep = $\cldot$}
4 \cmpd{a, b, f.two, f.three, k}\\
5 \cmpd[sub-output-sep = :]{a, b, f.two, f.three, k}
```

2, 3, 1b, 13, and 10

2, 3, 1-b, 1-3, and 10

2, 3, 1·b, 1·3, and 10

2, 3, 1:b, 1:3, and 10

10.8.3 Sublabels – input

As a default setting `CHEMNUM` use two points `..` as a marker to detect two sublabels as end and beginning of a range. But you can choose any tokens you like to do that. Tokens you *can't use* are `,` `%` `#` and you *shouldn't* use `@`. Furthermore you should be careful to use different symbols for the other separators.

`subrange-input-sep = <separator>`

type: literal

```

1 \cmpd[subrange-input-sep=]{q.{one:four}}\\
2 \cmpdsetup{subrange-input-sep=--}
3 \cmpd{q.{one--four}}

```

9a-d
9a-d

10.8.4 Sublabels – output

Maybe you don't like the output of **9a-e** and you want it displayed differently? Then you need this key:

<code>subrange-output-sep = <separator></code>	type: literal
--	---------------

```

1 \cmpd[subrange-output-sep={ {\normalfont to} }]{q.{one..four}}\\
2 \cmpdsetup{subrange-output-sep=-}
3 \cmpd{q.{one..four}}

```

9a to d
9a-d

10.8.5 Lists – input

If you like you can change the list input separator for both main label lists and lists of sublabels. In both cases `CHEMNUM` uses the comma , as default. You can change that as you like. Tokens you *can't use* are , % # and *you shouldn't use* @. Furthermore you should be careful to use different symbols for the other separators.

<code>list-input-sep = <separator></code>	type: literal
<code>sub-input-sep = <separator></code>	type: literal

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[list-input-sep = +]{a + b + f.two + k}\\
3 \cmpd{a, b, f.two, k}\\
4 \cmpdsetup{list-input-sep = ;}
5 \cmpd{a; b; f.two; k}

```

2, 3, 1b, and 10
2, 3, 1b, and 10
2, 3, 1b, and 10
2, 3, 1b, and 10

With this you can for example use IUPAC names as label names¹⁷.

¹⁷Thanks to Christina Lüdigg who in a way suggested this feature.


```
1 \cmpdsetup{list-input-sep=;}
2 \cmpd{1,3-dichlorocyclohexanole}
```

14

10.8.6 Lists – output

You can change the separator symbol with which lists are printed as well. **CHEMNUM** uses the comma for both cases in the default setting.

<code>list-output-sep = <separator></code>	type: literal
<code>sub-output-sep = <separator></code>	type: literal

```
1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[list-output-sep = ]{a, b, f.two, k}\\
3 \cmpd{a, b, f.two, k}\\
4 \cmpdsetup{list-output-sep = ;}
5 \cmpd{a, b, f.two, k}
```

2, 3, **1b**, and 10

2 3 **1b** and 10

2, 3, **1b**, and 10

2; 3; **1b**; and 10

10.8.7 Lists – last separator

The last separator in the main label list is a special case. **CHEMNUM** provides two keys with which it can be customized.

<code>list-last-sep = <separator></code>	type: literal
<code>list-lang = <lang></code>	type: choice

There is the choice key `list-lang`, which changes language dependent settings. You can choose between US (default), GB, DE, FR, ES and IT.

```
1 \cmpd[list-lang = US]{a, b, f.two, k} \cmpd[list-lang = US]{a, b}\\
2 \cmpd[list-lang = GB]{a, b, f.two, k} \cmpd[list-lang = GB]{a, b}\\
3 \cmpd[list-lang = DE]{a, b, f.two, k} \cmpd[list-lang = DE]{a, b}\\
4 \cmpd[list-lang = FR]{a, b, f.two, k} \cmpd[list-lang = FR]{a, b}\\
5 \cmpd[list-lang = ES]{a, b, f.two, k} \cmpd[list-lang = ES]{a, b}\\
6 \cmpd[list-lang = IT]{a, b, f.two, k} \cmpd[list-lang = IT]{a, b}
```

2, 3, **1b**, and **10** 2 and 3
 2, 3, **1b** and **10** 2 and 3
 2, 3, **1b** und **10** 2 und 3
 2, 3, **1b** et **10** 2 et 3
 2, 3, **1b** y **10** 2 y 3
 2, 3, **1b** e **10** 2 e 3

As you can see the language US uses the Harvard comma, all other languages don't. You'll find more on language support and the Harvard comma in section 11.

With the key `list-last-sep` you also can set the last separator individually.

```
1 \cmpdsetup{list-lang = GB}%
2 \cmpd[list-last-sep = {and also}]{a, b, f.two, k}\\
3 \cmpd[list-last-sep = ]{a, b, f.two, k}\\
4 \cmpd[list-last-sep = {as well as}]{a, b, f.two, k}\\
5 \cmpd[list-last-sep = empty]{a, b, f.two, k}
```

2, 3, **1b** and also **10**
 2, 3, **1b** **10**
 2, 3, **1b** as well as **10**
 2, 3, **1b**, **10**

`empty` is a special value. If it is used, the separator set with `list-output-sep` is used for the whole list.

```
1 \cmpdsetup{list-last-sep = empty}%
2 \cmpd{a, b, f.two, k}\\
3 \cmpd[list-output-sep = ;]{a, b, f.two, k}\\
4 \cmpd[list-output-sep = {\ und}]{a, b, f.two, k}
```

2, 3, **1b**, **10**
 2; 3; **1b**; **10**
 2 und 3 und **1b** und **10**

10.8.8 `\cmpdinit` – input

When you've changed the separator for `\cmpd` you might want to change it for `\cmpdinit` as well. `CHEMNUM` uses the comma in the default setting. You can change that as you like. Sie können aber eine beliebige andere Eingabe wählen. Tokens you *can't* use are `,` `%` `#` and you *shouldn't* use `@`.

<code>init-input-sep = <separator></code>	type: literal
---	---------------

10.8.9 An example

A small example how a customized input could look like:

```

1 \cmpdsetup{
2   sub-input-sep      = ! ,
3   subrange-input-sep = : ,
4   sublist-input-sep  = / ,
5   list-input-sep     = +
6 }
7 \cmpd{a + f ! one + q ! one / three : five}

```

2, 1a, and 9a,c-e

11 Language support

11.1 Cooperation with babel

CHEMNUM supports the language choice made with the babel¹⁸ package for a number of languages. This is done with the `\translate` command of the translator package, if it is available. If translator cannot be loaded, language support must be done manually, see section 10.8.7. Manual language settings can of course still be done if translator has been loaded.

```

1 \selectlanguage{ngerman}DE: \cmpd{r,s,t}\
2 \selectlanguage{english}EN: \cmpd{r,s,t}\
3 \selectlanguage{british}GB: \cmpd{r,s,t}\
4 \selectlanguage{american}US: \cmpd{r,s,t}\
5 \selectlanguage{french}FR: \cmpd{r,s,t}\
6 \selectlanguage{spanish}ES: \cmpd{r,s,t}\
7 \selectlanguage{italian}IT: \cmpd{r,s,t}

```

DE: 15, 16 und 17

EN: 15, 16, and 17

GB: 15, 16 and 17

US: 15, 16, and 17

FR : 15, 16 et 17

ES: 15, 16 y 17

IT: 15, 16 e 17

If you find a language missing, please send me an email¹⁹ with the corresponding babel option and the right translation, and I will gladly add it.

11.2 Harvard comma

In American English the so-called Harvard comma is quite commonly used. In British English and other European languages it is rather uncommon.

The serial comma (also known as the Oxford comma or Harvard comma, and sometimes referred to as the series comma) is the comma used immediately before a coordinating

¹⁸CTAN: babel

¹⁹contatc@mychemistry.eu

conjunction (usually and or or, and sometimes nor) preceding the final item in a list of three or more items. For example, a list of three countries can be punctuated as either "Portugal, Spain, and France" (with the serial comma) or as "Portugal, Spain and France" (without the serial comma).

Opinions vary among writers and editors on the usage or avoidance of the serial comma. In American English, the serial comma is standard usage in non-journalistic writing that follows the CHICAGO MANUAL OF STYLE. Journalists, however, usually follow the AP Stylebook, which advises against it. It is used less often in British English, where it is standard usage to leave it out, with some notable exceptions such as FOWLER'S MODERN ENGLISH USAGE. In many languages (e.g., French, German, Italian, Polish, Spanish), the serial comma is not the norm and may even go against punctuation rules. It may be recommended in many cases, however, to avoid ambiguity or to aid prosody.

Wikipedia, 2011/11/07

CHEMNUM uses the Harvard comma when American English is used and doesn't, when other languages are used. However, you have the possibility to disable or enable the Harvard comma as you wish:

```
list-serial-comma = true/false
```

type: boolean

```
1 \selectlanguage{ngerman}DE: \cmpd[list-serial-comma]{r,s,t} vs. \cmpd{r,s,t}\
2 \selectlanguage{american}US: \cmpd[list-serial-comma=false]{r,s,t} vs. \cmpd{r,s,t}
```

DE: 15, 16, und 17 vs. 15, 16 und 17

US: 15, 16 and 17 vs. 15, 16, and 17

If you choose the language manually, please note that this resets the default behaviour for that language. This means you have to use `list-serial-comma` after `list-lang`.

```
1 wrong: \cmpdsetup{list-serial-comma=false, list-lang=US}\cmpd{r,s,t}\
2 right: \cmpdsetup{list-lang=US, list-serial-comma=false}\cmpd{r,s,t}
```

wrong: 15, 16, and 17

right: 15, 16 and 17

12 Overview

12.1 Keys

In this section all keys are listed.

key	default	predefined	type
<code>cmpd-style</code>			macro

cmpd-weight	bold	bold	choice
cmpd-label			literal
cmpd-delim ^a	()		literal
cmpd-odelim			literal
cmpd-cdelim			literal
cmpd-prefix			literal
cmpd-suffix			literal
cmpd-space	\penalty \@m\	\penalty \@m\	literal
cmpd-counter	arabic	arabic	choice
cmpd-all	true	false	boolean
sub-only	true	false	boolean
sub-counter	alph	alph	choice
sub-output-sep		,	literal
sub-input-sep	.	.	literal
sublist-output-sep	,	,	literal
sublist-input-sep	,	,	literal
subrange-output-sep	--	--	literal
subrange-input-sep	literal
list-delim ^a	()		literal
list-odelim			literal
list-cdelim			literal
list-prefix			literal
list-suffix			literal
list-input-sep	,	,	literal
list-output-sep	,	,	literal
list-lang	US	US ^b	choice
list-last-sep	and ^c		literal
list-serial-comma	true	true ^c	boolean
ref-tag	TMP	TMP	literal
ref-style		\sffamily	macro
init-strict	true	false	boolean
init-sub	true	false	boolean
init-input-sep	,	,	literal
hyper-use	true	false	boolean
hyper-target			literal
hyper-link			literal

a value needs *two* tokens.

b depends in the usage of babel.

c depends on the value of `list-lang` and the usage of babel.

12.2 Commands

In this section all commands provided by `CHEMNUM` as well as their variants are listed.

command	description
<code>\cmpd[<keyval>]{<labels>}</code>	basic command, declares and prints labels or lists of labels, see pages 5ff.
<code>\cmpd*{<labels>}</code>	invisible, declares label, see page 5
<code>\cmpd-<[<keyval>]{<labels>}</code>	without delimiters, see page 21
<code>\cmpd+-<[<keyval>]{<labels>}</code>	label is read from auxiliary file, see page 12
<code>\cmpdinit[<keyval>]{<labels>}</code>	initialise and declare labels, see pages 9f.
<code>\cmpdinit*-<[<keyval>]{<labels>}</code>	initialise labels, see pages 9f.
<code>\cmpdref[<keyval>][<marker>]{<label>}</code>	replace temporary markers in eps files by labels, see page 13
<code>\cmpdref-<[<keyval>][<marker>]{<label>}</code>	replace temporary markers in eps files by labels without delimiters, see page 21
<code>\cmpdreset[<num>]</code>	reset counter, siehe page 11
<code>\cmpdsetup{<keyval>}</code>	setup CHEMNUM, siehe pages 17ff.

13 Credits

I would like to thank Joseph WRIGHT and Russell HEWITT who provided me with valuable feedback and suggestions to improve CHEMNUM.

Index

Section headings are typeset **bold**, packages sans serif, commands **brown**, and keys **green**.

A

auto-pst-pdf 14

B

babel 27, 29

bm 3

bpchem 5

C

chemcompounds 4 f.

chemcono 4

chemscheme 13 f.

`\cmpd` 5 ff., 12 f., 17, 20, 23, 26

`cmpd-all` 8

`cmpd-counter` 19

`cmpd-delim` 20 f.

`cmpd-label` 19

`cmpd-prefix` 21

`cmpd-sep` 3

`cmpd-space` 22

`cmpd-style` 18

`cmpd-suffix` 21

`cmpd-weight` 18

`\cmpd*` 5, 11 f., 16

`\cmpd+` 11 ff.

`\cmpd-` 21

`\cmpdinit` 3, 5, 9–12, 23, 26

`\cmpdinit*` 9 ff.

`\cmpdref` 3, 13, 18 f.

`\cmpdref-` 21

`\cmpdreset` 11

`\cmpdsetup` 10, 17 f., 21

Counter 19 f.

D

Delimiters 20 f.

E

etoolbox 3

expl3 2

F

Font style 18 f.

H

Harvard comma 27 f.

`hyper-link` 3, 17

`hyper-target` 3, 17

`hyper-use` 3, 16

`hyperref` 3, 16 ff.

I

`init-input-sep` 3, 11, 26

`init-strict` 3, 10

`init-sub` 3, 10

Initialize labels 9 ff.

L

`l3kernel` 2

`l3packages` 2

Language support 27 f.

`list-delim` 20 f.

`list-input-sep` 11, 24

`list-lang` 28

`list-lang` 25, 29

`list-last-sep` 25 f.

`list-output-sep` 25 f.

`list-prefix` 21

`list-serial-comma` 28

`list-suffix` 21

Lists 7

M

Main command 5 f.

O

Own labels 19

Own markers in images 19

P

Package options 4

`psfrag` 3, 14

R

`ref-style` 3

Reset the counter 11

S

Schemes by external programs 13–16

Separators

`\cmpdinit` 26

labels

input 22

output 23

list

last separator 25

lists

input 24

output 25

sublabels

input 23

output 24

Separators 22–27

`strict` 3

`sub-counter` 19

`sub-init` 3

`sub-input-sep` 3, 11, 23 f.

`sub-list` 3

`sub-marker` 3

`sub-only` 8

`sub-output-sep` 3, 23, 25

`sub-range-marker` 3

`sub-range-sep` 3

Sublabel

lists 7

ranges 7

Sublabel 6

`sublist-input-sep` 3, 11

`sublist-output-sep` 3

`subrange-input-sep` 3, 23

`subrange-output-sep` 3, 24

Suffix and prefix 21 f.

T

`textgreek` 3 f.

`translator` 27

U

Usage with hyperref 16 f.

X

`xparse` 2