

# chemstyle — Writing chemistry with style\*

Joseph Wright<sup>†</sup>

Released 2008/03/21

## Abstract

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

<b>1</b>	<b>Introduction</b>	<b>1</b>	<b>7</b>	<b>The package code</b>	<b>8</b>
			7.1	Setup code . . . . .	8
			7.2	Option handling . . . . .	8
			7.3	Extra units . . . . .	10
			7.4	Standard state symbol . . . . .	11
			7.5	Alkyl radicals . . . . .	11
			7.6	Float formatting . . . . .	12
			7.7	Cross-references . . . . .	12
			7.8	Latin phrases . . . . .	13
			7.9	Loading the style definition . . . . .	14
			7.10	Handling reference section naming . . . . .	15
<b>2</b>	<b>Style options for chemstyle</b>	<b>2</b>	<b>8</b>	<b>Configuration files</b>	<b>17</b>
2.1	Creating styles . . . . .	3	8.1	RSC style . . . . .	17
2.2	Package options . . . . .	3	8.2	<i>Angew. Chem.</i> style . . . . .	17
			8.3	<i>J. Organomet. Chem.</i> style . . . . .	17
<b>3</b>	<b>Naming of the references section</b>	<b>3</b>	8.4	<i>Tetrahedron Lett.</i> style . . . . .	17
			8.5	<i>J. Am. Chem. Soc.</i> style . . . . .	18
<b>4</b>	<b>Additional macros</b>	<b>4</b>	8.6	<i>Inorg. Chem.</i> style . . . . .	18
4.1	Additional units . . . . .	4	8.7	<i>J. Phys. Chem.</i> style . . . . .	18
4.2	The standard state symbol . . . . .	4	8.8	<i>Org. Lett.</i> style . . . . .	19
4.3	Alkyl radicals . . . . .	4	8.9	The empty style — none . . . . .	20
4.4	Latin phrases . . . . .	5	<b>9</b>	<b>Change History</b>	<b>20</b>
<b>5</b>	<b>Additional information</b>	<b>6</b>	<b>10</b>	<b>Index</b>	<b>21</b>
5.1	Advice for users of the rsc package . . . . .	6			
5.2	Interactions with other packages . . . . .	6			
5.3	Captions above floats . . . . .	6			
<b>6</b>	<b>A template for chemical articles</b>	<b>7</b>			

\*This file describes version v1.2a, last revised 2008/03/21.

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

Table 1: Styles provided by chemstyle

Option	Journals using this style
none	Not applicable
angew	<i>Angew. Chem., Chem. Eur. J.</i>
jomc	<i>J. Organomet. Chem., Coord. Chem. Rev.</i>
ic	<i>Inorg. Chem.</i>
jacs	<i>J. Am. Chem. Soc.</i>
jcp	<i>J. Phys. Chem. A, J. Phys. Chem. B</i>
orglett	<i>Org. Lett.</i>
rsc	<i>Chem. Commun., Org. Biomol. Chem.</i> <i>Dalton Trans.</i>
tetlett	<i>Tetrahedron, Tetrahedron Lett.</i>

## 1 Introduction

The aim of chemstyle is to provide a quick 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. chemstyle has grown out of the rsc package, which had a similar aim but was much more limited (and less robustly implemented). The chemstyle package is also designed with the use of biblatex in mind: the rsc package is closely bound to traditional BibTeX use.

As a successor to the rsc package, chemstyle provides a range of chemistry-related additional macros. The set provided here is an extended version of those provide by rsc. Everything that can be done using the rsc L<sup>A</sup>T<sub>E</sub>X package is therefore possible using the chemstyle package.

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.

## 2 Style options for chemstyle

journal chemstyle should be loaded with a package option specifying which journal style to follow. Currently, chemstyle is aware of a the styles listed in Table 1. New styles can be developed by creating a new file modelled on the existing definitions; chemstyle will automatically search for correctly-named styles. The style files provided with chemstyle have been derived from current practice in the target journals. It is not always easy to pick the correct stylistic settings from (sometimes inconsistent) real-world examples. The package author welcomes feedback on the styles provided.

The none style is notable as it is not based on a journal. Instead this is a minimal style, which provides the additional commands without making formatting changes. It is also the default style if no option is given. This

document has been compiled implicitly using the `none` option, for example.

## 2.1 Creating styles

The process of creating a new style for `chemstyle` is intended to be relatively easy. New styles should be saved as files with the extension `.jdf` (standing for Journal Definition File), and should be saved somewhere in the path searched by  $\text{\TeX}$ .<sup>1</sup> The definition files included in the package should provide a guide to the basic options available for producing new styles. Arbitrary  $\text{\TeX}$  commands can be included, if they are necessary for a particular style. For example, other packages can be loaded in `\usepackage`.

The maintainer of `chemstyle` is happy to add new styles to the package, either by contribution by users or on request (when he has sufficient time!). If you have a new style to add (or corrections to an existing one), please contact the package author.

## 2.2 Package options

<code>journal</code>	As of version 1.2, <code>chemstyle</code> uses the <code>keyval</code> system for option management. The choice of journal style can therefore be given as a normal option or by using the key <code>journal=</code> . The other package options provided by the package are described as they arise in the documentation.
<code>\cstsetup</code>	Some of the package options can only be set when loading the package. The choice of journal style is one of these. Others can be altered anywhere in the document, using the <code>\cstsetup</code> macro. This accepts a <code>keyval</code> list and processes it as needed.
<code>chemcompounds</code>	<code>chemstyle</code> loads the <code>chemscheme</code> package, and therefore will pass through the <code>optchemcompounds</code> option.

## 3 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 “Notes and References” naming commands are language-aware, *via* the `babel` interface. Currently, `chemstyle` includes appropriate labels for `babel` languages `english`, `UKenglish`, `ngerman` 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`.

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

<sup>1</sup>Depending on your  $\text{\TeX}$  distribution, you may need to rebuild your file database after creating a new style. For Mik $\text{\TeX}$  users, this can be done graphically or by typing `mpm -update-db` at the command line; for TeXLive, run `texhash`.

notesbefore 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.

## 4 Additional macros

### 4.1 Additional units

<code>\Hz</code>	Both the <code>Slunits</code> and <code>unitsdef</code> packages provide methods for handling a range
<code>\mmHg</code>	of units, but do not include a small number of unit macros useful to chemists.
<code>\molar</code>	The package therefore provides these, using <code>Slunits</code> by default. <sup>2</sup> The definitions
<code>\Molar</code>	follow the pattern of the appropriate package.
<code>\cmc</code>	
10 Hz	<code>\unit{10}{\Hz}\</code>
20 mmHg	<code>\unit{20}{\mmHg}\</code>
30 mol dm <sup>-3</sup>	<code>\unit{30}{\molar}\</code>
40 M	<code>\unit{40}{\Molar}\</code>
50 cm <sup>3</sup>	<code>\unit{50}{\cmc}</code>

The `si` package is currently under development as a replacement for `Slunits` and `unitsdef`.<sup>3</sup> `si` includes definitions for the above units, and so if it is loaded the definitions are taken from there, rather than provided by `chemstyle`. Looking to the future, once `si` reaches version 1.0, it will become the default support package for `chemstyle`.

### 4.2 The standard state symbol

`\standardstate` Related to the above, but not exactly a unit is the `\standardstate` command.<sup>4</sup> This generates the tricky `\standardstate` symbol.

the standard conditions are indicated: $\Delta S^\ominus$	the standard conditions are indicated: <code>\standardstate\</code> <code>\$\Delta S\standardstate\$</code>
---	--

### 4.3 Alkyl radicals

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

The alkyl group could be *n*-Pr, *i*-Pr or *n*-Bu without affecting the selectivity.

The alkyl group could be `\nPr`, `\iPr` or `\nBu` without affecting the selectivity.

`xspace` When `chemstyle` is loaded using the `xspace` option, the `xspace` package is automatically used to add space after the command names, so that `\iPr` group will result in “*i*-Pr group” being typeset.

`radhyphen` The appearance of these radical abbreviations is controlled by the package

`rademph`

`radsuper`

`radprefix`

<sup>2</sup>This is a change in v1.2: previous versions used `unitsdef`.

<sup>3</sup>By the maintainer of `chemstyle`.

<sup>4</sup>The `\standardstate` macro is only defined if the user does not have their own version.

options `radhyphen`, `rademph`, `radsuper` and `radprefix`. The journal styles set these automatically, but they can be redefined at any point. The options work as might be expected, and are Boolean switches.

$iPr$ $tBu$ $Bu-i$ $Bu-s$	<pre>\cstsetup{radhyphen=false,radsuper=true}\\ \iPr \tBu \cstsetup{radhyphen=true,radsuper=false,% radprefix=false,rademph=false}\\ \iBu \sBu</pre>
------------------------------	--

#### 4.4 Latin phrases

$\backslash latin$ $\backslash etc$ $\backslash eg$ $\backslash ie$ $\backslash etal$ $\backslash invacuo$ $abbr emph$	<p>The various Latin phrases commonly used in chemistry are made available as the obvious commands. By altering the definition of <math>\backslash latin</math>, this allows ready switching from italic to Roman typesetting. Notice that <math>\backslash etc</math>, <math>\backslash ie</math> and <math>\backslash eg</math> are aware of trailing periods, and so doubling-up should not occur. Once again, these macros use <code>xspace</code> when given as a package option to handle automatic addition of spaces after these phrases.</p> <p>The use of italic for these abbreviations is set by altering the package option <code>abbr emph</code>, which takes values <code>true</code> and <code>false</code>.</p>
--	---

<i>et al.</i> or <i>in vacuo</i> plus some text <i>i.e.</i> equals <i>e.g.</i> perhaps etc.	<pre>\etal or \invacuo plus \latin{some text}\\ \cstsetup{abbr emph=false} \ie equals \eg perhaps \etc</pre>
--	--

`abbr comma` 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,  $\backslash eg$ ,  $\backslash eg.$  and  $\backslash eg,$  will all result in typesetting “*e.g.,*”. The Boolean package option `abbr comma` controls this.

<i>e.g.</i> this <i>i.e.,</i> that	<pre>\eg this\\ \cstsetup{abbr comma=true} \ie that</pre>
---------------------------------------	---

The  $\backslash etc$  and  $\backslash etal$  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 <i>et al.</i> have shown	<pre>\etc\ more text \\ \etal~have shown</pre>
---	--

`nophrases` 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  $\backslash latin$  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.

$\backslash latin emph on$ $\backslash latin emph off$	<p>For backward-compatibility with previous versions of <code>chemstyle</code>, the macros <math>\backslash latin emph on</math> and <math>\backslash latin emph off</math> are provided. These work as would be expected, to alter the formatting produced by <math>\backslash latin</math>. The more general <code>keyval</code> method is now preferred for making this change.</p>
---	--

## 5 Additional information

### 5.1 Advice for users of the rsc package

The chemstyle package is intended as a replacement for the rsc package. As such, it covers almost everything the the rsc L<sup>A</sup>T<sub>E</sub>X package does, and more. Users of rsc are strongly encouraged to update to using chemstyle. The bibliography styles provided by rsc will continue to be required, of course. Migration of these styles to biblatex is an on-going project.

### 5.2 Interactions with other packages

The chemstyle package has been designed to avoid, as far as possible, clashes with other packages. The package requires the presence of the standard graphicx and varioref packages. If these packages need to be loaded with explicit options, this should be done *before* loading chemstyle. The chemscheme package is also needed, as it provides the floating scheme environment essential in synthetic chemistry documents.

### 5.3 Captions above floats

The scheme float type is generated using the float 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 float package will be needed. This requires the use of \floatstyle to load the desired settings, followed by \restylefloat to apply the settings. This is most usually done with tables.

```
\begin{table}[h]
  The float contents
  \caption{A caption below the float contents in the source}
\end{table}
\floatstyle{plain}
\restylefloat{table}
\begin{table}[h]
  The second float contents, which should be above the caption
  \caption{A second caption below the float contents
    in the source}
\end{table}
```

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

The float contents

The second float contents, which should be above the caption

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

---

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

## 6 A template for chemical articles

This is a very simple template for chemistry-related documents. Hopefully it contains a few extra hints for getting well-formatted documents quickly. For simplicity, the template assumes that the user is writing a thesis for a U.K. university. Hence it uses U.K. defaults and RSC-based styling. Most of the packages used have good documentation, but a brief summary of why they are recommended follows. The file `chemstyle.ltx` will be generated by compiling this documentation, and contains the example “ready to run”.

```
\documentclass[fontsize=10pt,paper=a4,english,UKenglish]
{scrreprt}
\usepackage{geometry,upgreek,booktabs,babel}
\usepackage[journal=rsc,xspace=true]{chemstyle}
\usepackage[version=3]{mhchem}
\usepackage[footnotes]{notes2bib}
\usepackage[final]{microtype}
\usepackage[colorlinks]{hyperref}
\begin{document}
  Document contents go here
\end{document}
```

The versatile KOMA-script bundle provides more advanced versions of the standard document classes. If you want paragraphs separated out, with no indents (a common style for theses), add the `parskip` option to the font and paper size ones given here.

The `babel` system is loaded to sort out hyphenation and so on, and could be useful if there are any foreign-language quotes.

`geometry` allows the users to alter page layout with ease: much better than trying to hack the raw L<sup>A</sup>T<sub>E</sub>X system. `booktabs` gives much nicer looking tables than the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> default. `upgreek` provides non-italic lowercase Greek letters, which should be used for things such as bond descriptions.

$\pi$ -bond  $\sigma^*$ -orbital

`\uppi` $\pi$ -bond `\upsigma` $\sigma^*$ -orbital

Load `chemstyle` (of course) to give not only some easy formatting, but also to automatically provide a float type for schemes, thanks to the `chemscheme` package. This also loads either `chemcompounds` or (optionally) `bpchem` to track compound numbers.

The `mhchem` package provides the `\ce` command for rapidly typesetting formulas, so that you can type `\ce{H2SO4}` and get H<sub>2</sub>SO<sub>4</sub>.<sup>6</sup>

The `microtype` package improves formatting when used with the pdfT<sub>E</sub>X engine. By giving the `final` option, it is active even when using `draft` as a global option.

Using `notes2bib` allows the user to automatically add notes to the bibliography from within the document body. So you can put `\bibnote{A note}` in the source, and this will move into the References section without any further effort. The `footnotes` option means that footnotes do the same.

Finally, the `hyperref` package makes headings, citations and so on into hyperlinks.

<sup>6</sup>There is a slight cheat here, as this document uses lower-case numerals in the text. The example is written as `\ce{H2SO4}`.

## 7 The package code

### 7.1 Setup code

First of all, the package identifies itself and loads other packages needed to function. Loading chemscheme is deferred until later, to allow processing of options at the correct point.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{chemstyle}
3 [2008/03/21 v1.2a Writing chemistry with style]
```

Packages that are needed under all circumstances are loaded here. There are some limitations on the age of xkeyval.

```
4 \RequirePackage{xkeyval}
5 \@ifpackagelater{xkeyval}{2005/05/07}
6 {}
7 {\PackageError{chemstyle}
8   {xkeyval >= 2.5 required}
9   {chemstyle requires the 'xkeyval' package, version 2.5
10    or later.\MessageBreak The version loaded is:
11    '\@nameuse{ver@xkeyval.sty}'.\MessageBreak
12    This is a fatal error: loading will abort.}
13  \endinput}
14 \RequirePackage{graphicx,varioref,caption,xspace}
```

\cst@tempa A couple of temporary private macros.

```
\cst@tempb 15 \newcommand*{\cst@tempa}{}
16 \newcommand*{\cst@tempb}{}

```

### 7.2 Option handling

\cst@boolkey Option handling is processed using xkeyval. As a number of Boolean keys are needed, a shortcut is handy.

```
17 \newcommand*{\cst@boolkey}[1]{
18   \define@boolkey[cst]{opt}[cst@]{#1}[true]{}

```

\ifcst@chemcompounds The Boolean keys are now defined.

```
\ifcst@xspace 19 \cst@boolkey{chemcompounds}
\ifcst@phrases 20 \cst@boolkey{xspace}
\ifcst@radprefix 21 \cst@boolkey{phrases}
\ifcst@rademph 22 \cst@boolkey{radprefix}
\ifcst@radsuper 23 \cst@boolkey{rademph}
\ifcst@radhyphen 24 \cst@boolkey{radsuper}
\ifcst@abbremph 25 \cst@boolkey{radhyphen}
\ifcst@abbrcomma 26 \cst@boolkey{abbremph}
\ifcst@notesbefore 27 \cst@boolkey{abbrcomma}
28 \cst@boolkey{notesbefore}

```

\ifcst@notes For notes, some choices are available. These need a bit of testing.

```
\ifcst@autonotes 29 \newif\ifcst@notes
\cst@tempa 30 \newif\ifcst@autonotes
\cst@tempb 31 \define@choicekey*{cst}{opt}{notes}[\cst@tempa]
32 {true,false,auto}[true]

```



```

33  {\cst@notesfalse
34  \cst@autonotesfalse
35  \edef\cst@tempb{auto}
36  \ifx\cst@tempa\cst@tempb
37    \cst@notesttrue
38    \cst@autonotesttrue
39  \fi
40  \edef\cst@tempb{true}
41  \ifx\cst@tempa\cst@tempb
42    \cst@notesttrue
43  \fi}

```

Some old options are no longer needed at all.

```

44 \DeclareOptionX[cst]<opt>{siunits}
45  {\PackageInfo{chemstyle}{Option siunits no longer required}}
46 \DeclareOptionX[cst]<opt>{SIunits}
47  {\PackageInfo{chemstyle}{Option siunits no longer required}}
48 \DeclareOptionX[cst]<opt>{nonotes}
49  {\ExecuteOptionsX[cst]<opt>{notes=false}}
50 \DeclareOptionX[cst]<opt>{nophrases}
51  {\ExecuteOptionsX[cst]<opt>{phrases=false}}

```

\cst@journal  
\cst@language

The journal option can be given with or without journal=.

```

52 \define@cmdkeys[cst]{opt}[cst@]{journal,language}
53 \DeclareOptionX*{\edef\cst@journal{\CurrentOption}}

```

The default options are executed, and the user options are processed.

```

54 \setkeys[cst]{opt}{
55  notes=true,
56  phrases=true,
57  chemcompounds=true,
58  xspace=true,
59  journal=none,
60  radprefix=true,
61  rademph=true,
62  radsuper=false,
63  radhyphen=true,
64  abbrevemph=true,
65  abbrcomma=false,
66  language=english,
67  notesbefore=true}
68 \ProcessOptionsX[cst]<opt>

```

With the options processed, those to be passed to chemscheme are sorted out.

```

69 \ifcst@chemcompounds
70   \PassOptionsToPackage{chemcompounds}{chemscheme}
71 \fi
72 \RequirePackage{chemscheme}

```

\cst@disablekey

Some of the options are no longer relevant once the package is loaded. These are disabled here with a suitable message.

```

73 \newcommand*{\cst@disablekey}[1]{
74   \define@key[cst]{opt}{#1}{\PackageWarning{chemstyle}
75     {Option '#1' only valid when loading package}}}

```

```

76 \AtBeginDocument{
77   \cst@disablekey{journal}
78   \cst@disablekey{phrases}
79   \cst@disablekey{nophrases}
80   \cst@disablekey{notes}
81   \cst@disablekey{nonotes}
82   \cst@disablekey{chemcompounds}}

```

`\cstsetup` A macro for changing the settings at run-time without needing to know the internals of the package.

```

83 \newcommand*{\cstsetup}[1]{%
84   \iffalse{\fi\ifnum0=}\fi
85   \setkeys[cst]{opt}{#1}%
86   \ifnum0=}\fi\iffalse}\fi}

```

### 7.3 Extra units

The upcoming `si` package<sup>7</sup> will include the extra chemistry units as a loadable modules (`alsoload=synchem`). A series of checks are therefore made before any definitions are given. If `si` is loaded, the necessary part of that package is loaded; the `\si@requirecfgs` macro from that package is used. The `\makeatletter` macro is needed here as otherwise the configuration file is loaded with the wrong category codes.

```

87 \AtBeginDocument{
88   \@ifpackageloaded{si}
89   {\makeatletter
90     \si@requirecfgs{synchem}
91     \makeatother}

```

`\cmc` Neither `unitsdef` nor `Slunits` define these two macros.

```

\Hz 92   {\newcommand*\cmc}{\cubiccentimeter}
93   \newcommand*\Hz}{\hertz}

```

`\cubiccentimeter` A check is for the `unitsdef` package, so that the package-dependant definitions can be given.

```

\molar 94   \@ifpackageloaded{unitsdef}
\mmHg 95   {\newunit{\cubiccentimeter}{\cm\unitsuperscript{3}}
\mol 96   \newunit{\Molar}{\textsc{m}}
97   \newunit{\molar}{\mole\unitsep\dm\unitsuperscript{--3}}
98   \newunit{\mmHg}{mmHg}
99   \newunit{\mol}{\mole}}

```

Two checks are made for `Slunits` to allow for odd capitalisation. If it is not found, then the package is loaded.<sup>8</sup>

```

100   {\@ifpackageloaded{SIunits}}{}
101   {\@ifpackageloaded{siunits}}{}
102   {\RequirePackage{SIunits}}}
103   \newcommand*\cubiccentimeter}{\centi\metre\cubed}
104   \newcommand*\Molar}{\textsc{m}}
105   \newcommand*\molar}{\mole\usk\deci\metre\rpcubed}

```

<sup>7</sup>An improved replacement for all of the existing unit packages.

<sup>8</sup>This will change when `si` reaches version 1.0.

```

106      \newcommand*{\mmHg}{\milli\metre Hg}
107      \newcommand*{\mol}{\mole}}}}

```

## 7.4 Standard state symbol

`\cst@varnothing` The “standard state” symbol is handy. This is produced by rotating the “varnothing” symbol from the AMS set. Note that the rotation angle here has been carefully checked, but is set by eye. The symbol is loaded directly here, rather than using the `amssymb` package, to avoid any clashes.

`\standardstate`

```

108 \DeclareSymbolFont{CSTAMS}{U}{msb}{m}{n}
109 \DeclareMathSymbol{\cst@varnothing}{\mathord}{CSTAMS}{"3F}
110 \providecommand*{\standardstate}%
111   {\textsuperscript{\rotatebox[origin=c]{138.8}%
112     {\ensuremath{\cst@varnothing}}}}

```

## 7.5 Alkyl radicals

`\cst@radical` The `\cst@radical` macro does the hard work of declaring each abbreviation. The commands are made robust so they can (hopefully) be used anywhere.

```

113 \newcommand*{\cst@radical}[2]{%
114   \expandafter\DeclareRobustCommand\expandafter*\expandafter{%
115     \csname #1#2\endcsname}%

```

`\cst@emph` To allow redefinition while running, the various component parts of the system are defined each time an abbreviation is used.

`\cst@hyphen`

`\cst@super`

```

116   \ifcst@radhyphen
117     \def\cst@hyphen{-}%
118   \else
119     \let\cst@hyphen\relax
120   \fi
121   \ifcst@radsuper
122     \let\cst@super\textsuperscript
123   \else
124     \let\cst@super\relax
125   \fi
126   \ifcst@rademph
127     \let\cst@emph\emph
128   \else
129     \let\cst@emph\relax
130   \fi

```

The final check is for a prefix, which is easiest to do here.

```

131   \ifcst@radprefix
132     \cst@super{\cst@emph{#1}}\cst@hyphen#2%
133   \else
134     #2\cst@hyphen\cst@super{\cst@emph{#1}}%
135   \fi
136   \cst@xspace}}

```

`\cst@xspace` A re-useable `\xspace`-based macro.

```

137 \newcommand*{\cst@xspace}{%
138   \ifcst@xspace

```

```

139     \expandafter\xspace%
140     \fi}

\nPr With a mechanism in place, the abbreviations are declared. The format of the
\nPr resulting output will depend upon the configuration file used.
\nBu 141 \cst@radical{n}{Pr}%
\nBu 142 \cst@radical{i}{Pr}%
\nBu 143 \cst@radical{n}{Bu}%
\nBu 144 \cst@radical{i}{Bu}%
\nBu 145 \cst@radical{s}{Bu}%
\nBu 146 \cst@radical{t}{Bu}%

```

## 7.6 Float formatting

The next step is to format the floats correctly. Unfortunately, memoir does not provide all of the commands needed to achieve this. Thus the float package is needed; in order to load it, the `\newfloat` command in memoir is killed off. The chemscheme package does not load float when memoir is being used, because creating a new float type is catered for directly by memoir. Unfortunately, the memoir system isn't flexible enough for what is needed by chemstyle, so the hard work of chemscheme is undone here! For other document classes, float will already have been loaded by chemscheme.

```

147 \@ifclassloaded{memoir}
148   {\let\newfloat\@undefined
149     \RequirePackage{float}}
150   {}

```

The standard float types are now restyled to place the captions correctly (for most journals). Normally in chemical documents the author expects the float to be “here” if possible; this is therefore set as the default.

```

151 \floatstyle{plaintop}
152 \restylefloat{table}
153 \floatstyle{plain}
154 \restylefloat{figure}
155 \floatplacement{table}{htbp}
156 \floatplacement{figure}{htbp}

```

Work with schemes needs to be done after hyperref might be loaded.

```

157 \AtBeginDocument{
158   \restylefloat{scheme}
159   \floatplacement{scheme}{htbp}}

```

The contents of floats are centred by default, using the hook from the chemscheme package.

```

160 \floatcontentscentre

```

## 7.7 Cross-references

The naming for cross-references is sorted out properly using the varioref package.

```

161 \labelformat{figure}{\figurename~#1}
162 \labelformat{table}{\tablename~#1}
163 \AtBeginDocument{\labelformat{scheme}{\schemename~#1}}

```

## 7.8 Latin phrases

`\latin` A series of Latin phrases are provided, with a quick switch to print them in Roman letters if needed. A mechanism is needed to alter the effect of the `\latin` command *only* if the user does not have their own version. This is achieved here, with precautions taken to ensure the user can define their own `\latin` command *after* loading `chemstyle` and still have everything work properly.

```
164 \newcommand*{\cst@latin}{%
165   \ifcst@abbremph
166     \expandafter\emph%
167   \fi}
168 \newcommand*{\latinemphon}{\cstsetup{abbremph=true}}
169 \newcommand*{\latinemphoff}{\cstsetup{abbremph=false}}
170 \AtBeginDocument{\providecommand*\latin{\cst@latin}}
```

`\etc` For the macros themselves, care is taken about trailing full stops. The `\xspace` command deals with any problems of spacing. Things could go wrong with complex punctuation, as no other checks are performed. All of these functions use `\providecommand` to avoid standing on the user's own versions, if they exist.

```
171 \ifcst@phrases
172   \AtBeginDocument{
173     \providecommand*\etc{
174       {\@ifnextchar.{\cst@etc}{\cst@etc.\cst@xspace}}
175     \providecommand*\invacuo{
176       {\latin{in vacuo}\cst@xspace}
177     \providecommand*\etal{
178       {\@ifnextchar.{\cst@etal}{\cst@etal.\cst@xspace}}
179     \providecommand*\eg{
180       {\ifcst@abbrcomma
181         \expandafter\cst@commaabbr%
182       \else
183         \expandafter\cst@nocommaabbr%
184       \fi
185       {eg}}
186     \providecommand*\ie{
187       {\ifcst@abbrcomma
188         \expandafter\cst@commaabbr%
189       \else
190         \expandafter\cst@nocommaabbr%
191       \fi
192       {ie}}}
```

`\cst@etal` Internal macros are used for items ending in a full stop, to allow clean handling of spacing. Notice that `\ie` and `\eg` cannot come at the end of a sentence, they are designed to give only an inter-word space.

```
\cst@etc
\cst@eg
\cst@ie
193 \newcommand*{\cst@etal}{\latin{et~al}}
194 \newcommand*{\cst@etc}{\latin{etc}}
195 \newcommand*{\cst@ie}{\latin{i.e\spacefactor999\relax}}
196 \newcommand*{\cst@eg}{\latin{e.g\spacefactor999\relax}}
```

`\cst@commaabbr` To handle the comma issue, two general macros are used.

```
\cst@nocommaabbr
197 \newcommand*{\cst@commaabbr}[1]{%
```

```

198 \csname cst@#1\endcsname%
199 \cst@addpunct}
200 \newcommand*{\cst@nocommaabbr}[1]{%
201 \ifnextchar.%
202 {\csname cst@#1\endcsname}
203 {\csname cst@#1\endcsname.\cst@xspace}}

```

`\cst@addpunct` The following macros are very closely based on those in the `cite` package used for moving citations after punctuation. The first macro is used as an initial hook. Notice that `\relax` is essential here, as it provides an argument for `\cst@addpunct` in the first round of checking.

```

204 \newcommand*{\cst@addpunct}{%
205 \cst@add@punct\relax}

```

`\cst@add@punct` Here, a plain TeX `\futurelet` is used to test the next character. Notice that this macro takes a single argument, which is used to recursively gobble up punctuation.

```

206 \newcommand*{\cst@add@punct}[1]{%
207 \futurelet\cst@tempa\cst@add@punct@}

```

`\cst@add@punct@` The checking occurs here. If a match is made, then the process is repeated to allow the punctuation to be gobbled.

```

\cst@tempa
\cst@tempb
208 \newcommand*{\cst@add@punct@}{%
209 \ifx.\cst@tempa
210 \let\cst@tempb\cst@add@punct
211 \else
212 \ifx,\cst@tempa
213 \let\cst@tempb\cst@add@punct
214 \else
215 \let\cst@tempb\cst@commapunct
216 \fi
217 \fi
218 \cst@tempb}

```

No match, and so new punctuation is to be added and the loop ended.

`\cst@commapunct` The `\cst@punct` macro holds the comma-containing punctuation to be added.

```

219 \newcommand*{\cst@commapunct}{.,\cst@xspace}

```

## 7.9 Loading the style definition

The style definition is loaded here, once everything necessary is in place. A style must be loaded, so a default is provided to be on the safe side. The journal style file must have extension `.jdf`.

```

220 \InputIfFileExists{\cst@journal.jdf}
221 {\PackageInfo{chemstyle}
222 {Loaded \cst@journal.jdf}}
223 {\PackageWarning{chemstyle}
224 {Requested style '\cst@journal' does not exist}\MessageBreak
225 Loading default style}
226 \InputIfFileExists{none.jdf}{}
227 {\PackageError{chemstyle}{Failed to load none.jdf}
228 {The default journal definition file was not

```

```

229         found\MessageBreak This is a core part of chemstyle
230         \MessageBreak Something is wrong with the installation}}}
```

## 7.10 Handling reference section naming

`\cst@refsec` To enable the renaming of `\bibname` or `\refname` to the desired value, some work is needed. Both `natbib` and `biblatex` alter the behaviour here subtly. The `babel` package also requires careful handling. The initial step is to find whether the document class defines chapters, as this affects whether `\refname` or `\bibname` should be changed. This is done inside a group so that a false definition is not left around.<sup>9</sup> A second check is needed, in case the document class does not define any reference section at all (for example, `minimal` does not).

```

231 \begingroup
232   \@ifundefined{chapter}
233   {\@ifundefined{refname}
234     {\PackageInfo{chemstyle}
235       {No bibliography name command found}
236       \cstsetup{notes=false}}
237     {\gdef\cst@refsec{\refname}}}}
238   {\gdef\cst@refsec{\bibname}}
239 \endgroup
```

`\c@bibnote` The business-end of the work is only carried out if the `notes` option is set to `true` or `auto`. The check above means that a reference section is also defined. First, some checking for the `auto` option occurs; the counter used by `notes2bib` needs to be defined even if the package is not loaded.

```

240 \ifcst@autonotes
241   \AtBeginDocument{
242     \@ifpackageloaded{notes2bib}{}
243     {\newcounter{bibnote}}}
244 \fi
```

`\cst@refname` The internal macros to actually display the reference section name deals with the various options. The `\expandafter`s are for caution here: nothing *should* go wrong with the nested `\if` statements.

```

\cst@ref@name
\cst@ref@name@
245 \newcommand*{\cst@refname}{%
246   \ifcst@notes
247     \ifcst@autonotes
248       \ifnum\the\value{bibnote}>\z@
249         \expandafter\expandafter\expandafter\expandafter\expandafter
250         \expandafter\expandafter\cst@ref@name%
251       \else
252         \expandafter\expandafter\expandafter\expandafter\expandafter
253         \expandafter\expandafter\cst@ref@name@%
254       \fi
255     \else
256       \expandafter\expandafter\expandafter\cst@ref@name%
257     \fi
258   \else
259     \expandafter\cst@ref@name@%
```

<sup>9</sup>If `\@undefined` is ever fixed, this will no longer be needed. If  $\epsilon$ -TeX was required, the `\ifdefined` would be a choice here; however,  $\epsilon$ -TeX is not going to be required just for this.

```

260 \fi}
261 \newcommand*{\cst@ref@name}[3]{%
262 \ifcst@notesbefore
263 #3\space#2\space#1%
264 \else
265 #1\space#2\space#3%
266 \fi}
267 \newcommand*{\cst@ref@name@}[3]{#1}

```

`\cst@setrefname` Setting up the section name is a tricky affair. If no section is defined, then the whole process is skipped. Otherwise, the first check made is if the default language matches the definition given. If so, the heading is changed now.

```

268 \@ifundefined{cst@refsec}
269 {\let\cst@setrefname\@gobblefour}
270 {\newcommand*{\cst@setrefname}[4]{
271 \edef\cst@tempa{#1}%
272 \ifx\cst@tempa\cst@language
273 \expandafter\renewcommand\expandafter*\expandafter{%
274 \cst@refsec}
275 {\cst@refname{#2}{#3}{#4}}
276 \fi

```

`cst@tempa` At the beginning of the document, a check needs to be made for `babel` and `biblatex`. For the later, a low-level hack is made to attach to the mechanism. For `babel` the appropriate commands are added to the `\captions...` macro.

```

277 \AtBeginDocument{
278 \ifpackageloaded{biblatex}
279 {\def\cst@tempa{\csgappto{bib@strings@#1}}
280 \expandafter\cst@tempa\expandafter{\expandafter\renewcommand%
281 \expandafter*\expandafter{\cst@refsec}{%
282 \cst@refname{#2}{#3}{#4}}}}
283 {\ifpackageloaded{babel}
284 {\def\cst@tempa{\expandafter\addto\expandafter{%
285 \csname captions#1\endcsname}}
286 \expandafter\cst@tempa\expandafter
287 {\expandafter\renewcommand\expandafter*\expandafter%
288 {\cst@refsec}{\cst@refname{#2}{#3}{#4}}}}

```

To make sure the changes are applied *now*, the current language is re-selected.

```

289 \expandafter\selectlanguage\expandafter{\language@name}
290 {}{}{}

```

Default names are loaded for a range of languages. Most journals stick to the same words, with only the order changing.

```

291 \cst@setrefname{english}{References}{and}{Notes}
292 \cst@setrefname{UKenglish}{References}{and}{Notes}
293 \cst@setrefname{ngerman}{Literatur}{und}{Notizen}
294 \cst@setrefname{german}{Literatur}{und}{Notizen}
295 \cst@setrefname{french}{R\'{e}f\'{e}rences}{et}{Notes}
296 \cst@setrefname{frenchb}{R\'{e}f\'{e}rences}{et}{Notes}

```



## 8 Configuration files

Each journal style needs slightly differing commands to get the formatting just right. This is handled here, with each style in a separate file. There is not a lot happening in most of these files, as the information is by its nature quite repetitive.

### 8.1 RSC style

```
297 \ProvidesFile{rsc.jdf}
298 [2008/03/21 v1.2a RSC journal style]
299 \AtBeginDocument{%
300   \renewcommand{\figurename}{Fig.}}
301 \captionsetup{labelsep=quad,labelfont=bf}
302 \setkeys{cst}{opt}{
303   radprefix=true,
304   rademph=true,
305   radsuper=false,
306   radhyphen=true,
307   abbrevemph=true,
308   abbrcomma=false}
```

### 8.2 *Angew. Chem.* style

```
309 \ProvidesFile{angew.def}
310 [2008/03/21 v1.2a Angew. Chem. journal style]
311 \captionsetup{labelsep=period,labelfont={bf,it},font=sf,singlelinecheck=off}
312 \captionsetup[table]{labelsep=colon}
313 \setkeys{cst}{opt}{
314   radprefix=true,
315   rademph=true,
316   radsuper=false,
317   radhyphen=false,
318   abbrevemph=false,
319   abbrcomma=false}
```

### 8.3 *J. Organomet. Chem.* style

```
320 \ProvidesFile{jomc.def}
321 [2008/03/21 v1.2a J. Organomet. Chem. journal style]
322 \captionsetup{labelsep=period}
323 \captionsetup[table]{labelsep=newline,singlelinecheck=off}
324 \AtBeginDocument{%
325   \renewcommand{\figurename}{Fig.}}
326 \setkeys{cst}{opt}{
327   radprefix=true,
328   rademph=true,
329   radsuper=true,
330   radhyphen=false,
331   abbrevemph=false,
332   abbrcomma=false,
333   notesbefore=true}
```

### 8.4 *Tetrahedron Lett.* style

```

334 \ProvidesFile{tetleTT.def}
335 [2008/03/21 v1.2a Tetrahedron Lett. journal style]
336 \captionsetup{labelsep=period,singlelinecheck=off,labelfont=bf}
337 \setkeys{cst}{opt}{
338   radprefix=true,
339   rademph=true,
340   radsuper=false,
341   radhyphen=true,
342   abbrevemph=false,
343   abbrcomma=true,
344   notesbefore=true}

```

## 8.5 *J. Am. Chem. Soc.* style

```

345 \ProvidesFile{jacs.jdf}
346 [2008/03/21 v1.2a J. Am. Chem. Soc. journal style]
347 \DeclareCaptionLabelSeparator{perquad}{.\quad}
348 \captionsetup{labelfont={bf,it,sf},textfont=sf,labelsep=perquad}
349 \captionsetup[figure]{textfont=rm}
350 \captionsetup{singlelinecheck=off}
351 \setkeys{cst}{opt}{
352   radprefix=true,
353   rademph=true,
354   radsuper=true,
355   radhyphen=false,
356   abbrevemph=false,
357   abbrcomma=true,
358   notesbefore=true}
359 \floatstyle{plaintop}
360 \restylefloat{scheme}
361 \floatstyle{plain}
362 \cst@notesbeforefalse

```

## 8.6 *Inorg. Chem.* style

Almost exactly the same as for *J. Am. Chem. Soc.*, so most of the work is left to `jacs.jdf`.

```

363 \ProvidesFile{ic.jdf}
364 [2008/03/21 v1.2a Inorg. Chem. journal style]
365 \input{jacs.jdf}
366 \captionsetup{textfont=rm}

```

## 8.7 *J. Phys. Chem.* style

```

367 \ProvidesFile{jpc.jdf}
368 [2008/03/21 v1.2a J. Phys. Chem. journal style]
369 \DeclareCaptionFormat{labelcaps}{\MakeUppercase{#1}#2#3}
370 \captionsetup{font=bf,labelsep=colon,format=labelcaps}
371 \captionsetup[figure]{format=plain,textfont=md,labelsep=period}
372 \captionsetup{singlelinecheck=off}
373 \setkeys{cst}{opt}{
374   radprefix=true,
375   rademph=true,
376   radsuper=false,
377   radhyphen=false,

```

```

378  abbrevemph=false,
379  abbrcomma=true,
380  notesbefore=true}
381 \floatstyle{plaintop}
382 \restylefloat{scheme}
383 \floatstyle{plain}
384 \cst@notesbeforefalse

```

## 8.8 *Org. Lett.* style

```

385 \ProvidesFile{orglett.jdf}
386 [2008/03/21 v1.2a Org. Lett. journal style]
387 \RequirePackage{xcolor}

```

`\OrgLettColour` To allow the user to control the colour of the “bars” in this style, an additional macro is provided. The purple colour looks about right for matching the printed journal, but if anyone has a better suggestion please let the package author know.

```

388 \newcommand*{\OrgLettColour}{purple}

```

`\fs@orglett` In order to get the distinctive coloured bars used by *Org. Lett.*, a new style for floats is needed. This is based on the `ruled` style from the `float` package.

```

389 \newcommand*{\fs@orglett}{%
390   \def\@fs@cfont{\bfseries}%
391   \let\@fs@capt\floatc@ruled
392   \def\@fs@pre{\begingroup\color{\OrgLettColour}
393     \hrule height12pt depth0pt \kern2pt\endgroup}%

```

This is the same `\@fs@mid` as is used in the `float` package for `plaintop` floats.

```

394   \def\@fs@mid{\vspace\belowcaptionskip\relax}%
395   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
396     \hrule height1.5pt depth0pt\endgroup}%
397   \let\@fs@iftopcapt\iftrue}

```

`\fs@orglettfig` Figures need to be slightly different, so an almost identical command is needed.

```

398 \newcommand*{\fs@orglettfig}{%
399   \def\@fs@cfont{\bfseries}%
400   \let\@fs@capt\floatc@plain
401   \def\@fs@pre{\begingroup\color{\OrgLettColour}%
402     \hrule height12pt depth0pt \kern2pt\endgroup}%
403   \def\@fs@mid{\vspace\abovecaptionskip\relax}%
404   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
405     \hrule height1.5pt depth0pt\endgroup}%
406   \let\@fs@iftopcapt\iffalse}

```

The new style is now applied. Users can change back to normal floats by changing back to the `plain` style.

```

407 \floatstyle{orglettfig}
408 \restylefloat{figure}
409 \floatstyle{orglett}
410 \restylefloat{scheme}
411 \restylefloat{table}

```

The more usual style commands now occur.

```

412 \DeclareCaptionLabelSeparator{perquad}{.\quad}
413 \captionsetup{labelfont=bf,labelsep=perquad}

```

```

414 \setkeys[cst]{opt}{
415   radprefix=true,
416   rademph=true,
417   radsuper=true,
418   radhyphen=false,
419   abbrevemph=false,
420   abbrcomma=true,
421   notesbefore=true}
422 \cst@notesbeforefalse

```

## 8.9 The empty style — none

To allow the user to load the extra macros provided here without any style changes, a “do nothing” style is provided. It simply makes sure that very little changes compared to the  $\text{\LaTeX}$  kernel. This requires undoing the defaults provided above. For commands where a default is needed (e.g. the `\latin` command) the style of the RSC is followed. As the float package has been loaded, notice that captions will be placed below floats even if the `\caption` command appears above the contents of the floating environment.

```

423 \ProvidesFile{none.jdf}
424 [2008/03/21 v1.2a Empty journal style]
425 \floatstyle{plain}
426 \restylefloat{table}
427 \labelformat{figure}{#1}
428 \labelformat{table}{#1}
429 \floatplacement{table}{tbp}
430 \floatplacement{figure}{tbp}
431 \AtBeginDocument{
432   \floatplacement{scheme}{tbp}
433   \labelformat{scheme}{#1}}

```

## 9 Change History

v1.0	General: Initial release of package . . . 1	Added siunits option . . . . . 1	Added xspace option . . . . . 1
v1.0a	General: Added <i>J. Phys. Chem.</i> style 18	Fixed (another) error in alkyl formatting . . . . . 17	Fixed error with spacing after <i>e.g.</i> and <i>i.e.</i> . . . . . 1
	Added <i>Org. Lett.</i> style . . . . . 19	License changed from GPL to LPPL 1	
	Fixed incorrect Latin formatting for <i>angew</i> option . . . . . 17	Phrases modified to better avoid clash with user’s own commands . . . . . 1	
	No longer load <i>fixltx2e</i> package . . . . . 8		
v1.0b	General: Added <i>microtype</i> and <i>notes2bib</i> to template . . . . . 1	<code>\cst@add@punct</code> : New macro . . . 14	<code>\cst@add@punct@</code> : New macro . . . 14
	Fixed formatting of alkyls . . . . 17	<code>\cst@addpunct</code> : New macro . . . 14	<code>\cst@latin</code> : New macro . . . . . 13
	Sorted a problem with <i>babel</i> and figure name format . . . . 17	<code>\cst@varnothing</code> : New macro . . . 11	<code>\eg</code> : Adds comma for ACS journals 13
v1.1	General: Added <i>phrases</i> option . . . 1	<code>\ie</code> : Adds comma for ACS journals 13	

v1.1a	General: Fixed error in documenta- tion compilation under L <sup>A</sup> T <sub>E</sub> X ...	1	Fixed problem with cross- referencing when babel is not loaded .....	12
v1.1b	General: Load caption even when journal style does not exist ...	14	v1.1h	General: Added tetlett style ..
v1.1c	General: Added Slunits option ....	1	v1.2	General: Added xkeyval interface ..
	Added frenchb alias for french .....	16		Re-write of reference-section re- naming code .....
	Added german alias for ngerman .....	16		Re-write of units section .....
	Documentation improved .....	1		Removed obsolete ch and chapter options .....
	Fixed problems with capitalisa- tion of Slunits .....	1		Various internal tweaks .....
	\standardstate: Altered angle of rotation for better appearance .	11		\cst@add@punct@: Internal tweaks .....
	Definition only occurs if user does not have own version .....	11		\cst@commaabbr: New macro ..
v1.1d	General: Require caption in all cases	8		\cst@commapunct: Name change from \cst@punct .....
v1.1e	General: Fixed packaging problem .	1		\cst@latin: Change of definition to allow better working with run-time options .....
v1.1f	\cst@addpunct: No longer used when comma not required ...	14		\cst@nocommaabbr: New macro
	\cst@eg: New macro .....	13		\cst@radical: Re-write to allow definition to change at run-time
	\cst@etal: New macro .....	13		\cst@ref@name: New macro ...
	\cst@etc: New macro .....	13		\cst@ref@name@: New macro ..
	\cst@ie: New macro .....	13		\cst@refname: New macro ....
	\eg: Definition comma-mode de- pendent .....	13		\cst@refsec: New macro .....
	Fixed problem with full stop spac- ing .....	13		\cst@setrefname: New macro .
	\etal: Fixed problem with full stop spacing .....	13		\cst@tempa: New macro .....
	\ie: Definition comma-mode de- pendent .....	13		\cst@tempb: New macro .....
	Fixed problem with full stop spac- ing .....	13		\cst@xspace: New definition ..
v1.1g	General: Added jomc style .....	17		\eg: Dynamic definition for comma mode .....
				\ie: Dynamic definition for comma mode .....
				\latinemphoff: New definition to work with xkeyval system ..
				\latinemphon: New definition to work with xkeyval system ....
			v1.2a	General: Fixed error with restyling scheme floats, caused by change in chemscheme .....

## 10 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>Symbols</b>	<b>A</b>	abbremph (option) ....	5
\' .....	295, 296	abbrcomma (option) ...	5

<b>C</b>		<b>L</b>	
\c@bibnote .....	<u>240</u>	\latin	<u>5</u> , <u>164</u> , <u>176</u> , <u>193–196</u>
chemcompounds (op-		\latinemphoff	<u>5</u> , <u>164</u>
tion) .....	<u>3</u>	\latinemphon	<u>5</u> , <u>164</u>
\cmc .....	<u>4</u> , <u>92</u>	<b>M</b>	
\cst@add@punct ..		\mmHg .....	<u>4</u> , <u>94</u>
. 205, 206, 210, 213		\mol .....	<u>94</u>
\cst@add@punct@ .		\Molar .....	<u>4</u> , <u>94</u>
..... 207, 208		\molar .....	<u>4</u> , <u>94</u>
\cst@addpunct 199, 204		<b>N</b>	
\cst@autonotesfalse		\nBu .....	<u>4</u> , <u>141</u>
..... 34		nonotes (option) .....	<u>3</u>
\cst@autonotesttrue		nophrases (option) ...	<u>5</u>
..... 38		notes (option) .....	<u>3</u>
\cst@boolkey <u>17</u> , 19–28		notesbefore (option) .	<u>3</u>
\cst@commaabbr ..		\nPr .....	<u>4</u> , <u>141</u>
..... 181, 188, <u>197</u>		<b>O</b>	
\cst@commapunct .		options:	
..... 215, <u>219</u>		abbrcomma .....	<u>5</u>
\cst@disablekey .. <u>73</u>		abbremph .....	<u>5</u>
\cst@eg .....	<u>193</u>	chemcompounds ..	<u>3</u>
\cst@emph <u>116</u> , 132, 134		journal .....	<u>2</u> , <u>3</u>
\cst@etal .... 178, <u>193</u>		nonotes .....	<u>3</u>
\cst@etc ..... 174, <u>193</u>		nophrases .....	<u>5</u>
\cst@hyphen <u>116</u> , 132, 134		notes .....	<u>3</u>
\cst@ie .....	<u>193</u>	notesbefore ....	<u>3</u>
\cst@journal ....		rademph .....	<u>4</u>
.. <u>52</u> , 220, 222, 224		radhyphen .....	<u>4</u>
\cst@language . <u>52</u> , 272		radprefix .....	<u>4</u>
\cst@latin .....	<u>164</u>	radsuper .....	<u>4</u>
\cst@nocommaabbr		xspace .....	<u>4</u>
..... 183, 190, <u>197</u>		\OrgLettColour ..	<u>388</u>
\cst@notesbeforefalse		<b>R</b>	
..... 362, 384, 422		rademph (option) .....	<u>4</u>
\cst@notesfalse .. 33		radhyphen (option) ...	<u>4</u>
\cst@notesttrue 37, 42		radprefix (option) ...	<u>4</u>
\cst@radical ....		radsuper (option) ....	<u>4</u>
..... <u>113</u> , 141–146		<b>S</b>	
\cst@ref@name ... <u>245</u>		\sBu .....	<u>4</u> , <u>141</u>
\cst@ref@name@ .. <u>245</u>		\standardstate	<u>4</u> , <u>108</u>
\cst@refname ....		<b>T</b>	
. 245, 275, 282, 288		\tBu .....	<u>4</u> , <u>141</u>
\cst@refsec .....		<b>X</b>	
. 231, 274, 281, 288		xspace (option) .....	<u>4</u>
\cst@setrefname .		<b>J</b>	
..... 268, 291–296		journal (option) ...	<u>2</u> , <u>3</u>
\cst@super <u>116</u> , 132, 134		<b>E</b>	
\cst@tempa .....		\eg .....	<u>5</u> , <u>171</u>
<u>15</u> , <u>29</u> , 207, <u>208</u> ,		\etal .....	<u>5</u> , <u>171</u>
271, 272, <u>277</u> ,		\etc .....	<u>5</u> , <u>171</u>
279, 280, 284, 286		<b>F</b>	
\cst@tempb . <u>15</u> , <u>29</u> , <u>208</u>		\fs@orglett ....	<u>389</u>
\cst@varnothing . <u>108</u>		\fs@orglettfig ..	<u>398</u>
\cst@xspace .....		<b>H</b>	
.. 136, <u>137</u> , 174,		\Hz .....	<u>4</u> , <u>92</u>
176, 178, 203, 219		<b>I</b>	
\cstsetup .....		\iBu .....	<u>4</u> , <u>141</u>
. 3, 83, 168, 169, 236		\ie .....	<u>5</u> , <u>171</u>
\cubiccentimeter		\ifcst@abbrcomma	
..... 92, <u>94</u>		..... 19, 180, 187	
<b>E</b>		\ifcst@abbremph <u>19</u> , 165	
\eg .....	<u>5</u> , <u>171</u>	\ifcst@autonotes	
\etal .....	<u>5</u> , <u>171</u>	..... 29, 240, 247	
\etc .....	<u>5</u> , <u>171</u>	\ifcst@chemcompounds	
<b>F</b>		..... 19, 69	
\fs@orglett ....	<u>389</u>	\ifcst@notes .. 29, 246	
\fs@orglettfig ..	<u>398</u>	\ifcst@notesbefore	
<b>H</b>		..... 19, 262	
\Hz .....	<u>4</u> , <u>92</u>	\ifcst@phrases <u>19</u> , 171	
<b>I</b>		\ifcst@rademph <u>19</u> , 126	
\iBu .....	<u>4</u> , <u>141</u>	\ifcst@radhyphen	
\ie .....	<u>5</u> , <u>171</u>	..... 19, 116	
\ifcst@abbrcomma		\ifcst@radprefix	
..... 19, 180, 187		..... 19, 131	
\ifcst@abbremph <u>19</u> , 165		\ifcst@radsuper <u>19</u> , 121	
\ifcst@autonotes		\ifcst@xspace . <u>19</u> , 138	
..... 29, 240, 247		\invacuo .....	<u>5</u> , <u>171</u>
\ifcst@chemcompounds		\iPr .....	<u>4</u> , <u>141</u>
..... 19, 69		<b>J</b>	
\ifcst@notes .. 29, 246		journal (option) ...	<u>2</u> , <u>3</u>
\ifcst@notesbefore		<b>X</b>	
..... 19, 262		xspace (option) .....	<u>4</u>
\ifcst@phrases <u>19</u> , 171		<b>T</b>	
\ifcst@rademph <u>19</u> , 126		<b>E</b>	
\ifcst@radhyphen		\eg .....	<u>5</u> , <u>171</u>
..... 19, 116		\etal .....	<u>5</u> , <u>171</u>
\ifcst@radprefix		\etc .....	<u>5</u> , <u>171</u>
..... 19, 131		<b>F</b>	
\ifcst@radsuper <u>19</u> , 121		\fs@orglett ....	<u>389</u>
\ifcst@xspace . <u>19</u> , 138		\fs@orglettfig ..	<u>398</u>
\invacuo .....	<u>5</u> , <u>171</u>	<b>H</b>	
\iPr .....	<u>4</u> , <u>141</u>	\Hz .....	<u>4</u> , <u>92</u>
<b>J</b>		<b>I</b>	
journal (option) ...	<u>2</u> , <u>3</u>	\iBu .....	<u>4</u> , <u>141</u>
<b>X</b>		\ie .....	<u>5</u> , <u>171</u>
xspace (option) .....	<u>4</u>	\ifcst@abbrcomma	