

# The chemstyle bundle — Schemes and style for chemistry\*

Joseph Wright<sup>†</sup>

Released 2010/03/29

## Abstract

The chemstyle bundle provides two packages: chemstyle and chemscheme. Both are intended to help chemists create floating graphics and match published styles.

The chemscheme package consists of two parts, both related to chemical schemes. The package adds a scheme float type to the L<sup>A</sup>T<sub>E</sub>X default types figure and table. The scheme float type acts in the same way as those defined by the L<sup>A</sup>T<sub>E</sub>X kernel, but is intended for chemical schemes. The package also provides a method for adding automatic chemical numbering to schemes.

The chemstyle package provides a “one-stop shop” for setting up formatting of L<sup>A</sup>T<sub>E</sub>X documents following the editorial policies of various chemical journals. It provides a number of handy chemistry-related commands, and loads several support packages to aid the chemist.

## Contents

		6.2 Reference numbers in graphics . . . . .	7
<b>1 Introduction</b>	<b>2</b>	<b>7 Horizontal positioning of all floats</b>	<b>10</b>
<b>2 Installation</b>	<b>2</b>	<b>8 Some thoughts on generating chemical schemes</b>	<b>11</b>
<b>3 Requirements</b>	<b>2</b>	8.1 Overview . . . . .	11
<b>4 The chemstyle package: tools for chemistry</b>	<b>3</b>	8.2 Macro-based methods . .	11
		8.3 Graphical methods . . .	12
<b>5 Naming of the references section</b>	<b>3</b>	<b>9 Additional information</b>	<b>12</b>
5.1 Additional units . . . . .	4	9.1 Interactions with other packages . . . . .	12
5.2 The standard state symbol	4	9.2 Captions above floats . .	12
5.3 Latin phrases . . . . .	4	<b>10 A demonstration file</b>	<b>13</b>
5.4 Alkyl radicals . . . . .	5	<b>11 Change History</b>	<b>13</b>
<b>6 The chemscheme package: graphics for chemistry</b>	<b>6</b>	<b>12 Index</b>	<b>14</b>
6.1 Floating schemes . . . . .	6		

\*This file describes version v2.0g, last revised 2010/03/29.

<sup>†</sup>E-mail: joseph.wright@morningstar2.co.uk

## 1 Introduction

The chemstyle bundle consists of two parts. The first, the chemstyle package itself, aims to provide an easy and reliable method to set up various document parameters (such as caption formatting), simply by specifying the model journal. The package has also been designed to allow rapid addition of new journal styles. Each style definition is a separate file, and new styles can be added very readily. The formatting system provided by chemstyle are intended for writing a variety of documents. Thus, the stylistic changes made by the package do not seek to reproduce the appearance of printed journal articles. The package aims to be suitable for use in preparing drafts of papers, but also for writing reports, theses and so on.

The second part of the bundle is the chemscheme package, which is loaded automatically by chemstyle. chemscheme undertakes two tasks. First, it provides a floating scheme environment, which acts in the same way as the kernel figure and table floats. Secondly, it interfaces with either the chemcompounds or bpchem packages to allow automatic numbering of chemical compounds inside graphics.

This manual is written to cover both packages. All of the material is relevant to users of the chemstyle package. For those only using chemscheme,

## 2 Installation

The entire bundle is supplied with the TDS-ready ZIP file, `chemstyle.tds.zip`. Simply unzip this into your local texmf tree and run your hash program: `texhash` should work with recent versions of either  $\text{\TeX}$  Live or  $\text{MiK}\text{\TeX}$ .

To extract the bundle of files from `chemstyle.dtx`, run `(pdf)\TeX` on `chemstyle.dtx`. This will produce all of the package files, and also `README.txt`. To extract the files and build the documentation, run `(pdf)\L\TeX` on `chemstyle.dtx`: you will need to enable “write18” if you compile in PDF mode.

## 3 Requirements

The chemstyle class requires the following packages:

- `amstext` (part of the AMS bundle);
- `caption`;
- either `float` or `floatrow`;
- `kvoptions` (part of the oberdiek bundle);
- either `siunitx` or `Slunits` (otherwise no units are defined);
- `psfrag`;
- `varioref` (part of the standard tools bundle): only loaded if not turned off;
- `xspace` (part of the standard tools bundle).

These are normally present in the current major TeX distributions, but are also available from [The Comprehensive TeX Archive Network](#).

`varioref` The option `varioref` can be used to determine whether `chemstyle` loads the `varioref` package. For example, users of `cleveref` will want to turn off loading of `varioref` using:

```
\usepackage[varioref=false]{chemstyle}
```

## 4 The chemstyle package: tools for chemistry

The package recognises a number of key-value options when loading. Some of these are also available in the document body, and are described along with the relevant commands. Options which apply to `chemscheme` can be given when loading `chemstyle` and will be applied correctly.

`journal` `chemstyle` is supplied with a number of configuration files, based on the styles adopted by a number of chemistry journals ([Table 1](#)). A style can be chosen by loading the package with the `journal=<style>` key.

Table 1: Styles provided by `chemstyle`

Option	Journals using this style
<code>angew</code>	<i>Angew. Chem., Chem. Eur. J.</i>
<code>jomc</code>	<i>J. Organomet. Chem., Coord. Chem. Rev.</i>
<code>ic</code>	<i>Inorg. Chem.</i>
<code>jacs</code>	<i>J. Am. Chem. Soc.</i>
<code>jcp</code>	<i>J. Phys. Chem. A, J. Phys. Chem. B</i>
<code>orglett</code>	<i>Org. Lett.</i>
<code>rsc</code>	<i>Chem. Commun., Org. Biomol. Chem.</i> <i>Dalton Trans.</i>
<code>tetlett</code>	<i>Tetrahedron, Tetrahedron Lett.</i>

`\cstsetup` Many of the package options can be altered anywhere in the document, using the `\cstsetup` macro. This accepts a keyval list and processes it as needed.

## 5 Naming of the references section

`chemstyle` alters the naming of the references section of a document. By default, `chemstyle` alters the value of `\bibname` or `\refname` (as appropriate) to the form of words chosen by the target journal for the “References” section.

The “References” naming commands are language-aware, *via* the `babel` interface. Currently, `chemstyle` includes appropriate labels for `babel` languages `english`, `german` and `french`. Other languages can be added if appropriate wordings are provided to the author. The naming system is designed to work correctly with both `natbib` and `biblatex`.

`notes` The package recognises the `notes` option for controlling how the references section is named. It accepts the values `true`, `false` and `auto`. The Boolean values either enable or disable the addition of “Notes and” to the “References” of the section title. The `auto` option works in conjunction with the `notes2bib`

`nonotes`

`notesbefore`

package. If notes are added, “Notes and” is included in the section title, whereas if no notes are given the section title remains as “References”. The `nonotes` option is equivalent to `notes=false`. The second option for this area is `notesbefore`. This takes `true` and `false` only, and sets whether “Notes and References” or “References and Notes” is produced.

## 5.1 Additional units

`\cmc` Both the `Slunits` and `unitsdef` packages provide methods for handling a range of  
`\Hz` units, but do not include a small number of unit macros useful to chemists. In  
`\molar` contrast, the `siunitx` package provides all of these using the `synchem` module. If  
`\Molar` `Slunits` or `unitsdef` are loaded, `chemstyle` provides the missing units. On the other  
`\mmHg` hand, if no unit package is loaded, `siunitx` is loaded and provides the units itself:  
 if `siunitx` is not available on the system, *these macros are not defined*.

10 cm <sup>3</sup>	<code>\SI{10}{\cmc}</code>	<code>\%</code>
20 Hz	<code>\SI{20}{\Hz}</code>	<code>\%</code>
30 mol dm <sup>-3</sup>	<code>\SI{30}{\molar}</code>	<code>\%</code>
40 M	<code>\SI{40}{\Molar}</code>	<code>\%</code>
50 mmHg	<code>\SI{50}{\mmHg}</code>	

## 5.2 The standard state symbol

`\standardstate` Related to the above, but not exactly a unit is the `\standardstate` command.<sup>1</sup>  
 This generates the tricky `\standardstate` symbol. The symbol will adapt to local  
 sizing.

the standard conditions are indicated: `\standardstate\%`  
 Common but not correct:  
 `$\Delta G_{\mathrm{f}}^{\sim\standardstate}$ or  
 $\Delta G_{\mathrm{f}}^{\sim\standardstate}$ \%  
 Better:  
 $\Delta_{\mathrm{f}}G^{\sim\standardstate}$ \%  
 Sizing:  
 $\int_{T^{\sim\standardstate}}^{T_{\mathrm{out}}}$  
 the standard conditions are indicated:  $\ominus$   
 Common but not correct:  $\Delta G_{\mathrm{f}}^{\ominus}$  or  $\Delta G_{\mathrm{f}}^{\ominus}$   
 Better:  $\Delta_{\mathrm{f}}G^{\ominus}$   
 Sizing:  $\int_{T^{\ominus}}^{T_{\mathrm{out}}}$`

## 5.3 Latin phrases

`\latin` The various Latin phrases commonly used in chemistry are made available as  
`\etc` the obvious commands. By altering the definition of `\latin`, this allows ready  
`\eg` switching from italic to Roman typesetting. Notice that `\etc`, `\ie` and `\eg` are  
`\ie` aware of trailing periods, and so doubling-up should not occur.  
`\etal`  
`\invacuo`

<sup>1</sup>The `\standardstate` macro is only defined if the user does not have their own version.

<i>some text</i>	<code>\latin{some text} \\</code>
<i>e.g.</i>	<code>\eg \\</code>
<i>et al.</i>	<code>\etal \\</code>
<i>etc.</i>	<code>\etc \\</code>
<i>i.e.</i>	<code>\ie \\</code>
<i>in vacuo</i>	<code>\invacuo</code>

**abbrmph** The use of italic for these abbreviations is set by altering the package option `abbrmph`, which takes values `true` and `false`.

<i>et al. i.e.</i>	<code>\etal\ \ie \\</code>
<i>et al. i.e.</i>	<code>\cstsetup{abbrmph=false}</code>
	<code>\etal\ \ie \\</code>

**abbrcomma** For American journals, where it is obligatory to follow “*e.g.*” and “*i.e.*” with a comma, the package provides a mechanism for handling this automatically. Thus, when using an appropriate journal style, `\eg`, `\eg.` and `\eg,` will all result in typesetting “*e.g.,*”. The Boolean package option `abbrcomma` controls this.

<i>e.g.</i> this	<code>\eg this\\</code>
<i>i.e.,</i> that	<code>\cstsetup{abbrcomma=true}</code>
	<code>\ie that</code>

The `\etc` and `\etal` commands are set up on the assumption that they come at the end of a sentence. Hence the spacing after these will default to an inter-sentence space. If you desire an inter-word space, use the normal methods

<i>etc.</i> more text	<code>\etc\ more text \\</code>
<i>et al.</i> have shown	<code>\etal~have shown</code>

**phrases** The definitions of all of the phrases are designed not to overwrite any given by the user *in the preamble*. So, if you have your own `\latin` macro, it will be used even if you load `chemstyle`. If you encounter any problems, try loading the package with the `nophrases` option; this option prevents the package even trying to define any of the phrase macros. The `phrases` option acts as a complement to `nophrases`, so that `phrases=false` is the same as `nophrases=true`.

## 5.4 Alkyl radicals

`\nPr` There are a few alkyl radicals that come up all of the time. No one seems to have put these into a package, so they are provided here. As you would expect, `\iPr` gives *i*-Pr, `\iBu` gives *i*-Bu and `\tBu` gives *t*-Bu, and so on. The style of the output depends on the journal style specified; most journals seem to favour one version of the abbreviation.

<code>\sBu</code>	The alkyl group could be	<code>\nPr</code>	The alkyl group could be
<code>\tBu</code>	<i>n</i> -Pr, <i>i</i> -Pr or <i>n</i> -Bu without	<code>\iPr</code>	<i>n</i> -Pr, <i>i</i> -Pr or <i>n</i> -Bu without
	affecting the selectivity.	<code>\sBu</code>	affecting the selectivity.

These should also work inside the `\ce` macro of `mhchem` when if escaped mode:

$n\text{-Bu}_2\text{CH}-\text{C}(i\text{-Pr})_3$	<code>\ce{\\$ \nBu\$2CH-C(\\$ \iPr\$)3}</code>
--	--

rademph      The appearance of these radical abbreviations is controlled by the package  
radhyphen    options rademph, radhyphen, radprefix and radsuper. The journal styles set  
radprefix    these automatically, but they can be redefined at any point. The options work as  
radsuper    might be expected, and are Boolean switches.

*i*Pr *t*Bu  
Bu-*i* Bu-*s*

```
\cstsetup{
  radhyphen = false,
  radsuper = true}
\iPr \tBu \\
\cstsetup{
  radhyphen = true,
  radsuper = false,
  radprefix = false,
  rademph = false
}
\iBu \sBu
```

xspace      The package can use xspace to automatically insert space after the alkyl  
radical macros. This is controlled using the xspace option.

Some text *i*-Pr some more text  
Some text *i*-Prsome more text

```
\cstsetup{
  xspace = true
}
Some text \iPr some more text\\
\cstsetup{
  xspace = false
}
Some text \iPr some more text
```

## 6 The chemscheme package: graphics for chemistry

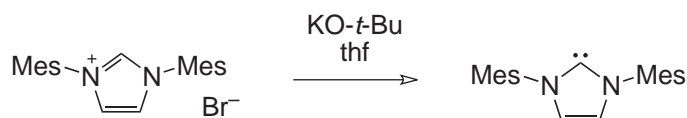
### 6.1 Floating schemes

By default, L<sup>A</sup>T<sub>E</sub>X defines two float types, figure and table. Synthetic chemists make heavy use of schemes, which need a scheme float type. This is provided by chemscheme, in a manner consistent with the kernel floats.

scheme      The package provides a new float type, scheme, accessed in the usual way.

```
\begin{scheme}[ht]
  \includegraphics{scheme-one}
  \caption{A scheme with no compound numbers.}
\end{scheme}
```

The scheme float is designed to behave in the same way as the standard L<sup>A</sup>T<sub>E</sub>X



Scheme 1: A scheme with no compound numbers.

float environments `figure` and `table`. Thus schemes will be placed at the top of a page, where possible. As shown in the example, the use of positional modifiers is allowed. Labelling and referencing schemes also follows the  $\LaTeX$  conventions. `chemscheme` works hard to emulate the document class in use, and so the exact behaviour will depend on whether the standard classes, `KOMA-Script` or `memoir` are being used.

`\schemename`      `\schemename` contains the text used in scheme captions (by default `Scheme`). This is used in the same manner as `\figurename` or `\tablename` to set up the text used in scheme captions.

```
\renewcommand*{\schemename}{Illustration}
\begin{scheme}[ht]
  \includegraphics{scheme-one}
  \caption{A scheme that is not a Scheme!}
\end{scheme}
```

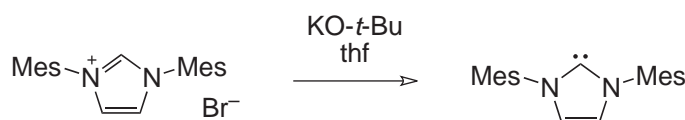


Illustration 2: A scheme that is not a Scheme!

`\listofschemes`      To match the `\listoffigures` and `\listoftables` macros provided by the  $\LaTeX$  kernel, `chemscheme` provides a `\listofschemes` command. This works in the same way as the kernel commands, with the default text stored in the macro `\listschemename`. The standard output is illustrated below.

## List of Schemes

1	A scheme with no compound numbers. . . . .	6
2	A scheme that is not a Scheme! . . . . .	7
3	A scheme with temporary compound numbers. . . . .	8
4	A scheme with automated compound numbers. . . . .	9
5	A scheme with explicitly numbered temporary labels. . . . .	9
6	A scheme with altered label formatting. . . . .	10
7	A flush-left scheme. . . . .	10
8	A flush-right scheme. . . . .	11

`floats`      Which method is used to generate the new float is controlled by the `floats` option, which recognises `float`, `floatrow` and `memoir`. The `memoir` option is ignored if not using the `memoir` class, or if `chemstyle` is in use.<sup>2</sup>

## 6.2 Reference numbers in graphics

There are a number of packages available for tracking compound reference numbers. The two with the most up to date and comprehensive features are

<sup>2</sup>`memoir` is not capable of some of the float manipulation needed by `chemstyle`.

bpchem and chemcompounds. Both allow in-text numbering to be handled automatically. However, neither will allow the use of these numbers directly in schemes, figures, *etc.* Both leave it to the user to manually adapt schemes to match any changes in numbering.

The chemscheme package provides a mechanism for rectifying this issue. The package makes use of the psfrag package, which means that it can only directly produce DVI output (using L<sup>A</sup>T<sub>E</sub>X). However, direct PDF output using pdfL<sup>A</sup>T<sub>E</sub>X is possible: see Section 10. However, it *will not work* with XeTeX as the underlying psfrag package does not with XeTeX.

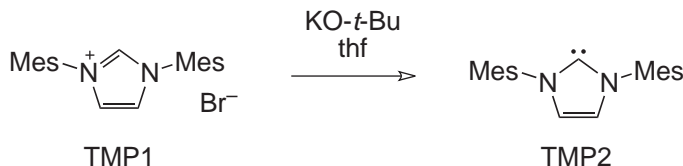
Getting automated numbers into schemes is a two step procedure. In the first step, schemes (or other graphics) should be prepared as normal and saved as encapsulated postscript (EPS) files. The most popular chemistry drawing package, CHEMDRAW, is able to do this from the Save As . . . dialogue. The positions where the auto-labels should be have to be marked in the EPS file. The marker should consist of an “indicator” that the text is to be replaced, followed by a reference number or letter. For automated substitution, the “indicator” text should be the same in all graphics.

`\schemeref` In the second step, the command `\schemeref` is used to indicate the mapping of the temporary markers to the automatically-managed numbering. The syntax of the command is `\chemschemeref[⟨temp-marker⟩]{⟨label⟩}`, where `⟨temp-marker⟩` is the marker used in the graphic, and `⟨label⟩` is the name assigned to the compound by the user. By default, chemscheme will assume that `⟨temp-marker⟩` consists of the marker plus a number, beginning at 1 and incrementing by 1 for each additional structure inside one float. Each replacement requires a separate `\chemschemeref`, all of which should appear before the relevant `\includegraphics` command.

An example will make usage clearer. In the example used in this document, the starting material is given label IMesHCl and the product is called IMes. As is shown in the next example, in the EPS file these are labelled TMP1 and TMP2, respectively. The automated package defaults are used.

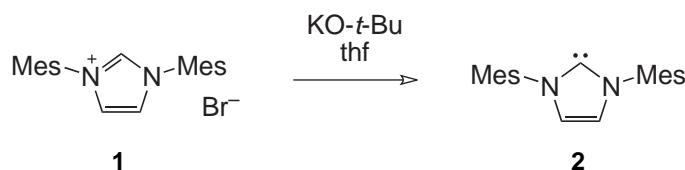
```
\begin{scheme}[ht]
  \includegraphics{scheme-two}
  \caption{A scheme with temporary compound numbers.}
\end{scheme}
\begin{scheme}[ht]
  \schemeref{IMesHCl}
  \schemeref{IMes}
  \includegraphics{scheme-two}
  \caption{A scheme with automated compound numbers.}
\end{scheme}
```

If the marker text is given as an optional argument to `\schemeref`, it must



Scheme 3: A scheme with temporary compound numbers.



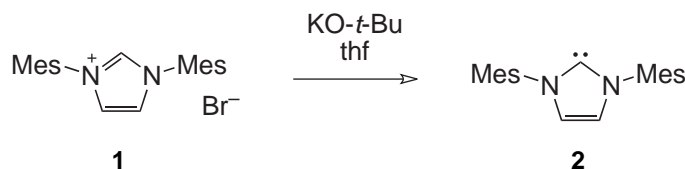


Scheme 4: A scheme with automated compound numbers.

include the entire text to be matched.

```
\begin{scheme}[ht]
\schemeref[TMP1]{IMesHCl}
\schemeref[TMP2]{IMes}
\includegraphics[scheme=two]
\caption{A scheme with explicitly numbered temporary labels.}
\end{scheme}
```

Notice that the new label is centred on the middle of the temporary marker, with



Scheme 5: A scheme with explicitly numbered temporary labels.

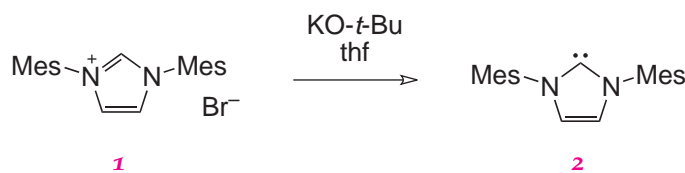
the same baseline. This should allow the user to obtain good alignment of labels and structures.

`\schemerefmarker` When using automatic substitution mode, the “marker” text to be searched for is stored as `\schemerefmarker`, which has default value `TMP`. Thus the graphics should contain labels `TMP1`, `TMP2`, etc. The value can be changed using `\renewcommand*`.

`\schemerefformat` The format of chemical references is controlled by the underlying package, `bpchem` or `chemcompounds`. However, it is useful to be able to specify additional formatting for schemes. By default, `chemscheme` formats all reference numbers in a sans serif font. This is controlled by `\schemerefformat`.

```
% This needs the color or xcolor package loaded
\renewcommand*{\schemerefformat}{%
\color{magenta}\textit
}%
\begin{scheme}[ht]
\schemeref{IMesHCl}
\schemeref{IMes}
\includegraphics[scheme=two]
\caption{A scheme with altered label formatting.}
\end{scheme}
```

`\schemerefsb` The `bpchem` package allows tracking of sub-labels (**1a**, **1b**, etc.). To allow use of these in schemes, the `\schemerefsb` command is provided. This takes an



Scheme 6: A scheme with altered label formatting.

additional argument  $\{\langle sub-label \rangle\}$ , which is used to generate the appropriate text. When used with chemcompounds, this command will gobble its arguments and issue a warning; no substitution will take place.

tracking

The choice of using chemcompounds or bpchem can be made by giving the key-value option `tracking= $\langle package \rangle$`  when loading chemscheme. This recognises the values bpchem and chemcompounds, with the later as the default.

## 7 Horizontal positioning of all floats

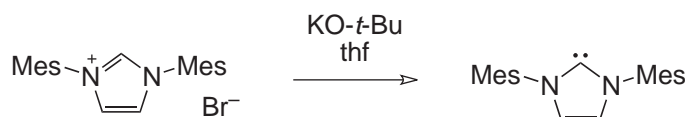
`\floatcontentscentre`  
`\floatcontentscenter`  
`\floatcontentsleft`  
`\floatcontentsright`

The L<sup>A</sup>T<sub>E</sub>X default is to position all float contents flush-left. There is no “hook” provided to alter this. The chemscheme package therefore provides commands to align all float contents automatically. As the macro names make clear, `\floatcontentscentre` will make all floats centred (for users speaking U.S. English, the alternative spelling `\floatcontentscenter` is also available). The default behaviour is restored using the command `\floatcontentsleft`. Finally, `\floatcontentsright` is provided for use if needed. Notice that the float positioning commands should be given *outside* floating environments, and apply to all subsequent floats.

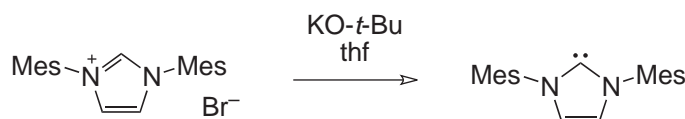
```

\floatcontentsleft
\begin{scheme}[ht]
  \includegraphics{scheme-one}
  \caption{A flush-left scheme.}
\end{scheme}
\floatcontentsright
\begin{scheme}[ht]
  \includegraphics{scheme-one}
  \caption{A flush-right scheme.}
\end{scheme}
\floatcontentscentre

```



Scheme 7: A flush-left scheme.



Scheme 8: A flush-right scheme.

It is important to note that the positioning mechanism used here relies on a low-level hack of the  $\LaTeX$  kernel. This has been tested with the standard  $\LaTeX$  classes, the memoir class and the KOMA-Script bundled. Other document classes may not give the desired behaviour.

## 8 Some thoughts on generating chemical schemes

### 8.1 Overview

There are a number of ways of generating the graphical content of schemes. The choice of method will depend on what is available to the user, and how complex the schemes desired are. In this section, an overview of several approaches is given.<sup>3</sup> The package author, who is a research worker in a university, favours using CHEMDRAW as it is regarded by many synthetic chemists as the best tool for this job. However, this is clearly overkill for users requiring a single diagram on a one-off basis. CHEMDRAW is also a commercial package running only under Windows and MacOS X. The following is necessarily somewhat brief and selective.

### 8.2 Macro-based methods

At the most basic, a chemical scheme is simply a collection of lines and symbols, as with any vector diagram. Hence, it is possible to construct schemes directly using packages such as PSTricks or pgf/Tikz. This is a complex method, and cannot be recommended for anyone except the very experienced and brave.

At a more practical level, there are two packages available which allow typesetting of chemical structures in  $\TeX$ , using specialised commands: XyM $\TeX$  and ppch $\TeX$ . Recent versions of the XyM $\TeX$  package have not been made available on CTAN, and the version held there is therefore considered to be obsolete. On the other hand, the ppch $\TeX$  system, developed originally for Con $\TeX$ t, is available. Both systems suffer from the lack of chemical logic in the input: it is very hard to tell from the code what is being represented. Drawing items such as “curly arrows”, or making subtle alterations to positioning, is very challenging in purely macro-based systems. For these reasons, it is usually much more sensible to examine the available graphical methods.

<sup>3</sup>Thanks to Norwid-R. Behrnd for suggesting this section and giving a number of useful examples and tips.

### 8.3 Graphical methods

Moving to graphical systems, there is no reason that general-purpose vector drawing packages cannot be used for schemes. There are obviously several commercial (CORELDRAW, ADOBE ILLUSTRATOR, *etc.*) and freeware (for example the GIMP) drawing packages that can be used in this way. Simply rings and lines can easily be constructed, although in general-purpose programs the user has to watch that all bonds are the same length.

For producing a large number of complex schemes, the particular abilities of dedicated software become a necessity. As well as the already-mentioned CHEMDRAW, programs such as ISIS DRAW and CHEMSKETCH are available free for personal use;<sup>4</sup> these programs are all Windows specific. In the open-source arena, there are a number of packages such as XDRAWCHEM and BKCHEM, which offer cross-platform functionality. The differences between the various packages are in the ease of use, and ability to generate well-formatted output (for example, aligning structures). A more recent addition to this group of packages is the cross-platform CHEMDOODLE, which look promising but is not free in any sense.

One which deserves mention for the T<sub>E</sub>X user is TpX. This is a general purpose Windows graphics program specifically aimed at producing T<sub>E</sub>X-friendly output (such as PSTricks and Tikz code) from a graphical interface. TpX can accept clipboard data from other programs, so can be used to produce EPS files from programs which do not have native export facilities (such as ISIS DRAW).

## 9 Additional information

### 9.1 Interactions with other packages

chemstyle and chemscheme are designed to avoid, as far as possible, clashes with other packages. The standard packages graphicx and varioref are used by the bundle. If you need to load these with specific options, this needs to be done *before* loading chemstyle (or chemscheme).

### 9.2 Captions above floats

The scheme float type is generated using either the float or floatrow package. This has the side-effect that the placement of captions for floats does not depended on where the \caption command comes inside the floating environment.<sup>5</sup> If you wish to alter the placement of captions, the mechanism of the underlying package will be needed. There are some subtle differences between the two: although floatrow provides the float macros, they are not all 100 % compatible. This document is compiled using floatrow, and so to fix the position of captions the following code is appropriate.

```
\begin{table}[ht]
  \fbox{First float contents}
  \caption{A caption below the float contents in the source.}
\end{table}
\floatsetup[table]{style=plain}
```

---

<sup>4</sup>“Free” as in without charge, not as in open source.

<sup>5</sup>Normally this is a good thing.

```

\begin{table}[ht]
  \fbox{Second float contents}
  \caption{A second caption below the float contents in the source.}
\end{table}

```

Using the float package, the same effect is achieved using:

Table 2: A caption below the float contents in the source.

First float contents

Second float contents

Table 3: A second caption below the float contents in the source.

```

\begin{table}[ht]
  \fbox{First float contents}
  \caption{A caption below the float contents in the source.}
\end{table}
\floatstyle{plain}
\restylefloat{table}
\begin{table}[ht]
  \fbox{Second float contents}
  \caption{A second caption below the float contents in the source.}
\end{table}

```

## 10 A demonstration file

This is a simple demonstration file for using the chemstyle package. By using the auto-pst-pdf package, it is possible to use EPS graphics with pdfLaTeX.

```

1 \demo
2 \documentclass[a4paper]{article}
3 \usepackage[T1]{fontenc}           % Modern font encoding
4 \usepackage[runs=2]{auto-pst-pdf} % Use EPS graphics with pdfLaTeX
5 \usepackage{booktabs}              % Better table layouts
6 \usepackage[journal=rsc]{chemstyle} % Of course!
7 \usepackage{geometry}              % Easy page layout
8 \usepackage{lmodern}               % Use Latin Modern fonts
9 \usepackage[version=3]{mhchem}     % Formula subscripts using \ce{}
10 \begin{document}
11 \end{document}
12 \demo

```

## 11 Change History

v1.0	v2.0f
General: Initial release of package . . . <span style="color: red;">1</span>	General: Detect use of achemso class and abort loading of chemstyle
v2.0a	
General: New varioref option . . . <span style="color: red;">2</span>	and chemscheme . . . . . <span style="color: red;">1</span>

## 12 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>A</b>		<b>L</b>	tracking ..... 10
abbrcomma (option) .... 5		\latin ..... <i>4</i>	varioref ..... 3
abbremph (option) .... 5		\listofschemes ..... <i>7</i>	xspace ..... 6
		\listschemename ..... <i>7</i>	
<b>C</b>		<b>M</b>	<b>P</b>
\ce ..... <i>9</i>		\mmHg ..... <i>4</i>	phrases (option) .... 5
\cmc ..... <i>4</i>		\Molar ..... <i>4</i>	
\cstsetup ..... <i>3</i>		\molar ..... <i>4</i>	<b>R</b>
<b>E</b>			rademph (option) .... 6
\eg ..... <i>4</i>			radhyphen (option) .... 6
environments:			radprefix (option) .... 6
scheme ..... <i>6</i>			radsuper (option) .... 6
\etal ..... <i>4</i>		<b>N</b>	
\etc ..... <i>4</i>		\nBu ..... <i>5</i>	<b>S</b>
		nonotes (option) .... 3	\sBu ..... <i>5</i>
		nophrases (option) .... 5	scheme (environment) .. <i>6</i>
		notes (option) .... 3	\schemename ..... <i>7</i>
		notesbefore (option) .. 3	\schemeref ..... <i>8</i>
		\nPr ..... <i>5</i>	\schemerefformat .... <i>9</i>
<b>F</b>			\schemerefmarker .... <i>9</i>
\floatcontentscenter <i>10</i>		<b>O</b>	\schemerefsb ..... <i>9</i>
\floatcontentscentre <i>10</i>		options:	\standardstate ..... <i>4</i>
\floatcontentsleft . <i>10</i>		abbrcomma ..... 5	
\floatcontentsright <i>10</i>		abbremph ..... 5	<b>T</b>
floats (option) ..... 7		floats ..... 7	\tBu ..... <i>5</i>
		journal ..... 3	tracking (option) .... 10
<b>H</b>		nonotes ..... 3	
\Hz ..... <i>4</i>		nophrases ..... 5	<b>V</b>
		notes ..... 3	varioref (option) .... 3
<b>I</b>		notesbefore ..... 3	
\iBu ..... <i>5</i>		phrases ..... 5	<b>X</b>
\ie ..... <i>4</i>		rademph ..... 6	xspace (option) ..... 6
\invacuo ..... <i>4</i>		radhyphen ..... 6	
\iPr ..... <i>5</i>		radprefix ..... 6	
		radsuper ..... 6	
<b>J</b>			
journal (option) .... 3			