CHEMNUM

v1.0 2013/12/04

CHEMNUM revisited

Clemens NIEDERBERGER

Table of Contents

1 License and Requirements

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

CHEMNUM requires the bundles l3kernel [The13a] and l3packages [The13b]. It also requires the translations package [Nie13b] and chemgreek from the chemmacros bundle [Nie13a].

2 News

The CHEMNUM package has been my first attempt to create a comprehensive labeling package for chemical compounds. However, it had and has more than one weakness and its code was – to be frank – a mess. Version 1 is now a complete re-write of CHEMNUM where I tried to achieve several points:

- A cleaner code internally.
- A cleaner user interface, *i. e.*, more user macros for different tasks, a unified naming of the commands and a less redundant naming of the options.
- Extended functionality such as sorting and compressing of sublabel lists.

Although the syntax is more or less the same some changes have been made that make version 1 incompatible with version 0. This is way Version 0 is still available through the option `version = {0}`.

Many commands have got a new name! The most important ones are:

- `\cmpdref`; this is now called `\replacecmpd`.
- `\cmpdinit`; this is now called `\initcmpd`.
- `\cmpdsetup`; this is now called `\setchemnum`.

3 Background

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 [Sch99] – 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 [Sch06] – 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 [Pedo4], 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.

4 Overview over the Available Commands

`\cmpd*+[`⟨*options*⟩`]{`⟨*list of IDs*⟩`}`

The main command for creating and refering to compound labels. This command is described in detail in section 6.

`\refcmpd[`⟨*options*⟩`]{`⟨*ID*⟩`}`

This command only refers to a already defined label but does not define a label itself. This is an alias of `\cmpd+`.

`\labelcmpd[`⟨*options*⟩`]{`⟨*ID*⟩`}`

This command only defines a new label but does not print it. This is an alias of `\cmpd*`.

* `\cmpdplain{`⟨*ID*⟩`}`

Reads a label and writes it expandably without formatting. It is not able to parse a list. Its sole purpose is usage in pdfstrings (*cf.* `\texorpdfstring{`⟨*TEX*⟩`}{`⟨*pdfstring*⟩`}`). This command is described in section 6.

* `\subcmpdplain{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}`

Reads a sublabel and writes it expandably without formatting. It is not able to parse a list. Its sole purpose is usage in pdfstrings (*cf.* `\texorpdfstring{`⟨*TEX*⟩`}{`⟨*pdfstring*⟩`}`). This command is described in section 6.

* `\submaincmpdplain{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}`

Reads a main and a sublabel and writes them expandably without formatting. It is not able to parse a list. Its sole purpose is usage in pdfstrings (*cf.* `\texorpdfstring{`⟨*TEX*⟩`}{`⟨*pdfstring*⟩`}`). This command is described in section 6.

`\initcmpd[`⟨*options*⟩`]{`⟨*list of IDs*⟩`}`
Initiate compound labels. This command can only be used in the preamble. It is desribed in section 6.

* `\cmpdproperty{`⟨*ID*⟩`}{`⟨*property*⟩`}`
Get the associated property ⟨*property*⟩ of compound ⟨*ID*⟩. This command is described in detail in section 6.

* `\subcmpdproperty{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}{`⟨*property*⟩`}`
Get the associated property ⟨*property*⟩ of subcompound ⟨*sub ID*⟩ of compound ⟨*main ID*⟩. This command is described in detail in section 6.

`\newcmpdlabelformat{`⟨*name*⟩`}{`⟨*command*⟩`}`
Makes the label format ⟨*name*⟩ known to CHEMNUM. ⟨*command*⟩ needs to be a command that takes an integer number as argument and should return a formatted version of it. In practice you should not need to use this command as the most common formats already are defined. This command is described in section 8.

`\resetcmpd[`⟨*integer*⟩`]` Default: `1`
Reset the numbering for main compound labels to start with ⟨*integer*⟩ again. This is the same as `\setcounter{cmpdmain}{`⟨*integer*⟩ − 1`}`. The command is described in section 8.

`\cmpdshowdef{`⟨*ID*⟩`}`
Internal command used to display ⟨*ID*⟩ of a newly defined compound label when the option `show-keys` is used. The command is described in section 12.

`\cmpdshowref{`⟨*ID*⟩`}`
Internal command used to display ⟨*ID*⟩ of a referencing compound label when the option `show-keys` is used. The command is described in section 12.

`\subcmpdshowdef{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}`
Internal command used to display ⟨*main ID*⟩ and ⟨*sub ID*⟩ of a newly defined subcompound label when the option `show-keys` is used. The command is described in section 12.

`\subcmpdshowref{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}`
Internal command used to display ⟨*main ID*⟩ and ⟨*sub ID*⟩ of a referencing subcompound label when the option `show-keys` is used. The command is described in section 12.

5 Numbering Compounds

5.1 Main command

The main command of this package is this one:

`\cmpd{`⟨*ID*⟩`}`
When ⟨*compound name*⟩ is used the first time, the label is created, saved (= declared) and printed. Each further use just prints the label.

```
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
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 it is possible with

`\cmpd*{`⟨*ID*⟩`}`

Declare the label but don't print anything.

```
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 **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 section 10). Be careful with blanks though! Leading and trailing spaces are ignored, spaces at other places are not. It's probably best not to use blanks in label names at all.

```
\cmpd{aa}, \cmpd{aa }, \cmpd{ aa}, and \cmpd{ aa } all have the same label.
Likewise \cmpd{a a}, \cmpd{a a }, \cmpd{ a a}, \cmpd{ a a }, \cmpd{a  a},
\cmpd{a  a }, \cmpd{ a  a}, and \cmpd{ a  a }.
```

6, **6**, **6**, and **6** all have the same label. Likewise **7**, **7**, **7**, **7**, **7**, **7**, **7**, and **7**.

5.2 Sublabel

If you want a label like **1a**, you need to use the following syntax:

`\cmpd{`⟨*main ID*⟩`.`⟨*sub ID*⟩`}`

⟨*main ID*⟩ is the main name which stays the same, ⟨*sub ID*⟩ varies. This syntax means that the point `.` *cannot* be a part of ⟨*main ID*⟩ or ⟨*sub ID*⟩. Instead of the point you also can use another symbol, see section 10.

```
\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}.
```

8a and **8b** are related, as are **9a** and **9b**. Of course these labels can be used again: **9b** and **8a**.

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

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

---

1 and its variants **1a** and **1b**

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

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

8 and **9**

How you can create a label like **8a**,**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)*⟩`}`
Treats each entry of the list as described before.

This means that with default settings the comma can't be part of the label name unless hidden in braces. As separator can be used another symbol, too, see section 10.

```
More than one label can be put inside \cs{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 **1**, **2**, **3**, **5**, and **9b** is printed.

The Harvard comma (see section 11) in `, and` between **5** and **9b** suggests that there are options to customize the list, see section 9.

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 **10a**, **10b**, **10c**, **10d**, and **10e** which for example differ in their substituents. It can be useful to refer to them all at once:

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

```
\setchemnum{compress=false}%
list of labels: \cmpd{q.one, q.two, q.three, q.four, q.five}\par
label with list of sublabels: \cmpd{q.{one,two,three,four,five}}
```

---

list of labels: **10a**, **10b**, **10c**, **10d**, and **10e**
label with list of sublabels: **10a**,**b**,**c**,**d**,**e**

Since the sublist is input with a comma in the default setting, you have to put them into braces. If you add a list of sublabels to a main label they will always be printed in the order the sublabels have been declared and not in the order they're input in the list:

```
\setchemnum{compress=false}%
compare \cmpd{q.{one,two,three,four,five}}
with \cmpd{q.{five,four,three,two,one}} and
\cmpd{q.{three,four,one,five,two}}
```

---

compare **10a**,**b**,**c**,**d**,**e** with **10a**,**b**,**c**,**d**,**e** and **10a**,**b**,**c**,**d**,**e**

Using this syntax you also can create ranges of sublabels. For this you enable the option `compress`. Or rather: this is the default setting.

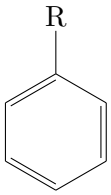
```
\cmpd{q.{two,four,three}} \par
\cmpd{q.{five,one,three,four}} \par
\cmpd{q.{one,three,five,two}}
```

10b–d
10a,**c–e**
10a–c,**e**

Obviously you can't use a comma as part of a sublabel name. You can change the input marker, though. See section 7 for available options.

```
% uses packages 'chemfig', 'chemformula' and 'booktabs'
\setatomsep{2em}%
\chemname{\chemfig{*6(=-=-(-R)=-)}}{\cmpd{benzene.{H,Me,OH,NH2}}}
\begin{tabular}{lll}
                                & \ch{-R}   & Name \\\midrule
  \cmpd[sub-only]{benzene.H}    & \ch{-H}   & Benzene \\
  \cmpd[sub-only]{benzene.Me}   & \ch{-CH3} & Toluene \\
```

```
  \cmpd[sub-only]{benzene.OH}  & \ch{-OH}  & Phenol \\
  \cmpd[sub-only]{benzene.NH2} & \ch{-NH2} & Phenylamine (Aniline)
\end{tabular}
```



11a–d

	−R	Name
a	−H	Benzene
b	$-CH_3$	Toluene
c	−OH	Phenol
d	$-NH_2$	Phenylamine (Aniline)

6 Details on Compound Labels

6.1 How things work

When you call `\cmpd` with a new label three things happen:

- The new label gets initiated. This is nothing more than adding it to an internal list. The purpose of this is explained in section 6.3.
- The new label gets declared. This means that a number of internal commands are defined. Amongst other things they hold a number of properties associated with the corresponding label. Those properties are explained in more detail in section 6.2. The necessary information of the label are also written to the `aux` file.
- The label gets printed.

Since new labels are declared when `\cmpd` is first used using it in section titles that are written to the table of contents may to lead to wrong numbering. In order to avoid this compound label information is written to the `aux` file. The command `\refcmpd[`⟨*options*⟩`]{`⟨*ID*⟩`}` only reads those information but does not declare a label. There is also a command which does the opposite: it declares a label if it hasn't been declared before but will not print the corresponding label: `\labelcmpd[`⟨*options*⟩`]{`⟨*ID*⟩`}`.

Both commands have shortcut versions: `\cmpd+` is the same as `\refcmpd`, `\cmpd*` is the same as `\labelcmpd`.

Another command available is `\cmpdplain{`⟨*ID*⟩`}`. This command is similar to `\refcmpd`. There are a few important differences, though: `\cmpdplain` does *not* take a list of labels as argument. It also is *not* able to interpret sublabels. `\cmpdplain` does not format the label with whatever format has been declared. And last but not least: it is expandable. This means it can be used to get labels in PDF bookmarks. It's equivalent `\subcmpdplain{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}` does the same for sublabels. A third sibling, `\submaincmpdplain{`⟨*main ID*⟩`}{`⟨*sub ID*⟩`}`, writes both the main and the sublabel.

6.2 Properties of compound labels

Every label has a number of properties. The first property is of course its ID which identifies the label. The other properties are:

number An internal unique number.

counter-representation The counter representation associated with the label. This is the actual label that get's printed.

pre-label-code Code to be inserted before the label is printed.

post-label-code Code to be inserted after the label is printed.

label-format Formatting commands for the label. This is most likely something like `\bfseries`.

* `\cmpdproperty{⟨ID⟩}{⟨property⟩}`
Get the associated property ⟨*property*⟩ of compound ⟨ID⟩. This command is expandable.

```
\newcommand*\expandfull{\romannumeral-‘0}%
\newcommand*\expandtwice{\detokenize\expandafter\expandafter\expandafter}%
\ttfamily
number: \cmpdproperty{benzene}{number}\par
counter-representation: \cmpdproperty{benzene}{counter-representation}\par
pre-label-code: \cmpdproperty{benzene}{pre-label-code}\par % empty
post-label-code: \cmpdproperty{benzene}{post-label-code}\par % empty
label-format: \expandtwice{\expandfull\cmpdproperty{benzene}{label-format}}
```

```
number:  11
counter-representation:  11
pre-label-code:
post-label-code:
label-format:  \protect \bfseries
```

Similarly a sublabel has associated properties. Additionally to the obvious ones – its ID and the ID the main label it belongs to – these are

number An internal unique number.

counter-representation The counter representation associated with the label. This is the actual label that get's printed.

* `\subcmpdproperty{⟨main ID⟩}{⟨sub ID⟩}{⟨property⟩}`
Get the associated property ⟨*property*⟩ of subcompound ⟨*sub* ID⟩ of compound ⟨*main* ID⟩. This command is expandable.

```
\ttfamily
main-compound: \subcmpdproperty{benzene}{OH}{main-compound}\par
number: \subcmpdproperty{benzene}{OH}{number}\par
counter-representation: \subcmpdproperty{benzene}{OH}{counter-representation}
```

---

main-compound: benzene
number: 3
counter-representation: c

6.3 Initiating labels

Initiating labels is not the same as declaring them. In fact this may only be useful in order to keep track of defined labels. If you set the option

7 Overview over the Available Options

Except for the `version` option all of the following options are either set as options to `\cmpd` or `\initcmpd` directly or via `\setchemnum{`⟨*options*⟩`}`, each time as a comma separated list of key/value pairs. Options that can be set via `\setchemnum` are marked with `global`, those that only have an effect when used with `\cmpd` and friends are marked with `compound`. Those marked with `all` can be set either way.

One of the options only has an effect when used with the `\replacecmpd` command. It is marked with `replace`.

`version` = `0`|`1` Default: `1`

Choose the package version. This option can only be set as a package option!

`global` » `counter-within` = {⟨*counter*⟩}

Reset the compound numbers when ⟨*counter*⟩ is stepped.

`all` » `counter-format` = `arabic`|`alph`|`Alph`|`roman`|`Roman`|`greek`|`Greek` Default: `arabic`

The format of the number associated with the main compounds.

`all` » `sub-counter-format` = `arabic`|`alph`|`Alph`|`roman`|`Roman`|`greek`|`Greek` Default: `alph`

The format of the number associated with the sub compounds.

`all` » `compress` = `true`|`false` Default: `false`

If set to true a list of sublabels is compressed, *i. e.*, **1oa**,**c**,**d**,**e** becomes **1oa**,**c**–**e**.

`compound` » `pre-label-code` = {⟨*code*⟩} (initially empty)

Code to be inserted before a label.

`compound` » `post-label-code` = {⟨*code*⟩} (initially empty)

Code to be inserted after a label.

`all` » `main-sub-sep` = {⟨*code*⟩} Default: `.`
The separator symbol that is used in `\cmpd` to separate the ⟨*main ID*⟩ from a ⟨*sub ID*⟩.

`all` » `list-label-sep` = {⟨*code*⟩} Default: `,`
The separator that is used to separate different ⟨*main ID*⟩s in `\cmpd`.

`all` » `format` = {⟨*formatting commands*⟩} Default: `\bfseries`
The default format of the labels.

`all` » `list-sep-two` = {⟨*code*⟩} Default: `␣and␣`
The output separator between labels in a list that contains of two items.

`all` » `list-sep-more` = {⟨*code*⟩} Default: `,␣`
The output separator between labels in a list that contains of more than two items.

`all` » `list-sep-last-two` = {⟨*code*⟩} Default: `,␣and␣`
The output separator between the last two labels in a list that contains of more than two items.

`compound` » `sub-only` = `true`|`false` Default: `false`
If true the command `\cmpd` will only print sublabels but no main labels.

`compound` » `sub-all` = `true`|`false` Default: `false`
If true the command `\cmpd` will print all sublabels belonging to the corresponding main label.

`all` » `sub-list-sep-two` = {⟨*code*⟩} Default: `,`
The output separator between labels in a sublist that contains of two items.

`all` » `sub-list-sep-more` = {⟨*code*⟩} Default: `,`
The output separator between labels in a sublist that contains of more than two items.

`all` » `sub-list-sep-last-two` = {⟨*code*⟩} Default: `,`
The output separator between the last two labels in a sublist that contains of more than two items.

`all` » `sub-list-sep-range` = {⟨*code*⟩} Default: `--`
The output separator between two labels in a sublist denoting a range. This is only used when the option `compress` is active.

`global` » `replace-auto` = `true`|`false` Default: `true`
When set to true this adds an incremented integer to the replacement tag.

`global` » `replace-tag` = {⟨*text*⟩} Default: `TMP`
The default replacement tag.

`replace` » `tag` = {⟨*text*⟩} Default: `TMP`
The local replacement tag.

`global` » `replace-style` = {⟨*code*⟩} Default: `\sffamily`
Additional TEX code that it placed before the `\cmpd` command in the replacement.

`replace` » `style` = {⟨*code*⟩} Default: `\sffamily`
Local additional TEX code that it placed before the `\cmpd` command in the replacement.

`global` » `replace-pos` = ⟨{⟨*TEX pos*⟩}{⟨*PS pos*⟩}⟩ Default: `bb`
Options for psfrag's `\psfrag`.

`replace` » `pos` = ⟨{⟨*TEX pos*⟩}{⟨*PS pos*⟩}⟩ Default: `bb`
Local options for psfrag's `\psfrag`.

`global` » `init` = `true`|`main`|`false`|`strict`|`main-strict` Default: `false`
Determines how labels have to be initiated. `false` means that labels are initiated when they're used the first time in the text. `true` means that labels should be initiated in the preamble with `\initcmpd`. `main` is the same as `true` but only for main labels. `strict` means that if an un-initiated label is used an error is thrown. `main-strict` is the same as `strict` but only for main labels.

`global` » `log` = `true`|`false`|`silent`|`verbose` Default: `false`
Determines how the declaration of the labels will be logged. `false` means that no information is written to the `.log` file. `true` means that basic information is written to the `.log` file when a label or a sublabel is declared. `silent` is an alias of `true`. `verbose` means that detailed information is written to the `.log` file when a label or a sublabel is declared.

8 The Counter Settings

9 Formatting Labels

10 Changing the Input Markers

11 Language Dependent Settings

12 Debugging Information

References

[Nie13a] Clemens Niederberger. chemmacros. version 4.1c, Nov. 20, 2013.
URL: http://mirror.ctan.org/macros/latex/contrib/chemmacros/.

[Nie13b] Clemens Niederberger. translations. version 1.1a, Sept. 30, 2013.
URL: http://mirror.ctan.org/macros/latex/contrib/translations/.

[Ped04] Bjørn Pedersen. bpchem. version 1.06, Nov. 25, 2004.
URL: http://mirror.ctan.org/macros/latex/contrib/bpchem/.

[Sch06] Stefan Schenk. chemcompounds. version 1.1.3, Dec. 1, 2006.
URL: http://mirror.ctan.org/macros/latex/contrib/chemcompounds/.

[Sch99] Stefan Schulz. chemcono. version 1.3, Sept. 9, 1999.
URL: http://mirror.ctan.org/macros/latex/contrib/chemcono/.

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

[The13b] The LaTeX3 Project Team. l3packages. version SVN 4582, July 28, 2013.
URL: http://mirror.ctan.org/macros/latex/contrib/l3packages/.

Index