

chemstyle — Writing chemistry with style*

Joseph Wright[†]

Released 2008/07/23

Abstract

The chemstyle package provides a “one-stop shop” for setting up formatting of L^AT_EX 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

1	Introduction	2	7	The package code	8
			7.1	Setup code	8
			7.2	Option handling	8
			7.3	Extra units	10
2	Style options for chemstyle	2	7.4	Standard state symbol . . .	11
2.1	Creating styles	3	7.5	Alkyl radicals	11
2.2	Package options	3	7.6	Float formatting	12
3	Naming of the references section	3	7.7	Cross-references	12
			7.8	Latin phrases	12
			7.9	Loading the style definition	14
4	Additional macros	4	7.10	Handling reference section naming	14
4.1	Additional units	4	8	Configuration files	16
4.2	The standard state symbol	4	8.1	RSC style	16
4.3	Alkyl radicals	4	8.2	<i>Angew. Chem.</i> style . . .	17
4.4	Latin phrases	5	8.3	<i>J. Organomet. Chem.</i> style . . .	17
5	Additional information	6	8.4	<i>Tetrahedron Lett.</i> style . .	17
5.1	Advice for users of the rsc package	6	8.5	<i>J. Am. Chem. Soc.</i> style . .	18
5.2	Interactions with other packages	6	8.6	<i>Inorg. Chem.</i> style	18
5.3	Captions above floats . . .	6	8.7	<i>J. Phys. Chem.</i> style . . .	18
			8.8	<i>Org. Lett.</i> style	18
			8.9	The empty style — none	20
6	A template for chemical articles	7	9	Change History	20
			10	Index	21

*This file describes version v1.3b, last revised 2008/07/23.

[†]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^AT_EX 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 \TeX .¹ The definition files included in the package should provide a guide to the basic options available for producing new styles. Arbitrary \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>floatrow</code>	<code>chemstyle</code> loads the <code>chemscheme</code> package, and therefore will pass through the <code>chemcompounds</code> option. It can also request floats are created either by <code>float</code> or <code>floatrow</code> ; this is controlled by the <code>floatrow</code> option. Setting <code>floatrow=false</code> means that <code>float</code> is used to generate floats; the default value is <code>true</code> .

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> <code>nonotes</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
--	---

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

if no notes are given the section title remains as “References”. The `nonotes` option is equivalent to `notes=false`.

`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 of
<code>\mmHg</code>	units, but do not include a small number of unit macros useful to chemists. In
<code>\molar</code>	contrast, the <code>siunitx</code> package provides all of these using the <code>synchem</code> module. If
<code>\Molar</code>	<code>Slunits</code> or <code>unitsdef</code> are loaded, <code>chemstyle</code> provides the missing units. On the other
<code>\cmc</code>	hand, if no unit package is loaded, <code>siunitx</code> is loaded and provides the units itself.

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

4.2 The standard state symbol

`\standardstate` Related to the above, but not exactly a unit is the `\standardstate` command.² This generates the tricky `\standardstate` symbol. The symbol will adapt to local sizing.

the standard conditions are indicated:	the standard conditions are indicated: <code>\standardstate\</code>
\ominus	Common but not correct:
Common but not correct: ΔG_f^\ominus or ΔG_f^\ominus	<code>\$\Delta G_\mathrm{f}^\mathrm{\standardstate}\$</code> or <code>\$\Delta G_\mathrm{f}^\mathrm{\standardstate}\$ \</code>
Better: $\Delta_f G^\ominus$	Better:
Sizing: $\int_{T^\ominus}^{T_{\text{out}}}$	<code>\$\Delta_\mathrm{f} G^\mathrm{\standardstate}\$ \</code>
	Sizing:
	<code>\$\int_{T^\mathrm{\standardstate}}^{T_{\text{out}}}\$</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 <i>n</i> -Pr, <i>i</i> -Pr or <i>n</i> -Bu without affecting the selectivity.	The alkyl group could be <code>\nPr</code> , <code>\iPr\</code> or <code>\nBu</code> without affecting the selectivity.
--	---

`xspace` When `chemstyle` is loaded using the `xspace` option, the `xspace` package is

²The `\standardstate` macro is only defined if the user does not have their own version.

automatically used to add space after the command names, so that `\iPr` group will result in “*i*-Pr group” being typeset.

The appearance of these radical abbreviations is controlled by the package 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.

<code>radhyphen</code>	<code>\cstsetup{radhyphen=false,radsuper=true}\\</code>
<code>rademph</code>	<code>\iPr \tBu</code>
<code>radsuper</code>	<code>\cstsetup{radhyphen=true,radsuper=false,%</code>
<code>radprefix</code>	<code>radprefix=false,rademph=false}\\</code>
<code>iPr</code>	<code>\iBu \sBu</code>
<code>tBu</code>	

4.4 Latin phrases

<code>\latin</code>	The various Latin phrases commonly used in chemistry are made available as the obvious commands. By altering the definition of <code>\latin</code> , this allows ready switching from italic to Roman typesetting. Notice that <code>\etc</code> , <code>\ie</code> and <code>\eg</code> 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.
<code>\etc</code>	
<code>\eg</code>	
<code>\ie</code>	
<code>\etal</code>	
<code>\invacuo</code>	
<code>abbremph</code>	

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

<code>et al.</code>	or	<code>\invacuo</code>	plus	<code>\latin{some text}\\</code>
<code>i.e.</code>	equals	<code>e.g.</code>	perhaps	<code>\etc</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.

<code>e.g.</code>	this	<code>\eg this\\</code>
<code>i.e.</code>	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

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

`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 `\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.

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

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^AT_EX 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 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.³ 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]
  The float contents
  \caption{A caption below the float contents in the source}
\end{table}
\floatsetup[table]{style=plain} % When using floatrow
% \floatstyle{plain}
% \restylefloat{table} When using float
\begin{table}[ht]
  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

³Normally this is a good thing.

The second float contents, which should be above the caption

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

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.

```
1 \documentclass[fontsize=10pt,paper=a4,english,UKenglish]
2 {scrreprt}
3 \usepackage{geometry,upgreek,booktabs,babel}
4 \usepackage[journal=rsc,xspace=true]{chemstyle}
5 \usepackage[version=3]{mhchem}
6 \usepackage[footnotes]{notes2bib}
7 \usepackage[final]{microtype}
8 \usepackage[final,inactive]{pst-pdf}
9 \usepackage[colorlinks]{hyperref}
10 \begin{document}
11   Document contents go here
12 \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 \LaTeX system. `booktabs` gives much nicer looking tables than the $\text{\LaTeX}_{2\epsilon}$ default. `upgreek` provides non-italic lowercase Greek letters, which should be used for things such as bond descriptions.

π -bond σ^* -orbital

`\uppi` π -bond `\upsigma^*` σ^* -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_2SO_4 .⁴

The `microtype` package improves formatting when used with the `pdf \TeX` 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

⁴There is a slight cheat here, as this document uses lower-case numerals in the text. The example is written as `\ce{H2SO4}`.

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.

7 The package code

7.1 Setup code

`\cst@id` 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.

```
13 \NeedsTeXFormat{LaTeX2e}
14 \def\cst@id$#1: #2.#3 #4 #5-#6-#7 #8 #9${%
15     \def\cst@ver{#5/#6/#7\space v1.3b\space}}
16 \cst@id $Id: chemstyle.dtx 46 2008-07-23 11:29:06Z joseph $
17 \ProvidesPackage{chemstyle}
18 [\cst@ver Writing chemistry with style]
```

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

```
19 \RequirePackage{xkeyval}[2005/05/07]
20 \RequirePackage{graphicx,varioref,caption,xspace}
```

`\cst@tempa` A couple of temporary private macros.

```
\cst@tempb 21 \newcommand*\cst@tempa{}
22 \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.

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

`\ifcst@chemcompounds` The Boolean keys are now defined.

```
\ifcst@floatrow 25 \cst@boolkey{chemcompounds}
\ifcst@xspace 26 \cst@boolkey{floatrow}
\ifcst@xspace 27 \cst@boolkey{xspace}
\ifcst@phrases 28 \cst@boolkey{phrases}
\ifcst@radprefix 29 \cst@boolkey{radprefix}
\ifcst@rademph 30 \cst@boolkey{rademph}
\ifcst@radsuper 31 \cst@boolkey{radsuper}
\ifcst@radhyphen 32 \cst@boolkey{radhyphen}
\ifcst@abbremph 33 \cst@boolkey{abbremph}
\ifcst@abbrcomma 34 \cst@boolkey{abbrcomma}
\ifcst@notesbefore 35 \cst@boolkey{notesbefore}
```

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

```
\ifcst@autonotes 36 \newif\ifcst@notes
\cst@tempa 37 \newif\ifcst@autonotes
\cst@tempb
```



```

38 \define@choicekey*[cst]{opt}{notes}[\cst@tempa]
39   {true,false,auto}[true]
40   {\cst@notesfalse
41     \cst@autonotesfalse
42     \edef\cst@tempb{auto}
43     \ifx\cst@tempa\cst@tempb
44       \cst@notesttrue
45       \cst@autonotesttrue
46     \fi
47     \edef\cst@tempb{true}
48     \ifx\cst@tempa\cst@tempb
49       \cst@notesttrue
50     \fi}

```

Some old options are no longer needed at all.

```

51 \DeclareOptionX[cst]<opt>{siunits}
52   {\PackageInfo{chemstyle}{Option siunits no longer required}}
53 \DeclareOptionX[cst]<opt>{SIunits}
54   {\PackageInfo{chemstyle}{Option siunits no longer required}}
55 \DeclareOptionX[cst]<opt>{nonotes}
56   {\ExecuteOptionsX[cst]<opt>{notes=false}}
57 \DeclareOptionX[cst]<opt>{nophrases}
58   {\ExecuteOptionsX[cst]<opt>{phrases=false}}

```

\cst@journal The journal option can be given with or without journal=.

\cst@language 59 \define@cmdkeys[cst]{opt}{cst@}{journal,language}
60 \DeclareOptionX*{\edef\cst@journal{\CurrentOption}}

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

```

61 \setkeys[cst]{opt}{
62   notes=true,
63   phrases=true,
64   chemcompounds=true,
65   xspace=true,
66   floatrow=true,
67   journal=none,
68   radprefix=true,
69   rademph=true,
70   radsuper=false,
71   radhyphen=true,
72   abbrevemph=true,
73   abbrcomma=false,
74   language=english,
75   notesbefore=true}
76 \ProcessOptionsX[cst]<opt>

```

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

```

77 \ifcst@chemcompounds
78   \PassOptionsToPackage{chemcompounds}{chemscheme}
79 \fi
80 \ifcst@floatrow
81   \RequirePackage[floats=floatrow]{chemscheme}
82 \else
83   \RequirePackage[floats=float]{chemscheme}
84 \fi

```

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

```

85 \newcommand*{\cst@disablekey}[1]{
86   \define@key[cst]{opt}{#1}{\PackageWarning{chemstyle}
87     {Option '#1' only valid when loading package}}
88 \AtBeginDocument{
89   \cst@disablekey{floatrow}
90   \cst@disablekey{journal}
91   \cst@disablekey{phrases}
92   \cst@disablekey{nophrases}
93   \cst@disablekey{notes}
94   \cst@disablekey{nonotes}
95   \cst@disablekey{chemcompounds}}

```

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

```

96 \newcommand*{\cstsetup}[1]{%
97   \iffalse{\fi\ifnum0='}\fi
98   \setkeys[cst]{opt}{#1}%
99   \ifnum0='{ \fi\iffalse}\fi}

```

7.3 Extra units

`\cmc` The `siunitx` package provides all of the units desired here. So if it is loaded, the
`\Hz` module `synchem` is all that needs to be loaded. Otherwise, a series of tests are
`\cubiccentimeter` needed for `SIunits` and `unitsdef`. If either of these are loaded, they are used; if not,
`\Molar` then `siunitx` is loaded in any case.

```

\molar 100 \@ifpackageloaded{siunitx}
\mmHg 101   {\requiresiconfigs{synchem}}
\mol 102   {\@ifpackageloaded{SIunits}
103     {\newcommand*\cubiccentimeter}{\centi\metre\cubed}
104     \newcommand*\Molar{\textsc{m}}
105     \newcommand*\molar{\mole\usl\deci\metre\rpcubed}
106     \newcommand*\mmHg{\milli\metre\Hg}
107     \newcommand*\mol{\mole}
108     \newcommand*\cmc{\cubiccentimeter}
109     \newcommand*\Hz{\hertz}}
110   {\@ifpackageloaded{unitsdef}
111     {\newunit{\cubiccentimeter}{%
112       \cm\unitsuperscript{3}}
113     \newunit{\Molar}{\textsc{m}}
114     \newunit{\molar}{%
115       \mole\unitsep\dm\unitsuperscript{--3}}
116     \newunit{\mmHg}{mmHg}
117     \newunit{\mol}{\mole}
118     \newcommand*\cmc{\cubiccentimeter}
119     \newcommand*\Hz{\hertz}}
120   {\RequirePackage[alsoload=synchem]{siunitx}}

```

7.4 Standard state symbol

`\standardstate` The “standard state” symbol is handy. This definition is adapted from one posted several times to `comp.text.tex` by Donald Arseneau.

```
\cst@sstate
\cst@s@state 121 \providecommand*\standardstate{%
122   {\ensuremath{\protect\cst@sstate}}}
123 \newcommand*\cst@sstate{\mathpalette\cst@s@state\circ}
124 \newcommand*\cst@s@state[2]{%
125   \ooalign{\hfil$#1-$\hfil\cr\hfil$#1#2$\hfil\cr}}
```

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.

```
126 \newcommand*\cst@radical[2]{%
127   \expandafter\DeclareRobustCommand\expandafter*\expandafter{%
128     \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 129   \ifcst@radhyphen
130     \def\cst@hyphen{-}%
131   \else
132     \let\cst@hyphen\relax
133   \fi
134   \ifcst@radsuper
135     \let\cst@super\textsuperscript
136   \else
137     \let\cst@super\relax
138   \fi
139   \ifcst@rademph
140     \let\cst@emph\emph
141   \else
142     \let\cst@emph\relax
143   \fi
```

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

```
144   \ifcst@radprefix
145     \cst@super{\cst@emph{#1}}\cst@hyphen#2%
146   \else
147     #2\cst@hyphen\cst@super{\cst@emph{#1}}%
148   \fi
149   \cst@xspace}}
```

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

```
150 \newcommand*\cst@xspace{%
151   \ifcst@xspace
152     \expandafter\xspace%
153   \fi}
```

`\nPr` With a mechanism in place, the abbreviations are declared. The format of the resulting output will depend upon the configuration file used.

`\iPr`
`\nBu`
`\iBu`
`\sBu`
`\tBu`

```

154 \cst@radical{n}{Pr}%
155 \cst@radical{i}{Pr}%
156 \cst@radical{n}{Bu}%
157 \cst@radical{i}{Bu}%
158 \cst@radical{s}{Bu}%
159 \cst@radical{t}{Bu}%

```

7.6 Float formatting

The next step is to format the floats correctly. 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.

```

160 \floatstyle{plaintop}
161 \restylefloat{table}
162 \floatstyle{plain}
163 \restylefloat{figure}
164 \floatplacement{table}{htbp}
165 \floatplacement{figure}{htbp}

```

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

```

166 \AtBeginDocument{
167   \restylefloat{scheme}
168   \floatplacement{scheme}{htbp}}

```

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

```

169 \floatcontentscentre

```

7.7 Cross-references

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

```

170 \labelformat{figure}{\figurename~#1}
171 \labelformat{table}{\tablename~#1}
172 \AtBeginDocument{\labelformat{scheme}{\schemename~#1}}

```

7.8 Latin phrases

```

\latin
\cst@latin
\latinemphon
\latinemphoff

```

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.

```

173 \newcommand*{\cst@latin}{%
174   \ifcst@abbremph
175     \expandafter\emph%
176   \fi}
177 \newcommand*{\latinemphon}{\cstsetup{abbremph=true}}
178 \newcommand*{\latinemphoff}{\cstsetup{abbremph=false}}
179 \AtBeginDocument{\providecommand*\latin{\cst@latin}}

```

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

```

180 \ifcst@phrases
181   \AtBeginDocument{
182     \providecommand*\etc{
183       {\@ifnextchar.{\cst@etc}{\cst@etc.\cst@xspace}}
184     \providecommand*\invacuo{
185       {\latin{in vacuo}\cst@xspace}
186     \providecommand*\etal{
187       {\@ifnextchar.{\cst@etal}{\cst@etal.\cst@xspace}}
188     \providecommand*\eg{
189       {\ifcst@abbrcomma
190         \expandafter\cst@commaabbr%
191       \else
192         \expandafter\cst@nocommaabbr%
193       \fi
194       {eg}}
195     \providecommand*\ie{
196       {\ifcst@abbrcomma
197         \expandafter\cst@commaabbr%
198       \else
199         \expandafter\cst@nocommaabbr%
200       \fi
201       {ie}}}

```

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

```

202 \newcommand*\cst@etal{\latin{et~al}}
203 \newcommand*\cst@etc{\latin{etc}}
204 \newcommand*\cst@ie{\latin{i.e\spacefactor999\relax}}
205 \newcommand*\cst@eg{\latin{e.g\spacefactor999\relax}}

```

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

```

\cst@nocommaabbr 206 \newcommand*\cst@commaabbr[1]{%
207   \csname cst@#1\endcsname%
208   \cst@addpunct}
209 \newcommand*\cst@nocommaabbr[1]{%
210   \@ifnextchar.%
211   {\csname cst@#1\endcsname}
212   {\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@add@punct` in the first round of checking.

```

213 \newcommand*\cst@addpunct{%
214   \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.

```

215 \newcommand*{\cst@add@punct}[1]{%
216   \futurelet\cst@tempa\cst@add@punct@}

```

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

```

217 \newcommand*{\cst@add@punct@}{%
218   \ifx.\cst@tempa
219     \let\cst@tempb\cst@add@punct
220   \else
221     \ifx,\cst@tempa
222       \let\cst@tempb\cst@add@punct

```

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

```

223   \else
224     \let\cst@tempb\cst@commapunct
225   \fi
226 \fi
227 \cst@tempb}

```

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

```

228 \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`.

```

229 \InputIfFileExists{\cst@journal.jdf}
230 {\PackageInfo{chemstyle}
231   {Loaded \cst@journal.jdf}}
232 {\PackageWarning{chemstyle}
233   {Requested style '\cst@journal' does not exist\MessageBreak
234     Loading default style}}
235 \InputIfFileExists{none.jdf}{}
236 {\PackageError{chemstyle}{Failed to load none.jdf}
237   {The default journal definition file was not
238     found\MessageBreak This is a core part of
239     chemstyle\MessageBreak Something is wrong with
240     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.⁵ A second check is needed, in case the document class does not

⁵If `\@undefined` is ever fixed, this will no longer be needed. If $\epsilon\text{-}\TeX$ was required, the `\ifdefined` would be a choice here; however, $\epsilon\text{-}\TeX$ is not going to be required just for this.

define any reference section at all (for example, minimal does not).

```

241 \begingroup
242   \@ifundefined{chapter}
243     {\@ifundefined{refname}
244       {\PackageInfo{chemstyle}
245         {No bibliography name command found}
246         \cstsetup{notes=false}}
247       {\gdef\cst@refsec{\refname}}}}
248   {\gdef\cst@refsec{\bibname}}
249 \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.

```

250 \ifcst@autonotes
251   \AtBeginDocument{
252     \@ifpackageloaded{notes2bib}{}
253     {\newcounter{bibnote}}}
254 \fi

```

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

```

255 \newcommand*{\cst@refname}{%
256   \ifcst@notes
257     \ifcst@autonotes
258       \ifnum\the\value{bibnote}>\z@
259         \expandafter\expandafter\expandafter\expandafter
260         \expandafter\expandafter\expandafter\cst@ref@name
261       \else
262         \expandafter\expandafter\expandafter\expandafter
263         \expandafter\expandafter\expandafter\cst@ref@name@
264       \fi
265     \else
266       \expandafter\expandafter\expandafter\cst@ref@name
267     \fi
268   \else
269     \expandafter\cst@ref@name@%
270   \fi}
271 \newcommand*{\cst@ref@name}[3]{%
272   \ifcst@notesbefore
273     #3\space#2\space#1%
274   \else
275     #1\space#2\space#3%
276   \fi}
277 \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.

```

278 \@ifundefined{cst@refsec}
279   {\let\cst@setrefname\@gobblefour}

```

```

280   {\newcommand*\cst@setrefname}[4]{
281     \edef\cst@tempa{#1}%
282     \ifx\cst@tempa\cst@language
283       \expandafter\renewcommand\expandafter*\expandafter{%
284         \cst@refsec}
285       {\cst@refname{#2}{#3}{#4}}
286     \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.

```

287   \AtBeginDocument{
288     \ifpackageloaded{biblatex}
289     {\def\cst@tempa{\csgappto{bib@strings@#1}}
290     \expandafter\cst@tempa\expandafter{\expandafter
291       \renewcommand\expandafter*\expandafter{%
292         \cst@refsec}{%
293           \cst@refname{#2}{#3}{#4}}}}
294     {\ifpackageloaded{babel}
295     {\def\cst@tempa{\expandafter\addto\expandafter{%
296       \csname captions#1\endcsname}}
297     \expandafter\cst@tempa\expandafter
298     {\expandafter\renewcommand\expandafter*%
299       \expandafter{\cst@refsec}{%
300         \cst@refname{#2}{#3}{#4}}}}

```

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

```

301       \expandafter\selectlanguage\expandafter{%
302         \language#1}
303     {}}}}

```

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

```

304 \cst@setrefname{english}{References}{and}{Notes}
305 \cst@setrefname{UKenglish}{References}{and}{Notes}
306 \cst@setrefname{ngerman}{Literatur}{und}{Notizen}
307 \cst@setrefname{german}{Literatur}{und}{Notizen}
308 \cst@setrefname{french}{R\'{e}f\'{e}rences}{et}{Notes}
309 \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

```

310 \ProvidesFile{rsc.jdf}
311 [\cst@ver RSC journal style]
312 \AtBeginDocument{%

```



```

313 \renewcommand{\figurename}{Fig.}}
314 \captionsetup{labelsep=quad,labelfont=bf}
315 \setkeys[cst]{opt}{
316   radprefix=true,
317   rademph=true,
318   radsuper=false,
319   radhyphen=true,
320   abbrevemph=true,
321   abbrcomma=false}

```

8.2 *Angew. Chem. style*

```

322 \ProvidesFile{angew.def}
323 [\cst@ver Angew. Chem. journal style]
324 \captionsetup{labelsep=period,labelfont={bf,it},font=sf,
325   singlelinecheck=off}
326 \captionsetup[table]{labelsep=colon}
327 \setkeys[cst]{opt}{
328   radprefix=true,
329   rademph=true,
330   radsuper=false,
331   radhyphen=false,
332   abbrevemph=false,
333   abbrcomma=false}

```

8.3 *J. Organomet. Chem. style*

```

334 \ProvidesFile{jomc.def}
335 [\cst@ver J. Organomet. Chem. journal style]
336 \captionsetup{labelsep=period}
337 \captionsetup[table]{labelsep=newline,singlelinecheck=off}
338 \AtBeginDocument{%
339   \renewcommand{\figurename}{Fig.}}
340 \setkeys[cst]{opt}{
341   radprefix=true,
342   rademph=true,
343   radsuper=true,
344   radhyphen=false,
345   abbrevemph=false,
346   abbrcomma=false,
347   notesbefore=true}

```

8.4 *Tetrahedron Lett. style*

```

348 \ProvidesFile{tetlett.def}
349 [\cst@ver Tetrahedron Lett. journal style]
350 \captionsetup{labelsep=period,singlelinecheck=off,labelfont=bf}
351 \setkeys[cst]{opt}{
352   radprefix=true,
353   rademph=true,
354   radsuper=false,
355   radhyphen=true,
356   abbrevemph=false,
357   abbrcomma=true,
358   notesbefore=true}

```

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

```
359 \ProvidesFile{jacs.jdf}
360 [\cst@ver J. Am. Chem. Soc. journal style]
361 \DeclareCaptionLabelSeparator{perquad}{.\quad}
362 \captionsetup{labelfont={bf,it,sf},textfont=sf,
363   labelsep=perquad}
364 \captionsetup[figure]{textfont=rm}
365 \captionsetup{singlelinecheck=off}
366 \setkeys{cst}{opt}{
367   radprefix=true,
368   rademph=true,
369   radsuper=true,
370   radhyphen=false,
371   abbrevemph=false,
372   abbrcomma=true,
373   notesbefore=true}
374 \floatstyle{plaintop}
375 \restylefloat{scheme}
376 \floatstyle{plain}
377 \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.

```
378 \ProvidesFile{ic.jdf}
379 [\cst@ver Inorg. Chem. journal style]
380 \input {jacs.jdf}
381 \captionsetup{textfont=rm}
```

8.7 *J. Phys. Chem.* style

```
382 \ProvidesFile{jpc.jdf}
383 [\cst@ver J. Phys. Chem. journal style]
384 \DeclareCaptionFormat{labelcaps}{\MakeUppercase{#1}#2#3}
385 \captionsetup{font=bf,labelsep=colon,format=labelcaps}
386 \captionsetup[figure]{format=plain,textfont=md,labelsep=period}
387 \captionsetup{singlelinecheck=off}
388 \setkeys{cst}{opt}{
389   radprefix=true,
390   rademph=true,
391   radsuper=false,
392   radhyphen=false,
393   abbrevemph=false,
394   abbrcomma=true,
395   notesbefore=true}
396 \floatstyle{plaintop}
397 \restylefloat{scheme}
398 \floatstyle{plain}
399 \cst@notesbeforefalse
```

8.8 *Org. Lett.* style

```
400 \ProvidesFile{orglett.jdf}
```

```

401 [\cst@ver Org. Lett. journal style]
402 \RequirePackage{xcolor}

```

Currently, this style only works with float.

```

403 \@ifpackageloaded{floatrow}
404 {\PackageError{chemstyle}
405   {Org. Lett. style only works with float}
406   {Load chemstyle with the floatrow=false option
407    to\MessageBreak use the orglett style}}
408 {}

```

`\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.

```

409 \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.

```

410 \newcommand*{\fs@orglett}{%
411   \def\@fs@cfont{\bfseries}%
412   \let\@fs@capt\floatc@ruled
413   \def\@fs@pre{\begingroup\color{\OrgLettColour}
414     \hrule height12pt depth0pt \kern2pt\endgroup}%

```

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

```

415   \def\@fs@mid{\vspace\belowcaptionskip\relax}%
416   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
417     \hrule height1.5pt depth0pt\endgroup}%
418   \let\@fs@iftopcapt\iftrue}

```

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

```

419 \newcommand*{\fs@orglettfig}{%
420   \def\@fs@cfont{\bfseries}%
421   \let\@fs@capt\floatc@plain
422   \def\@fs@pre{\begingroup\color{\OrgLettColour}%
423     \hrule height12pt depth0pt \kern2pt\endgroup}%
424   \def\@fs@mid{\vspace\abovecaptionskip\relax}%
425   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
426     \hrule height1.5pt depth0pt\endgroup}%
427   \let\@fs@iftopcapt\iffalse}

```

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

```

428 \floatstyle{orglettfig}
429 \restylefloat{figure}
430 \floatstyle{orglett}
431 \restylefloat{scheme}
432 \restylefloat{table}

```

The more usual style commands now occur.

```

433 \DeclareCaptionLabelSeparator{perquad}{.\quad}
434 \captionsetup{labelfont=bf,labelsep=perquad}
435 \setkeys{cst}{opt}{
436   radprefix=true,
437   rademph=true,

```

```

438 radsuper=true,
439 radhyphen=false,
440 abbrevemph=false,
441 abbrcomma=true,
442 notesbefore=true}
443 \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 L^AT_EX 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.

```

444 \ProvidesFile{none.jdf}
445 [\cst@ver Empty journal style]
446 \floatstyle{plain}
447 \restylefloat{table}
448 \labelformat{figure}{#1}
449 \labelformat{table}{#1}
450 \floatplacement{table}{tbp}
451 \floatplacement{figure}{tbp}
452 \AtBeginDocument{
453   \floatplacement{scheme}{tbp}
454   \labelformat{scheme}{#1}}

```

9 Change History

v1.0	Fixed (another) error in alkyl formatting	16
General: Initial release of package		1
v1.0a	Fixed error with spacing after <i>e.g.</i> and <i>i.e.</i>	1
General: Added <i>J. Phys. Chem.</i> style	License changed from GPL to LPPL	1
Added <i>Org. Lett.</i> style	Phrases modified to better avoid clash with user’s own commands	1
Fixed incorrect Latin formatting for <i>angew</i> option		16
No longer load <i>fixltx2e</i> package		8
v1.0b	<code>\cst@add@punct</code> : New macro	13
General: Added <i>microtype</i> and <i>notes2bib</i> to template	<code>\cst@add@punct@</code> : New macro	13
Fixed formatting of alkyls	<code>\cst@addpunct</code> : New macro	13
Sorted a problem with <i>babel</i> and figure name format	<code>\cst@latin</code> : New macro	12
	<code>\eg</code> : Adds comma for ACS journals	12
	<code>\ie</code> : Adds comma for ACS journals	12
v1.1	v1.1a	
General: Added <i>phrases</i> option	General: Fixed error in documentation compilation under L ^A T _E X	1
Added <i>siunits</i> option	v1.1b	
Added <i>xspace</i> option	General: Load caption even when journal style does not exist	14

v1.1c	General: Added Slunits option	1	Various internal tweaks	1
	Added frenchb alias for		\cst@add@punct@: Internal	
	french	16	tweaks	13
	Added german alias for		\cst@commaabbr: New macro ..	13
	ngerman	16	\cst@commapunct: Name change	
	Documentation improved	1	from \cst@punct	14
	Fixed problems with capitalisa-		\cst@latin: Change of definition	
	tion of Slunits	1	to allow better working with	
			run-time options	12
v1.1d	General: Require caption in all cases	8	\cst@nocommaabbr: New macro	13
v1.1e	General: Fixed packaging problem .	1	\cst@radical: Re-write to allow	
			definition to change at run-time	10
v1.1f	\cst@addpunct: No longer used		\cst@ref@name: New macro ...	15
	when comma not required ...	13	\cst@ref@name@: New macro ..	15
	\cst@eg: New macro	13	\cst@refname: New macro	15
	\cst@etal: New macro	13	\cst@refsec: New macro	14
	\cst@etc: New macro	13	\cst@setrefname: New macro .	15
	\cst@ie: New macro	13	\cst@tempa: New macro	8
	\eg: Definition comma-mode de-		\cst@tempb: New macro	8
	pendent	12	\cst@xspace: New definition ..	11
	Fixed problem with full stop spa-		\eg: Dynamic definition for comma	
	cing	12	mode	12
	\etal: Fixed problem with full stop		\ie: Dynamic definition for comma	
	spacing	12	mode	12
	\ie: Definition comma-mode de-		\latinemphoff: New definition	
	pendent	12	to work with xkeyval system ..	12
	Fixed problem with full stop spa-		\latinemphon: New definition to	
	cing	12	work with xkeyval system	12
v1.1g	General: Added jomc style	17	v1.2a	General: Fixed error with resty-
	Fixed problem with cross-			ling scheme floats, caused by
	referencing when babel is not			change in chemscheme
	loaded	12	v1.2b	General: Change due to name
v1.1h	General: Added tetlett style ..	17		change of si package to siunitx .
v1.2	General: Added xkeyval interface ..	8	v1.2c	General: Altered call to siunitx func-
	Re-write of reference-section rena-			tion
	ming code	14	v1.3a	\standardstate: Completely
	Re-write of units section	10		new definition
	Removed obsolete ch and		v1.3b	General: Default to loading siunitx
	chapter options	1		10

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.

A	abbremph (option)	5
abbrcomma (option)		5

C		<code>\etc</code> 4 , 180
<code>\c@bibnote</code> 250	F	
<code>chemcompounds (option)</code> 3	<code>floatrow (option)</code> 3	
<code>\circ</code> 123	<code>\fs@orglett</code> 410	
<code>\cmc</code> 3 , 100	<code>\fs@orglettfig</code> 419	
<code>\cr</code> 125	H	
<code>\cst@add@punct</code> ... 214 , 215 , 219 , 222	<code>\hfil</code> 125	
<code>\cst@add@punct@</code> 216 , 217	<code>\Hz</code> 3 , 100	
<code>\cst@addpunct</code> 208 , 213	I	
<code>\cst@autonotesfalse</code> 41	<code>\iBu</code> 4 , 154	
<code>\cst@autonotestru</code> 45	<code>\ie</code> 4 , 180	
<code>\cst@boolkey</code> 23 , 25–35	<code>\ifcst@abbrcomma</code> 25 , 189 , 196	
<code>\cst@commaabbr</code> 190 , 197 , 206	<code>\ifcst@abbremph</code> 25 , 174	
<code>\cst@commapunct</code> 224 , 228	<code>\ifcst@autonotes</code> 36 , 250 , 257	
<code>\cst@disablekey</code> 85	<code>\ifcst@chemcompounds</code> 25 , 77	
<code>\cst@eg</code> 202	<code>\ifcst@floatrow</code> 25 , 80	
<code>\cst@emph</code> 129 , 145 , 147	<code>\ifcst@notes</code> 36 , 256	
<code>\cst@etal</code> 187 , 202	<code>\ifcst@notesbefore</code> 25 , 272	
<code>\cst@etc</code> 183 , 202	<code>\ifcst@phrases</code> 25 , 180	
<code>\cst@hyphen</code> 129 , 145 , 147	<code>\ifcst@rademph</code> 25 , 139	
<code>\cst@id</code> 13	<code>\ifcst@radhyphen</code> 25 , 129	
<code>\cst@ie</code> 202	<code>\ifcst@radprefix</code> 25 , 144	
<code>\cst@journal</code> 59 , 229 , 231 , 233	<code>\ifcst@radsuper</code> 25 , 134	
<code>\cst@language</code> 59 , 282	<code>\ifcst@xspace</code> 25 , 151	
<code>\cst@latin</code> 173	<code>\invacuo</code> 4 , 180	
<code>\cst@nocommaabbr</code> 192 , 199 , 206	<code>\iPr</code> 4 , 154	
<code>\cst@notesbeforefalse</code> 377 , 399 , 443	J	
<code>\cst@notesfalse</code> 40	<code>journal (option)</code> 2 , 3	
<code>\cst@notestru</code> 44 , 49	L	
<code>\cst@radical</code> 126 , 154–159	<code>\latin</code> 4 , 173 , 185 , 202–205	
<code>\cst@ref@name</code> 255	<code>\latinemphoff</code> 5 , 173	
<code>\cst@ref@name@</code> 255	<code>\latinemphon</code> 5 , 173	
<code>\cst@refname</code> 255 , 285 , 293 , 300	M	
<code>\cst@refsec</code> 241 , 284 , 292 , 299	<code>\mathpalette</code> 123	
<code>\cst@s@state</code> 121	<code>\mmHg</code> 3 , 100	
<code>\cst@setrefname</code> 278 , 304–309	<code>\mol</code> 100	
<code>\cst@sstate</code> 121	<code>\Molar</code> 3 , 100	
<code>\cst@super</code> 129 , 145 , 147	<code>\molar</code> 3 , 100	
<code>\cst@tempa</code> 21 , 36 , 216 , 217 , 281 , 282 , 287 , 289 , 290 , 295 , 297	N	
<code>\cst@tempb</code> 21 , 36 , 217	<code>\nBu</code> 4 , 154	
<code>\cst@ver</code> 13 , 311 , 323 , 335 , 349 , 360 , 379 , 383 , 401 , 445	<code>nonotes (option)</code> 3	
<code>\cst@xspace</code> 149 , 150 , 183 , 185 , 187 , 212 , 228	<code>nophrases (option)</code> 5	
<code>\cstsetup</code> 3 , 96 , 177 , 178 , 246	<code>notes (option)</code> 3	
<code>\cubiccentimeter</code> 100	<code>notesbefore (option)</code> 3	
E		<code>\nPr</code> 4 , 154
<code>\eg</code> 4 , 180		
<code>\etal</code> 4 , 180		

O		P	
\oalign	125	\protect	122
options:			
abbrcomma	5	R	
abbremph	5	rademph (option)	4
chemcompounds	3	radhyphen (option)	4
floatrow	3	radprefix (option)	4
journal	2, 3	radsuper (option)	4
nonotes	3	S	
nophrases	5	\sBu	4, 154
notes	3	\standardstate	4, 121
notesbefore	3	T	
rademph	4	\tBu	4, 154
radhyphen	4	X	
radprefix	4	xspace (option)	4
radsuper	4		
xspace	4		
\OrgLettColour	409		