

CHEMNUM

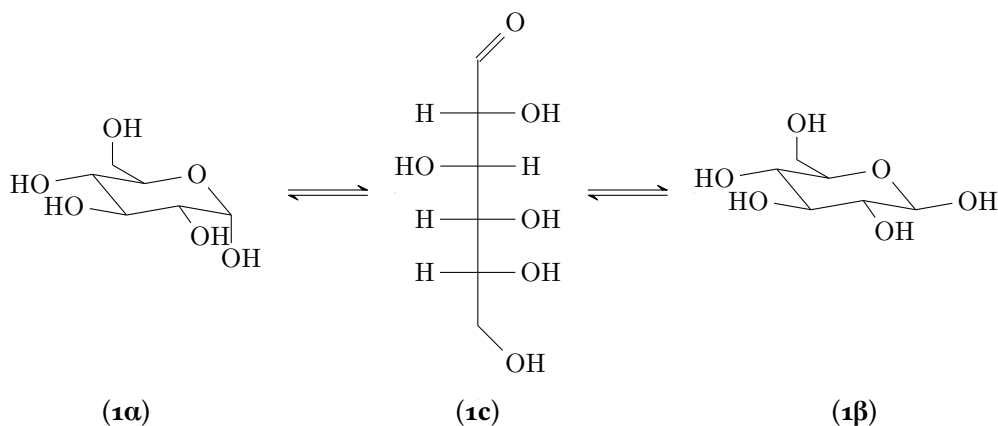
vo.6b 2013/06/05

a comprehensive approach for the numbering of chemical compounds

Clemens NIEDERBERGER

<https://bitbucket.org/cgnieder/chemnum/>
contact@mychemistry.eu

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

	5	Numbering compounds	4
	5.1	Main command	4
	5.2	Sublabel	5
	5.3	Lists	6
	5.4	Lists and ranges of sublabels . .	6
1	2	Initialize labels at once	8
	2.1	Version 0.4	2
	2.2	Version 0.5	3
2	3	Usage	8
	6.1.1	Main labels	8
	6.1.2	Sublabels	9
3	3	Error instead of warning	9
	6.2		
4	3	Customized input	10
	6.3		

6.4	Why initiate labels?	10	10.8	Separators	21
			10.8.1	Labels – input	21
7	Resetting the counter	10	10.8.2	Labels – output	21
8	Special usage	11	10.8.3	Sublabels – input	21
8.1	Moving arguments – section titles	11	10.8.4	Sublabels – output	22
8.2	Floating environments	11	10.8.5	Lists – input	22
8.3	Schemes created by external programs	12	10.8.6	Lists – output	23
			10.8.7	Lists – last separator	23
			10.8.8	<code>\cmpdinit</code> – input	24
			10.8.9	An example	24
9	Usage with hyperref¹	15	11	Language support	25
9.1	Basics	15	11.1	Cooperation with babel	25
9.2	Selected usage	15	11.2	Harvard comma	25
10	Customization	16	12	Overview	26
10.1	The principle	16	12.1	Options	26
10.2	Font style	17	12.2	Commands	27
10.3	Own labels	17			
10.4	Markers in images	17	Credits		28
10.5	Counter	18	Index		28
10.6	Delimiters	18			
10.7	Suffix and prefix	19			

1 A very short readme

CHEMNUM vo.6b 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 `xparse`.² `xparse` is part of the `l3packages`³ bundle, `expl3` is part of the `l3kernel`⁴ bundle. This means, **CHEMNUM** needs L^AT_EX3 Support.

CHEMNUM also needs the packages `etoolbox`,⁵ `psfrag`⁶ and either `textgreek`⁷ or `bm`,⁸ see section 3.

2 News and a word of apology

2.1 Version 0.4

In the update to vo.4 there have been some changes made with the options. 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 developped. Once it reaches v1.0, such changes won't be made any more.⁹

These options have new names:

¹ CTAN: hyperref ² CTAN: xparse ³ CTAN: l3packages ⁴ CTAN: l3kernel ⁵ CTAN: etoolbox ⁶ CTAN: psfrag

⁷ CTAN: textgreek ⁸ CTAN: bm ⁹ Or only in absolutely necessary cases.

old name	new name
<code>cmpd-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 23.
<code>sublist-input-sep</code>	list separation in the input of labels like 1a,b , see page 22.
<code>init-input-sep</code>	input in <code>\cmpdinit</code> , see page 24.
<code>hyper-use</code>	usage of hyperlinks, see page 15.
<code>hyper-target</code>	
<code>hyper-link</code>	
<code>ref-style</code>	Output of labels with <code>\cmpdref</code> , see pages 12 and 17.

Also new is the way how lists and ranges of sublabels are input. The option `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`, see section 9.

2.2 Version 0.5

With v0.5 there is a new option `ref-pos` which allows to change the positioning options of the `\psfrag` command that is used by the `\cmpdref` command internally, see p. 17.

The optional arguments of `\cmpdref` have changed their order, see p. 12!

3 Package options

`CHEMNUM` has one package option.

`textgreek` = `artemisia|cbgreek|euler|false` Default: false
 Choose if and with which option textgreek is used.

This document was set with `\usepackage[textgreek=artemisia]{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 – chemcono¹⁰ – 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 – chemcompounds¹¹ – 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.
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,¹² 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>}`

¹⁰ CTAN: chemcono ¹¹ CTAN: chemcompounds ¹² CTAN: bpchem

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:
2 \cmpd{a}. No pre-declaring is necessary. Compounds like \cmpd{c} are
3 numbered in the order they appear in the text.\par
4 Once again: \cmpd{b}, \cmpd{a}, \cmpd{c}.

Compounds 1 and 2 are declared and can be used any time: 1. No pre-declaring is necessary.
Compounds like 3 are numbered in the order they appear in the text.
Once again: 2, 1, 3.
```

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 8ff).

```

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

The hidden version declares the label but doesn't print anything. The next 5 continues to count
with the next number. With 4 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_}
2 _all_have_the_same_label._As_well_as_\cmpd{aa_},_\cmpd{a_a},_\cmpd{_aa}_and
3 _\cmpd{aa}_.

6, 6, 6, 6, 6 and 6 all have the same label. As well as 6, 6, 6 and 6.
```

5.2 Sublabel

If you want a label like **1a**, 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 \compd{f.one} and \compd{f.two} are related, as are \compd{g.one} and
2 \compd{g.two}. Of course, these labels can be used again: \compd{g.two} and
3 \compd{f.one}.

```

7a and **7b** are related, as are **8a** and **8b**. Of course, these labels can be used again: **8b** and **7a**.

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

```

1 \compd{a} and its variants \compd{a.one} and \compd{a.two}
1 and its variants 1a and 1b

```

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

```

1 \compd{f} and \compd{g}
7 and 8

```

How you can create a label like **7a,b** is explained in section 5.4.

5.3 Lists

There is actually more to the `\compd` command. It also prints lists of labels. The right description would be something like:

`\compd{<(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 22. Section 10.8.9 shows an example how a changed input could look like.

```

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

More than one label can be put inside \compd, separated by commas. Then a list like 1, 2, 3, 5, and 8b
is printed.

```

The Harvard comma (see 11.2) in **1, 2, 3, 5, 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: \compd{q.one, q.two, q.three, q.four, q.five}\\
2 label with list of sublabels: \compd{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>\compd{q.{two..four}}\l</code>	9b–d
2	<code>\compd{q.{one,three..five}}\l</code>	9a,c–e
3	<code>\compd{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 an option (see section 10 for more details), with which all sublabels of a main label can be printed as range.

`compd-all = true|false`

Default: false

Show all sublabels.

1	<code>\compd[compd-all]{q}</code>	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.comp` 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 option:

`sub-only = true|false`

Default: false

Print only the sublabel.

```
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 & & & & \ch{-R} & & Name \\\midrule
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}
```


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.
*****
```

If you need to use own labels (see page 17) 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 ...>}`

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 options

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

or use `\cmpdsetup`, see section 10.

6.3 Customized input

When you're using other tokens for `sub-input-sep` (page 21) or `sublist-input-sep` (page 22), you have to use them with `\cmpdinit` as well.

A change of `list-input-sep` (page 22) 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 24.

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 `\cmpd*` (section 5.1) and `\cmpd+` (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: \cmpd{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: \cmpd{a}, \cmpd{h}
```

Same label: **1**, **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 8) 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 8 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)}}
```

8.2 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[<tag>][<options>]{<label name>}`

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

This command is inspired by the `\schemeref` command from the package `chemscheme`.¹⁴ It works in a very similar way. You create the scheme and save it with temporary labels as an eps file.

Like `\cmpd+` the plussed variant only references a label but does not declare it.

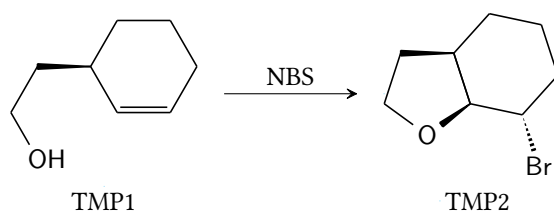
```

1 \begin{scheme}[ht]
2   \centering
3   \includegraphics{scheme-tmp.ps}
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
10  % \cmpdref{third} would replace TMP3
11  \includegraphics{scheme-tmp.ps}
12  \caption{Scheme with automated labels.}
13 \end{scheme}

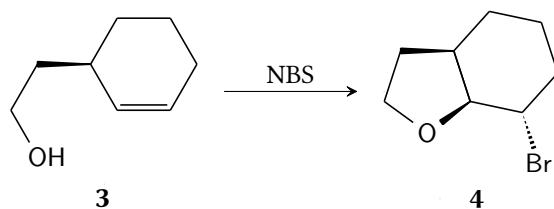
```

So you numerate the compounds with TMP1, TMP2 etc. These markers will then be replaced with the corresponding labels. Like `chemscheme` CHEMNUM uses the `\psfrag` command of the

¹⁴ CTAN: chemscheme



Scheme 3: Scheme with temporary markers.



Scheme 4: Scheme with automated labels.

psfrag package to do that. This means you have to either compile via L^AT_EX, DVIPS, PS2PDF, or for instance use the package auto-pst-pdf¹⁵ to be able to use pdfL^AT_EX. It doesn't work with X_YL^AT_EX, neither probably with LuaL^AT_EX, the author didn't test that, though.

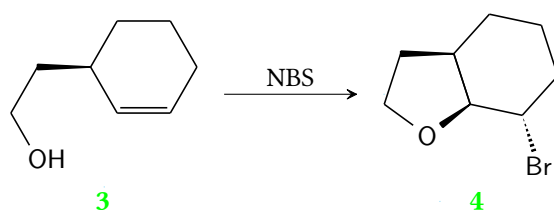
In each case it is important that the marker is saved as *text* in the eps file.

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

```

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

```



Scheme 5: Scheme with automated labels.

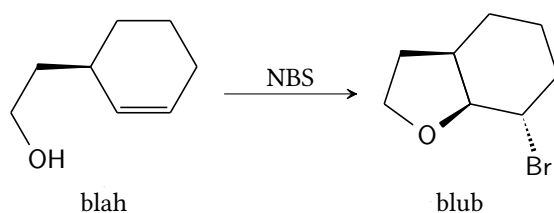
You also can replace arbitrary text by using the second optional argument.

¹⁵ CTAN: auto-pst-pdf

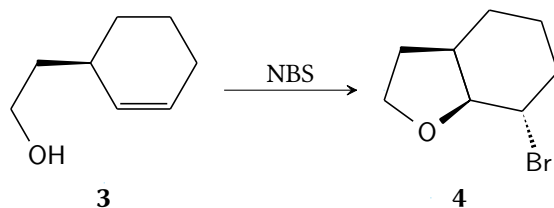
```

1 \begin{scheme}[ht]
2 \centering
3 \includegraphics{scheme-bla.ps}
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.ps}
12 \caption{Scheme with explicitly placed labels.}
13 \end{scheme}

```



Scheme 6: Scheme with arbitrary markers.



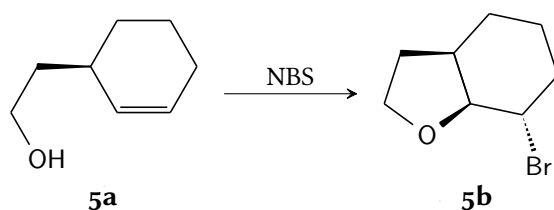
Scheme 7: Scheme with explicitly placed labels.

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.ps}
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.

`hyper-use` = `true`|`false`

Default: `false`

Link labels with `hyperref`'s hyperlink commands.

Setting this option (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
3 created: \cmpd{a.two,d,e}.
4
5 This doesn't work with labels containing a list of sublabels:
6 \cmpd{a,f.{one,two},c}. Since \cmpd{a.two} and \cmpd{a} are different
7 labels
7 they have different link targets.
```

The first time targets are created: **1b**, **2**, **3**, **4**, **5**, and **7**. Then links are created: **1b**, **4**, and **5**. This doesn't work with labels containing a list of sublabels: **1**, **7a,b**, and **3**. Since **1b** and **1** 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 options:

`hyper-target` = <link name>

(initially empty)

Set a hyperlink target.

`hyper-link` = <link name>

(initially empty)

Set a hyperlink.

With `hyper-target` you can set the target. **Caution:** if you use this option 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 option 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
2 corresponding target: \cmpd[hyper-target=mylink,cmpd-all]{f}.
```

Create a link **7a-4** but don't forget the corresponding target: **7a-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>}
```

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**.

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

- **cmpd**: These options change the whole label or the main label.
- **sub**, **sublist**, **subref**: These options have an impact on sublabels.
- **list**: Options which change the output of lists.
- **ref**: Options which have an impact on `\cmpdref`.
- **hyper**: Options which have an impact on the usage with hyperref.
- **init**: Options which have an impact in the initialization of the labels.

10.2 Font style

With the keys

`cmpd-style` = <style> (initially empty)
formatting commands for the labels. The last command can demand an argument.

`cmpd-weight` = bold|normal Default: bold
setting labels in bold series. The setting for this is extra since greek labels internally have to be treated a little differently.

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}
```

1, 2, 7b, and 10
1, 2, 7b, and 10
1, 2, 7b, and 10
1, 2, 7b, and 10
1, 2, 7b, 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

`cmpd-label` = <label> (initially empty)
This key let's you choose the label associated with the label name.

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

Own label: XYZ; in a list *all new compounds* get this label! XYZ, 1, and XYZ

10.4 Markers in images

There are several options that affect labels in external images placed with `\cmpdref`.

`ref-marker` = <marker> Default: TMP
This changes the default marker that is to be replaced.

`ref-pos = {<tex-pos>}{<ps-pos>}`
explained below

Default: `bb`

`ref-style = <style>`
change the format of the labels in external schemes.

Default: `\sffamily`

With `ref-pos` you can change the default way how `\cmpdref` places the label. The command `\psfrag` that is used internally has two options for that. The first one determines the position of the \TeX -replacement, the second one how it is placed in the box from the marker that is replaced. For details please refer to the documentation of `psfrag`.

10.5 Counter

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

`cmpd-counter = arabic|alph|Alph|greek|Greek|roman|Roman|Symbol`
The counter format for the main labels.

Default: `arabic`

`sub-counter = arabic|alph|Alph|greek|Greek|roman|Roman|Symbol`
The counter format for sublabels.

Default: `alph`

Please note: a change of format only affects labels *which are not yet declared!*

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

1, 2, 7b, and 10

K1, K2, and L1

73 and 74

13 α and 13 β

10.6 Delimiters

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

`cmpd-delim = <odelim><cdelim>`

(initially empty)

Delimiters to the single labels, need to be *two* tokens!

`list-delim = <odelim><cdelim>`

(initially empty)

Delimiters to a list of labels, need to be *two* 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
```

1, 2, 7b, and 10
(1), (2), (7b), and (10)
1, 2, 7b, and 10
(1), (2), (7b), and (10)

Please be aware that the default values (see pages 16 and 26) of `compd-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 `\compd` contains at least two label names.

```

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

1, 2, 7b, and 10
[1, 2, 7b, and 10]
1, 2, 7b, and 10
[1, 2, 7b, and 10]
1

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

```

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

normally: **(2), (3), (4a)**, but sometimes so **5** or so **5**.

With

`\compd- {<label name>}`

`\compdref- {<label name>}`

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

```

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

1, 2, 7b, and 10
[(1), (2), (7b), and (10)]
[1, 2, 7b, and 10]

Using `\compdsetup{compd-delim = }` and `\compdsetup{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.

`cmpd-prefix` = <prefix> (initially empty)
Set a prefix for a label.

`cmpd-suffix` = <prefix> (initially empty)
Set a suffix for a label.

`list-prefix` = <prefix> (initially empty)
Set a prefix for a list.

`list-suffix` = <prefix> (initially empty)
Set a suffix for a list.

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!

1, 2, 7b, and 10
Nr. 1, Nr. 2, Nr. 7b, and Nr. 10
→ 1, → 2, → 7b, and → 10
→ 1
```

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!

1, 2, 7b, and 10
list: 1, 2, 7b, and 10
collection: 1, 2, 7b, and 10
1
```

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>

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

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 # 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>

Default: .

The token that separates sublabels from main labels.

1	<code>\cmpdsetup{sub-input-sep = !}</code>	7a and 8b
2	<code>\cmpd{f!one, g!two}\\</code>	7a and 8b
3	<code>\cmpd[sub-input-sep= +]{f+one, g+two}\\</code>	7a and 8b
4	<code>\cmpd[sub-input-sep= ~]{f~one, g~two}\\</code>	7a and 8b
5	<code>\cmpd[sub-input-sep= &]{f&one, g&two}\\</code>	7a and 8b
6	<code>\cmpd[sub-input-sep= *]{f*one, g*two}</code>	7a and 8b

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>

(initially empty)

the separator between main and sublabels in the output.

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

1, 2, 7b, 73, and 10
1, 2, 7-b, 7-3, and 10
1, 2, 7·b, 7·3, and 10
1, 2, 7:b, 7:3, 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 # and *you shouldn't* use @. Furthermore you should be careful to use different symbols for the other separators.

subrange-input-sep = <separator>

Default: . .

The tokens that indicate a range of sublabels

```
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:

subrange-output-sep = <separator>

Default: -

The tokens that are placed between the sublabels of a range in the output.

```
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 # and *you shouldn't* use @. Furthermore you should be careful to use different symbols for the other separators.

list-input-sep = <separator>

Default: ,

Separator between main labels in the input.

sub-input-sep = <separator>

Default: ,

Separator between sublabels in the input.

```
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}
```

1, 2, 7b, and 10
 1, 2, 7b, and 10
 1, 2, 7b, and 10
 1, 2, 7b, and 10

With this you can for example use IUPAC names as label names.¹⁶

```
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.

list-output-sep = <separator> Default: ,
 Separator between main labels in the output of a list.

sub-output-sep = <separator> Default: ,
 Separator between sublabels in the output of a sublist.

```
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}
```

1, 2, 7b, and 10
 1 2 7b and 10
 1, 2, 7b, and 10
 1; 2; 7b; and 10

10.8.7 Lists – last separator

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

list-last-sep = <separator> (initially empty)
 The separator between the last two labels in the output of a list.

list-lang = US|GB|DE|FR|ES|IT Default: US
 The language setting for a list.

```
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}
```

¹⁶ Thanks to Christina Lüdigg who in a way suggested this feature.

1, 2, 7b, and 10 1 and 2
 1, 2, 7b and 10 1 and 2
 1, 2, 7b und 10 1 und 2
 1, 2, 7b et 10 1 et 2
 1, 2, 7b y 10 1 y 2
 1, 2, 7b e 10 1 e 2

As you can see the language `list-lang = 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}
```

1, 2, 7b and also 10
 1, 2, 7b 10
 1, 2, 7b as well as 10
 1, 2, 7b, 10

`list-last-sep = 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}
```

1, 2, 7b, 10
 1; 2; 7b; 10
 1 und 2 und 7b 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. Tokens you *can't* use are %, , and # and you *shouldn't* use @.

`init-input-sep = <separator>`

Default: ,

The separator for the input of labels in `\cmpdinit`.

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}

```

1, 7a, and 9a,c–e

11 Language support

11.1 Cooperation with babel

CHEMNUM supports the language choice made with the babel¹⁷ package for a number of languages. This is done with the `\GetTranslation` command of the translations¹⁸ package.

```

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 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

¹⁷ CTAN: babel ¹⁸ translations is part of the exsheets bundle. ¹⁹ contatc@mychemistry.eu

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` Default: true
 (De-) Activate the serial comma.

```
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 Options

In this section all options are listed.

option	default	predefined	type
<code>cmpd-style</code>			macro
<code>cmpd-weight</code>	bold	bold	choice
<code>cmpd-label</code>			literal
<code>cmpd-delim^a</code>	()		literal
<code>cmpd-odelim</code>			literal
<code>cmpd-cdelim</code>			literal
<code>cmpd-prefix</code>			literal
<code>cmpd-suffix</code>			literal
<code>cmpd-space</code>	<code>\penalty \@m \</code>	<code>\penalty \@m \</code>	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
ref-pos ^a	bb	bb	literal
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 4ff.
<code>\cmpd*{<labels>}</code>	invisible, declares label, see page 5
<code>\cmpd- [<keyval>]{<labels>}</code>	without delimiters, see page 19
<code>\cmpd+- [<keyval>]{<labels>}</code>	label is read from auxiliary file, see page 11
<code>\cmpdinit[<keyval>]{<labels>}</code>	initialise and declare labels, see pages 8f.
<code>\cmpdinit* [<keyval>]{<labels>}</code>	initialise labels, see pages 8f.
<code>\cmpdref [<keyval>][[]<marker>]{<label>}</code>	replace temporary markers in eps files by labels, see page 12
<code>\cmpdref- [<keyval>][[]<marker>]{<label>}</code>	replace temporary markers in eps files by labels without delimiters, see page 19
<code>\cmpdreset [<num>]</code>	reset counter, see page 10
<code>\cmpdsetup{<keyval>}</code>	setup CHEMNUM , see pages 16ff.

Credits

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

Many thanks also to Jonas RIVETTI who provided the Italian translation.

Index

Section headings are typeset **bold**, packages sans serif, commands **brown**, and options **yellow**.

A	cmpd-style 17	exsheets 25
auto-pst-pdf 13	cmpd-suffix 20	
	cmpd-weight 17	F
B	\cmpd* 5, 10 f., 15	Font style 17
babel 25, 27	\cmpd+ 10 f.	
bm 2	\cmpd- 19	H
bpchem 4	\cmpdinit 3, 5–11, 21	Harvard comma 25 f.
	\cmpdinit* 8 ff.	hyper-link 3, 15 f.
C	\cmpdref 3, 12 ff.	hyper-target 3, 15 f.
chemcompounds 4, 8	\cmpdref+ 12	hyper-use 3, 15
chemcono 4	\cmpdref- 19	hyperref 2 f., 15 f.
chemscheme 12	\cmpdreset 10	
\cmpd 4 ff., 11	\cmpdsetup 10, 16	I
\cmpd+ 12	Counter 18	init-input-sep 3, 10
cmpd-all 7		init-strict 3, 9 f.
cmpd-counter 18	D	init-sub 3, 9 f.
cmpd-delim 18 f.	Delimiters 18 f.	Initialize labels 8 ff.
cmpd-label 17		
cmpd-prefix 20	E	L
cmpd-sep 3	etoolbox 2	l3kernel 2
cmpd-space 20	expl3 2	l3packages 2

Language support	25 f.
<code>list-delim</code>	18 f.
<code>list-input-sep</code>	10, 22
<code>list-lang</code>	26
<code>list-lang</code>	23 f., 27
<code>list-last-sep</code>	23 f.
<code>list-output-sep</code>	23 f.
<code>list-prefix</code>	20
<code>list-serial-comma</code>	26
<code>list-suffix</code>	20
Lists	6
M	
Main command	4 f.
Markers in images	17 f.
O	
Own labels	17
P	
Package options	3
<code>psfrag</code>	2, 13
<code>\psfrag</code>	3, 12
R	
<code>ref-marker</code>	17
<code>ref-pos</code>	3, 18

<code>ref-style</code>	3, 18
Reset the counter	10
S	
Schemes by external	
programs	12 ff.
Separators	
<code>\cmpdinit</code>	24
labels	
input	21
output	21
list	
last separator	23
lists	
input	22
output	23
sublabels	
input	21
output	22
Separators	21–25
Special usage	11–14
<code>strict</code>	3
<code>sub-counter</code>	18
<code>sub-init</code>	3
<code>sub-input-sep</code>	3, 10 f.
<code>sub-list</code>	3

<code>sub-marker</code>	3
<code>sub-only</code>	7
<code>sub-output-sep</code>	3, 21
<code>sub-range-marker</code>	3
<code>sub-range-sep</code>	3
Sublabel	
lists	6
ranges	6
Sublabel	5 f.
<code>sublist-input-sep</code>	3, 10
<code>sublist-output-sep</code>	3
<code>subrange-input-sep</code>	3, 22
<code>subrange-output-sep</code>	3, 22
Suffix and prefix	19 ff.
T	
<code>textgreek</code>	3
<code>textgreek</code>	2 f.
translations	25
U	
Usage with hyperref	
Usage with hyperref ...	15 f.
X	
<code>xparse</code>	2