

chemstyle — Writing chemistry with style*

Joseph Wright[†]

Released 2008/03/08

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
		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
2 Style options for chemstyle	2	8 Configuration files	17
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
3 Naming of the references section	3	8.4 <i>Tetrahedron Lett.</i> style	17
		8.5 <i>J. Am. Chem. Soc.</i> style	18
4 Additional macros	4	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	9 Change History	20
5 Additional information	6	10 Index	21
5.1 Advice for users of the rsc package	6		
5.2 Interactions with other packages	6		
5.3 Captions above floats	6		
6 A template for chemical articles	7		

*This file describes version v1.2, last revised 2008/03/08.

[†]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>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>	

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`.

`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

`\Hz` Both the `Slunits` and `unitsdef` packages provide methods for handling a range
`\mmHg` of units, but do not include a small number of unit macros useful to chemists.
`\molar` The package therefore provides these, using `Slunits` by default.² The definitions
`\Molar` follow the pattern of the appropriate package.
`\cmc`
 10 Hz `\unit{10}{\Hz}\`
 20 mmHg `\unit{20}{\mmHg}\`
 30 mol dm⁻³ `\unit{30}{\molar}\`
 40 M `\unit{40}{\Molar}\`
 50 cm³ `\unit{50}{\cmc}`

The `si` package is currently under development as a replacement for `Slunits` and `unitsdef`.³ `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.⁴ This generates the tricky `\standardstate` symbol.

H^{\ominus} `\emph{H}\standardstate\`
 ΔS^{\ominus} `$\Delta S\standardstate$`

4.3 Alkyl radicals

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

²This is a change in v1.2: previous versions used `unitsdef`.

³By the maintainer of `chemstyle`.

⁴The `\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.

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

4.4 Latin phrases

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

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

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

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

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

`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>\latinemphoff</code>	<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 <code>keyval</code>
	method is now preferred for making this change.

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 the float package. This has the side-effect that the placement of captions for floats does not depend on where the \caption command comes inside the floating environment.⁵ 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]
  \caption{A caption above the float contents}
  The float contents
\end{table}
\floatstyle{plaintop}
\restylefloat{table}
\begin{table}[h]
  \caption{A second caption above the float contents}
  The second float contents
\end{table}
```

Table 2: A caption above the float contents

The float contents

Table 3: A second caption above the float contents

The second float contents

⁵Normally 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]{scrreprt}
\usepackage[english,UKenglish]{babel}
\usepackage{geometry,upgreek,booktabs}
\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 \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`-bond `\upsigma^*`-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 .

The `microtype` package improves formatting when used with the `pdfTeX` 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.

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 [2008/03/08 v1.2 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` package⁶ 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.⁷

```

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}

```

⁶An improved replacement for all of the existing unit packages.

⁷This 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{scheme}
155 \restylefloat{figure}
156 \floatplacement{table}{htbp}
157 \floatplacement{scheme}{htbp}
158 \floatplacement{figure}{htbp}

```

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

```

159 \floatcontentscentre

```

7.7 Cross-references

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

```

160 \labelformat{scheme}{\schemename~#1}
161 \labelformat{figure}{\figurename~#1}
162 \labelformat{table}{\tablename~#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.

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

```
170 \ifcst@phrases
171   \AtBeginDocument{
172     \providecommand*\etc{
173       {\@ifnextchar.{\cst@etc}{\cst@etc.\cst@xspace}}
174     \providecommand*\invacuo{
175       {\latin{in vacuo}\cst@xspace}
176     \providecommand*\etal{
177       {\@ifnextchar.{\cst@etal}{\cst@etal.\cst@xspace}}
178     \providecommand*\eg{
179       {\ifcst@abbrcomma
180         \expandafter\cst@commaabbr%
181       \else
182         \expandafter\cst@nocommaabbr%
183       \fi
184       {eg}}
185     \providecommand*\ie{
186       {\ifcst@abbrcomma
187         \expandafter\cst@commaabbr%
188       \else
189         \expandafter\cst@nocommaabbr%
190       \fi
191       {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
192 \newcommand*{\cst@etal}{\latin{et~al}}
193 \newcommand*{\cst@etc}{\latin{etc}}
194 \newcommand*{\cst@ie}{\latin{i.e\spacefactor999\relax}}
195 \newcommand*{\cst@eg}{\latin{e.g\spacefactor999\relax}}
```

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

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

```

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

```

203 \newcommand*{\cst@addpunct}{%
204 \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.

```

205 \newcommand*{\cst@add@punct}[1]{%
206 \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
207 \newcommand*{\cst@add@punct@}{%
208 \ifx.\cst@tempa
209 \let\cst@tempb\cst@add@punct
210 \else
211 \ifx,\cst@tempa
212 \let\cst@tempb\cst@add@punct
213 \else
214 \let\cst@tempb\cst@commapunct
215 \fi
216 \fi
217 \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.

```

218 \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`.

```

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

```

```

228         found\MessageBreak This is a core part of chemstyle
229         \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.⁸ A second check is needed, in case the document class does not define any reference section at all (for example, `minimal` does not).

```

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

```

239 \ifcst@autonotes
240   \AtBeginDocument{
241     \@ifpackageloaded{notes2bib}{}
242     {\newcounter{bibnote}}
243   \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@
244 \newcommand*{\cst@refname}{%
245   \ifcst@notes
246     \ifcst@autonotes
247       \ifnum\the\value{bibnote}>\z@
248         \expandafter\expandafter\expandafter\expandafter\expandafter
249         \expandafter\expandafter\cst@ref@name%
250       \else
251         \expandafter\expandafter\expandafter\expandafter\expandafter
252         \expandafter\expandafter\cst@ref@name@%
253       \fi
254     \else
255       \expandafter\expandafter\expandafter\cst@ref@name%
256     \fi
257   \else
258     \expandafter\cst@ref@name@%

```

⁸If `\@undefined` is ever fixed, this will no longer be needed. If ϵ -TeX was required, the `\ifdefined` would be a choice here; however, ϵ -TeX is not going to be required just for this.

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```
296 \ProvidesFile{rsc.jdf}[2007/10/31 v1.1h]
297 \AtBeginDocument{%
298   \renewcommand{\figurename}{Fig.}}
299 \captionsetup{labelsep=quad,labelfont=bf}
300 \setkeys{cst}{opt}{
301   radprefix=true,
302   rademph=true,
303   radsuper=false,
304   radhyphen=true,
305   abbrevemph=true,
306   abbrcomma=false}
```

8.2 *Angew. Chem.* style

```
307 \ProvidesFile{angew.def}[2007/10/31 v1.1h]
308 \captionsetup{labelsep=period,labelfont={bf,it},font=sf,singlelinecheck=off}
309 \captionsetup[table]{labelsep=colon}
310 \setkeys{cst}{opt}{
311   radprefix=true,
312   rademph=true,
313   radsuper=false,
314   radhyphen=false,
315   abbrevemph=false,
316   abbrcomma=false}
```

8.3 *J. Organomet. Chem.* style

```
317 \ProvidesFile{jomc.def}[2007/10/31 v1.1h]
318 \captionsetup{labelsep=period}
319 \captionsetup[table]{labelsep=newline,singlelinecheck=off}
320 \AtBeginDocument{%
321   \renewcommand{\figurename}{Fig.}}
322 \setkeys{cst}{opt}{
323   radprefix=true,
324   rademph=true,
325   radsuper=true,
326   radhyphen=false,
327   abbrevemph=false,
328   abbrcomma=false,
329   notesbefore=true}
```

8.4 *Tetrahedron Lett.* style

```
330 \ProvidesFile{tetlett.def}[2007/10/31 v1.1h]
331 \captionsetup{labelsep=period,singlelinecheck=off,labelfont=bf}
332 \setkeys{cst}{opt}{
```

```

333 radprefix=true,
334 rademph=true,
335 radsuper=false,
336 radhyphen=true,
337 abbrevemph=false,
338 abbrcomma=true,
339 notesbefore=true}

```

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

```

340 \ProvidesFile{jacs.jdf}[2007/10/31 v1.1h]
341 \DeclareCaptionLabelSeparator{perquad}{.\quad}
342 \captionsetup{labelfont={bf,it,sf},textfont=sf,labelsep=perquad}
343 \captionsetup[figure]{textfont=rm}
344 \captionsetup{singlelinecheck=off}
345 \setkeys{cst}{opt}{
346   radprefix=true,
347   rademph=true,
348   radsuper=true,
349   radhyphen=false,
350   abbrevemph=false,
351   abbrcomma=true,
352   notesbefore=true}
353 \floatstyle{plaintop}
354 \restylefloat{scheme}
355 \floatstyle{plain}
356 \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`.

```

357 \ProvidesFile{ic.jdf}[2007/10/31 v1.1h]
358 \input {jacs.jdf}
359 \captionsetup{textfont=rm}

```

8.7 *J. Phys. Chem.* style

```

360 \ProvidesFile{jpc.jdf}[2007/10/31 v1.1h]
361 \DeclareCaptionFormat{labelcaps}{\MakeUppercase{#1}#2#3}
362 \captionsetup{font=bf,labelsep=colon,format=labelcaps}
363 \captionsetup[figure]{format=plain,textfont=md,labelsep=period}
364 \captionsetup{singlelinecheck=off}
365 \setkeys{cst}{opt}{
366   radprefix=true,
367   rademph=true,
368   radsuper=false,
369   radhyphen=false,
370   abbrevemph=false,
371   abbrcomma=true,
372   notesbefore=true}
373 \floatstyle{plaintop}
374 \restylefloat{scheme}
375 \floatstyle{plain}
376 \cst@notesbeforefalse

```

8.8 *Org. Lett.* style

```
377 \ProvidesFile{orglett.jdf}[2007/10/31 v1.1h]
378 \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.

```
379 \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.

```
380 \newcommand*{\fs@orglett}{%
381   \def\@fs@cfont{\bfseries}%
382   \let\@fs@capt\floatc@ruled
383   \def\@fs@pre{\begingroup\color{\OrgLettColour}
384     \hrule height12pt depth0pt \kern2pt\endgroup}%
385   \def\@fs@mid{\vspace\belowcaptionskip\relax}%
386   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
387     \hrule height1.5pt depth0pt\endgroup}%
388   \let\@fs@iftopcapt\iftrue}
```

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

```
385   \def\@fs@mid{\vspace\belowcaptionskip\relax}%
386   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
387     \hrule height1.5pt depth0pt\endgroup}%
388   \let\@fs@iftopcapt\iftrue}
```

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

```
389 \newcommand*{\fs@orglettfig}{%
390   \def\@fs@cfont{\bfseries}%
391   \let\@fs@capt\floatc@plain
392   \def\@fs@pre{\begingroup\color{\OrgLettColour}%
393     \hrule height12pt depth0pt \kern2pt\endgroup}%
394   \def\@fs@mid{\vspace\abovecaptionskip\relax}%
395   \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
396     \hrule height1.5pt depth0pt\endgroup}%
397   \let\@fs@iftopcapt\iffalse}
```

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

```
398 \floatstyle{orglettfig}
399 \restylefloat{figure}
400 \floatstyle{orglett}
401 \restylefloat{scheme}
402 \restylefloat{table}
```

The more usual style commands now occur.

```
403 \DeclareCaptionLabelSeparator{perquad}{.\quad}
404 \captionsetup{labelfont=bf,labelsep=perquad}
405 \setkeys{cst}{opt}{
406   radprefix=true,
407   rademph=true,
408   radsuper=true,
409   radhyphen=false,
410   abbrevemph=false,
411   abbrcomma=true,
412   notesbefore=true}
413 \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 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.

```

414 \ProvidesFile{none.jdf}[2007/10/31 v1.1h]
415 \floatstyle{plain}
416 \restylefloat{table}
417 \labelformat{scheme}{#1}
418 \labelformat{figure}{#1}
419 \labelformat{table}{#1}
420 \floatplacement{table}{tbp}
421 \floatplacement{scheme}{tbp}
422 \floatplacement{figure}{tbp}

```

9 Change History

v1.0	General: Initial release of package 1	\cst@latin: New macro 13	\cst@varnothing: New macro 11
v1.0a	General: Added <i>J. Phys. Chem.</i> style 18	\eg: Adds comma for ACS journals 13	\ie: Adds comma for ACS journals 13
	Added <i>Org. Lett.</i> style 19		
	Fixed incorrect Latin formatting for <i>angew</i> option 17		
	No longer load <i>fixltx2e</i> package 8		
v1.0b	General: Added <i>microtype</i> and <i>notes2bib</i> to template 1		
	Fixed formatting of alkyls 17		
	Sorted a problem with <i>babel</i> and figure name format 17		
v1.1	General: Added <i>phrases</i> option 1		
	Added <i>siunits</i> option 1		
	Added <i>xspace</i> option 1		
	Fixed (another) error in alkyl formatting 17		
	Fixed error with spacing after <i>e.g.</i> and <i>i.e.</i> 1		
	License changed from GPL to LPPL 1		
	Phrases modified to better avoid clash with user’s own commands 1		
	\cst@add@punct: New macro 14		
	\cst@add@punct@: New macro 14		
	\cst@addpunct: New macro 14		
		v1.1a	General: Fixed error in documentation compilation under LaTeX 1
		v1.1b	General: Load caption even when journal style does not exist 14
		v1.1c	General: Added <i>Slunits</i> option 1
			Added <i>frenchb</i> alias for <i>french</i> 16
			Added <i>german</i> alias for <i>ngerman</i> 16
			Documentation improved 1
			Fixed problems with capitalisation of <i>Slunits</i> 1
			\standardstate: Altered angle of rotation for better appearance 11
			Definition only occurs if user does not have own version 11
		v1.1d	General: Require caption in all cases 8
		v1.1e	General: Fixed packaging problem 1
		v1.1f	\cst@addpunct: No longer used when comma not required 14

\cst@eg: New macro	13	Various internal tweaks	1
\cst@etal: New macro	13	\cst@add@punct@: Internal	
\cst@etc: New macro	13	tweaks	14
\cst@ie: New macro	13	\cst@commaabbr: New macro ..	13
\eg: Definition comma-mode dependent	13	\cst@commapunct: Name change from \cst@punct	14
Fixed problem with full stop spacing	13	\cst@latin: Change of definition to allow better working with run-time options	13
\etal: Fixed problem with full stop spacing	13	\cst@nocommaabbr: New macro	13
\ie: Definition comma-mode dependent	13	\cst@radical: Re-write to allow definition to change at run-time	11
Fixed problem with full stop spacing	13	\cst@ref@name: New macro ...	15
v1.1g		\cst@ref@name@: New macro ..	15
General: Added jomc style	17	\cst@refname: New macro	15
Fixed problem with cross-referencing when babel is not loaded	12	\cst@refsec: New macro	15
v1.1h		\cst@setrefname: New macro .	16
General: Added tetlett style ..	17	\cst@tempa: New macro	8
v1.2		\cst@tempb: New macro	8
General: Added xkeyval interface ..	8	\cst@xspace: New definition ..	11
Re-write of reference-section re-naming code	15	\eg: Dynamic definition for comma mode	13
Re-write of units section	10	\ie: Dynamic definition for comma mode	13
Removed obsolete ch and chapter options	1	\latinemphoff: New definition to work with xkeyval system ..	13
		\latinemphon: New definition to work with xkeyval system	13

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.

Symbols	\cst@autonotesfalse	\cst@journal
\' 294, 295 34	.. <u>52</u> , 219, 221, 223
	\cst@autonotestru	\cst@language . <u>52</u> , 271
A 38	\cst@latin <u>163</u>
abbrcomma (option) ... 5	\cst@boolkey <u>17</u> , 19–28	\cst@nocommaabbr
abbremph (option) 5	\cst@commaabbr 182, 189, <u>196</u>
 180, 187, <u>196</u>	\cst@notesbeforefalse
C	\cst@commapunct 356, 376, 413
\c@bibnote <u>239</u> <u>214</u> , <u>218</u>	\cst@notesfalse .. 33
chemcompounds (option) 3	\cst@disablekey .. <u>73</u>	\cst@notestru 37, 42
\cmc <u>4</u> , <u>92</u>	\cst@eg <u>192</u>	\cst@radical
\cst@add@punct ..	\cst@emph <u>116</u> , 132, 134 <u>113</u> , 141–146
. 204, <u>205</u> , 209, 212	\cst@etal 177, <u>192</u>	\cst@ref@name ... <u>244</u>
\cst@add@punct@ .	\cst@etc 173, <u>192</u>	\cst@ref@name@ .. <u>244</u>
..... 206, <u>207</u>	\cst@hyphen <u>116</u> , 132, 134	\cst@refname
\cst@addpunct 198, <u>203</u>	\cst@ie 192	. <u>244</u> , 274, 281, 287
		\cst@refsec

. 230, 273, 280, 287	\ifcst@autonotes	nophrases (option) ... 5
\cst@setrefname 29, 239, 246	notes (option) 3
..... 267, 290–295	\ifcst@chemcompounds	notesbefore (option) . 4
\cst@super 116, 132, 134 19, 69	\nPr 4, 141
\cst@tempa	\ifcst@notes .. 29, 245	
15, 29, 206, 207,	\ifcst@notesbefore	O
270, 271, 276, 19, 261	options:
278, 279, 283, 285	\ifcst@phrases 19, 170	abbrcomma 5
\cst@tempb . 15, 29, 207	\ifcst@rademph 19, 126	abbremph 5
\cst@varnothing . 108	\ifcst@radhyphen	chemcompounds .. 3
\cst@xspace 19, 116	journal 2, 3
.. 136, 137, 173,	\ifcst@radprefix	nonotes 3
175, 177, 202, 218 19, 131	nophrases 5
\cstsetup	\ifcst@radsuper 19, 121	notes 3
. 3, 83, 167, 168, 235	\ifcst@xspace . 19, 138	notesbefore 4
\cubiccentimeter	\invacuo 5, 170	rademph 4
..... 92, 94	\iPr 4, 141	radhyphen 4
		radprefix 4
E		radsuper 4
\eg 5, 170	J	xspace 4
\etal 5, 170	journal (option) ... 2, 3	\OrgLettColour .. 379
\etc 5, 170		
	L	R
F	\latin 5, 163, 175, 192–195	rademph (option) 4
\fs@orglett 380	\latinemphoff . 5, 163	radhyphen (option) ... 4
\fs@orglettfig .. 389	\latinemphon .. 5, 163	radprefix (option) ... 4
		radsuper (option) 4
H	M	
\Hz 4, 92	\mmHg 4, 94	S
	\mol 94	\sBu 4, 141
I	\Molar 4, 94	\standardstate 4, 108
\iBu 4, 141	\molar 4, 94	
\ie 5, 170		T
\ifcst@abbrcomma	N	\tBu 4, 141
..... 19, 179, 186	\nBu 4, 141	
\ifcst@abbremph 19, 164	nonotes (option) 3	X
		xspace (option) 4