

# chemstyle — Writing chemistry with style <sup>\*</sup>

Joseph Wright <sup>†</sup>

Released 2007/08/23

## Abstract

The `chemstyle` package provides a “one-stop shop” for setting up formatting of LaTeX 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.

## 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` LaTeX 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

`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.

---

<sup>\*</sup>This file describes version v1.0a, last revised 2007/08/23.

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

Option	Journals using this style
none	Not applicable
angew	<i>Angew. Chem., Chem. Eur. J.</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>

Table 1: Styles provided by `chemstyle`

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.

## 2.1 Package options

In addition to the style options, `chemstyle` will pass options through to the `chemscheme` package, which is used to provide support for schemes. Users should consult the `chemscheme` documentation for the options applicable.

The package recognises the `nonotes` option, which affects how the References section is headed. 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 and Notes” section. The `nonotes` option prevents this, and results in only “References” being output.

The “References” naming commands are language-aware, *via* the `babel` interface. Currently, `chemstyle` includes appropriate labels for `babel` languages `english`, `UKenglish`, `ngerman` and `french`.<sup>1</sup> Other languages can be added if appropriate wordings are provided to the author.

<sup>1</sup>The `french` option for `babel` causes a clash with `unitsdef` concerning the command `\fg`. `babel` uses this for guillemets, while `unitsdef` uses it as an abbreviation for `\femtogram`. The `chemstyle` package prevents the clash, and leaves the `babel` definition intact.

## 3 Additional macros

### 3.1 Additional units

The `unitsdef` package provides a great number of easy to use unit commands, and handles spacing between numbers and units very well. However, it misses out a few useful units for chemistry. The `unitsdef` package is loaded by `rsc.sty`, and is then used to provide five additional units. The most obvious of these is `\Hz`, which simply gives Hz. The `unitsdef` package only defines `\hertz`, which is not as obvious. The command `\mmHg` has a non-breaking thin space, leading to mmHg. Two related commands are given for concentration: `\molar` gives mol dm<sup>-3</sup>, whilst `\Molar` gives M. Finally, the command `\cmc` is provided for generating cm<sup>3</sup>. All space correctly with numbers, so inputting `10\cmc` gives 10 cm<sup>3</sup>, with a non-breaking space.

### 3.2 The standard state symbol

`\standardstate` Related to, but not exactly a unit is the `\standardstate` command. This generates the tricky  $\ominus$  symbol, letting you put  `$\Delta S \standardstate$`  to generate  $\Delta S^\ominus$ . Note that this is safe in either text or maths mode.

### 3.3 Alkyl radicals

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

### 3.4 Latin phrases

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

## 4 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.

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.

```
1 \documentclass[fontsize=10pt,paper=a4]{scrreprt}
```

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

```
2 \usepackage[english,UKenglish]{babel}
```

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 `bpchem` or (optionally) `chemcompounds` to track compound numbers.

```
3 \usepackage[rsc]{chemstyle}
```

The `mhchem` package provides the `\ce` command for rapidly typesetting formulas, so that you can type `\ce{H2SO4}` and get  $\text{H}_2\text{SO}_4$ .

```
4 \usepackage[version=3]{mhchem}
```

Greek letters should be italic if used for variables, but upright (Roman) if used otherwise. So the standard LaTeX `\delta` is fine for typesetting an NMR shift (e.g.  $\delta = \dots$  ppm.), but not for a  $\sigma$ -bond or a  $\pi$ -complex. The `upgreek` package provides commands such as `\upsigma` and `\uppi` which are ideal for this.

```
5 \usepackage{upgreek}
```

The default LaTeX table formatting is not very good. The `booktabs` package does things properly, and has good advice in the manual. A highly-recommended package for those interested in formatting (*i.e.* all (La)TeX users!).

```
6 \usepackage{booktabs}
```

The bulk of the document can then start.

```
7 \begin{document}
8 \end{document}
```

## 5 Advice for users of `rsc` and `achemso`

The `chemstyle` package is intended as a replacement for the `rsc` package. As such, it covers almost everything the `rsc` LaTeX package does, and more. Users of `rsc` are strongly encouraged to update to using `chemstyle`.

Both `achemso` and `rsc` provide appropriate BibTeX styles. These will continue to be supported, but

## 6 The package code

### 6.1 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.

```

9 \NeedsTeXFormat{LaTeX2e}
10 \ProvidesPackage{chemstyle}
11 [2007/08/23 v1.0a Writing chemistry with style]

```

The `\fg` macro is defined in French as a *guillemet* sign, and so to avoid a clash a bit of patching is needed. A check to see if `babel` has already defined `\fg`. If it has, the definition is saved and deleted.

```

12 \ifx\fg\@undefined
13 \else
14   \ifx\fg\relax
15   \else
16     \let\cst@fg\fg
17     \let\fg\@undefined
18   \fi
19 \fi

```

Now package loading can take place.

```
20 \RequirePackage{graphicx,amssymb,varioref,xspace,unitsdef}
```

If `\cst@fg` exists, it is restored here, otherwise `\fg` is undefined. At the beginning of the document, `\fg` is defined as an abbreviation for a femtogram if it is not being used by `babel`. This way there is not a problem if `babel` is loaded after `chemstyle`.

```

21 \ifx\cst@fg\@undefined
22   \let\fg\@undefined
23 \else
24   \let\fg\cst@fg
25 \fi
26 \AtBeginDocument{%
27   \ifx\fg\@undefined
28     \newcommand{\fg}{\femtogram}
29   \fi
30 }

```

## 6.2 Extra units and related commands

`\cubiccentimeter` A few additional unit types are provided, which the `unitsdef` package omits for some reason. These do not change with different journal styles, and so are declared directly.

```

\cubiccentimeter 31 \newunit{\cubiccentimeter}{\cm\unitsuperscript{3}}
\cmc              32 \newcommand{\cmc}{\cubiccentimeter}
\Hz              33 \newcommand{\Hz}{\hertz}
\Molar           34 \newunit{\Molar}{\textsc{m}}
\molar           35 \newunit{\molar}{\mole\unitsep\dm\unitsuperscript{--3}}
\mmHg            36 \newunit{\mmHg}{mm\unitsep}{Hg}

```

`\standardstate` In a very similar vein, the “standard state” symbol is handy. This is produced by rotating a symbol from the AMS symbol set. Note that the rotation angle here has been carefully checked, but is set by eye. Changing the font used for `\varnothing` will mess this up.

```

37 \newcommand*{\standardstate}%
38   {\textsuperscript{\rotatebox[origin=c]{140.8}%
39     {\ensuremath{\varnothing}}}}

```

### 6.3 Alkyl radicals

Alkyl radical abbreviations are produced using the macro `\cst@radical`. This uses a series of formatting commands, which have default values provided here. The names are hopefully pretty self-explanatory.

```

40 \let\cst@emph\relax
41 \def\cst@hyph{}
42 \let\cst@super\relax
43 \newif \ifcst@prefix \cst@prefixtrue

```

`\cst@radical` The `\cst@radical` macro does the hard work of declaring each abbreviation. To ensure there is no unexpected clash of names, `\newcommand` is used first to check for any problems. The commands are made robust so they can (hopefully) be used anywhere.

```

44 \def\cst@radical#1#2{%
45   \expandafter\newcommand\expandafter{\csname #1#2\endcsname}{}%
46   \expandafter\DeclareRobustCommand\expandafter{%
47     \csname #1#2\endcsname}{%
48       \ifcst@prefix
49         \cst@super{\cst@emph{#1}}\cst@hyph#2%
50       \else
51         #2\cst@hyph\cst@super{\cst@emph{#1}}%
52       \fi%
53       \xspace
54     }
55 }

```

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

```

56 \cst@radical{n}{Pr}%
57 \cst@radical{i}{Pr}%
58 \cst@radical{n}{Bu}%
59 \cst@radical{i}{Bu}%
60 \cst@radical{s}{Bu}%
61 \cst@radical{t}{Bu}%

```

### 6.4 Option handling

Firstly, the “pass through” options are handled, as these will be sent through to chemscheme. The package option `nonotes` is also handled here.

```

62 \newif \ifcst@notes \cst@notesttrue
63 \DeclareOption{ch}{\ExecuteOptions{chapter}}
64 \DeclareOption{chapter}%
65   {\PassOptionsToPackage{chapter}{chemscheme}}

```

```

66 \DeclareOption{chemcompounds}%
67   {\PassOptionsToPackage{chemcompounds}{chemscheme}}
68 \DeclareOption{nonotes}{\cst@notesfalse}

```

In order to handle the loading of style definitions correctly, a bit of care is needed. The package options need to be loaded *before* chemscheme, but the style definitions need to be loaded after chemscheme. This is achieved by saving the options and looking at them again once the package loading is done. Notice that only one style definition can be used, so this is taken care of with a warning if needed.

```

69 \let\cst@option\relax
70 \DeclareOption*{%
71   \ifx\cst@option\relax\else
72     \PackageWarning{chemstyle}
73       {Extra option \cst@option\space ignored}
74   \fi
75   \edef\cst@option{\CurrentOption}
76 }
77 \ProcessOptions*
78 \RequirePackage{chemscheme}

```

## 6.5 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.

```

79 \@ifclassloaded{memoir}
80   {\let\newfloat\@undefined
81     \RequirePackage{float}}
82   {}

```

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.

```

83 \floatstyle{plaintop}
84 \restylefloat{table}
85 \floatstyle{plain}
86 \restylefloat{scheme}
87 \restylefloat{figure}
88 \floatplacement{table}{htbp}
89 \floatplacement{scheme}{htbp}
90 \floatplacement{figure}{htbp}

```

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

```
91 \floatcontentscentre
```

## 6.6 Cross-references

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

```
92 \labelformat{scheme}{\schemename~#1}
93 \labelformat{figure}{\figurename~#1}
94 \labelformat{table}{\tablename~#1}
```

## 6.7 Latin words

`\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. The default is then set to emphasise Latin words.

```
95 \ifx\latin\@undefined
96   \newcommand{\latinemphon}{\let\latin\emph}
97   \newcommand{\latinemphoff}{\let\latin\relax}
98   \latinemphon
99 \else
100  \let\latinemphon\relax
101  \let\latinemphoff\relax
102 \fi
103
```

`\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.

`\invacuo`

`\etal`

`\eg`

`\ie`

```
104 \providecommand{\etc}{\latin{etc}\@ifnextchar.{}{.\xspace}}
105 \providecommand{\invacuo}{\latin{in vacuo}\xspace}
106 \providecommand{\etal}{\latin{et~al}\@ifnextchar.{}{.\xspace}}
107 \providecommand{\eg}{\latin{e.g}\@ifnextchar.{}{.\xspace}}
108 \providecommand{\ie}{\latin{i.e}\@ifnextchar.{}{.\xspace}}
```

## 6.8 Handling section naming

First, a new `if` is needed to differentiate between “Notes and References” and “References and Notes.”

```
109 \newif \ifcst@notesbefore \cst@notesbeforetrue
```

`\cst@name@refs` To keep life simple in the main macro, and to save on redundant code, the ordering of “Notes,” “and” and “References” is handled here.

```
110 \def\cst@name@refs#1#2#3{%
111   \ifcst@notesbefore
112     #3\ #2\ #1%
```



```

113 \else
114   #1\ #2\ #3%
115 \fi
116 }

```

`\cst@language` The default language of the document is probably English. However, this is alterable if needed, as it is not set in stone. Anyone needing to mess with this is probably happy using `\makeatletter`, so the command is kept out of user space.

```

117 \def\cst@language{english}

```

`\cst@refsection` The existence of a suitable reference section name command is checked here. Some document classes (*e.g.* `minimal`) may not define a suitable command. Assuming that is not the case, `\cst@refsection` is used to store the name of the macro holding the references section.

```

118 \ifx\refname\@undefined
119   \ifx\bibname\@undefined
120     \PackageWarning{chemstyle}%
121       {No bibliography name command defined in document class}
122     \def\cst@namerefs#1#2#3#4{%
123   \else
124     \def\cst@refsection{\bibname}
125   \fi
126 \else
127   \def\cst@refsection{\refname}
128 \fi

```

`\cst@namerefs` The main macro of the heading-altering section takes four parameters,  $\{\langle language \rangle\}$ ,  $\{\langle references \rangle\}$ ,  $\{\langle and \rangle\}$  and  $\{\langle notes \rangle\}$ . The first is the babel label for the language that the words are four, and the other three parameters and the words “References”, “and” and “Notes”, respectively.

`\cst@language@check`

```

129 \ifx\cst@namerefs\@undefined
130   \def\cst@namerefs#1#2#3#4{%
131     \ifcst@notes%

```

The non-babel version of the function only does anything if the language passed in #1 is the default document language (probably English). The check for this needs the macro `\cst@language@check` to get `\ifx` to behave as desired.

```

132     \def\cst@language@check{#1}
133     \ifx\cst@language\cst@language@check
134       \expandafter\renewcommand\expandafter{\cst@refsection}%
135       {\cst@name@refs{#2}{#3}{#4}}%
136     \fi

```

To ensure that things work whether babel is loaded or not, a few hoops have to be jumped through. Altering the babel strings is delayed until the start of the document, in case babel is loaded after `chemstyle`. There is also a complication that MikTeX defines `\language` even if babel is not loaded. This may

contain an undesired value, and so the presence of `babel` needs to be tested for to avoid strange errors.

```

137 \AtBeginDocument{%
138 \ifpackageloaded{babel}{%
139 \expandafter\addto\expandafter{\csname captions#1\endcsname}%
140 {\expandafter\renewcommand\expandafter{\cst@refsection}%
141 {\cst@name@refs{#2}{#3}{#4}}}%
142 \expandafter\selectlanguage\expandafter{\language}%
143 {}%
144 \else%
145 \ifx#1\cst@language
146 \expandafter\renewcommand\expandafter{\cst@refsection}{#2}%
147 \fi
148 \AtBeginDocument{%
149 \ifpackageloaded{babel}{%
150 \expandafter\addto\expandafter{\csname captions#1\endcsname}%
151 {\expandafter\renewcommand\expandafter{\cst@refsection}{#2}}%
152 \expandafter\selectlanguage\expandafter{\language}%
153 {}%
154 \fi%
155 }
156 \fi

```

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

```

157 \cst@namerefs{english}{References}{and}{Notes}
158 \cst@namerefs{UKenglish}{References}{and}{Notes}
159 \cst@namerefs{ngerman}{Literatur}{und}{Notizen}
160 \cst@namerefs{french}{R\'{e}f\'{e}rences}{et}{Notes}

```

## 6.9 Loading the style definition

The style definition is loaded here, once else everything 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`.

```

161 \ifx\cst@option\relax
162 \PackageWarning{chemstyle}
163 {You didn't say which style to use \MessageBreak
164 Defaulting to the ``no change'' style: none}
165 \def\cst@option{none}
166 \fi
167 \InputIfFileExists{\cst@option.jdf}
168 {\PackageInfo{chemstyle}
169 {Loaded \cst@option.jdf}}
170 {\PackageWarning{chemstyle}
171 {Required style \cst@option\space does not exist}}

```

## 7 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.

### 7.1 RSC style

```
172\ProvidesFile{rsc.jdf}[2007/08/23 v1.0a]
173\renewcommand{\figurename}{Fig.}
174\RequirePackage[labelsep=quad,labelfont=bf]{caption}
175\let\cst@emph\relax
176\def\cst@hyph{}
177\let\cst@super\textsuperscript
178\cst@prefixfalse
```

### 7.2 *Angew. Chem.* style

```
179\ProvidesFile{cst-angew.def}[2007/08/23 v1.0a]
180\RequirePackage[labelsep=period,labelfont={bf,it},font=sf]{caption}
181\captionsetup[table]{labelsep=colon}
182\captionsetup{singlelinecheck=off}
183\let\cst@emph\emph
184\def\cst@hyph{}
185\let\cst@super\relax
186\cst@prefixtrue
187\latinemphoff
```

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

```
188\ProvidesFile{jacs.jdf}[2007/08/23 v1.0a]
189\RequirePackage[labelfont={bf,it,sf},textfont=sf]{caption}
190\DeclareCaptionLabelSeparator{perquad}{.\quad}
191\captionsetup[labelsep=perquad]
192\captionsetup[figure]{textfont=rm}
193\captionsetup{singlelinecheck=off}
194\let\cst@emph\emph
195\def\cst@hyph{}
196\let\cst@super\textsuperscript
197\cst@prefixtrue
198\floatstyle{plaintop}
199\restylefloat{scheme}
200\floatstyle{plain}
201\latinemphoff
202\cst@notesbeforefalse
```

### 7.4 *Inorg. Chem.* style

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

```
203\ProvidesFile{ic.jdf}[2007/08/23 v1.0a]
```

```
204 \input {jacs.jdf}
205 \captionsetup{textfont=rm}
```

## 7.5 *J. Phys. Chem.* style

```
206 \ProvidesFile{jpc.jdf}[2007/08/23 v1.0a]
207 \RequirePackage[font=bf,labelsep=colon]{caption}
208 \DeclareCaptionFormat{labelcaps}{\MakeUppercase{#1}#2#3}
209 \captionsetup{format=labelcaps}
210 \captionsetup[figure]{format=plain,textfont=md,labelsep=period}
211 \captionsetup{singlelinecheck=off}
212 \let\cst@emph\emph
213 \def\cst@hyph{}
214 \let\cst@super\textsuperscript
215 \cst@prefixtrue
216 \floatstyle{plaintop}
217 \restylefloat{scheme}
218 \floatstyle{plain}
219 \latinemphoff
220 \cst@notesbeforefalse
```

## 7.6 *Org. Lett.* style

```
221 \ProvidesFile{orglett.jdf}[2007/08/23 v1.0a]
222 \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.

```
223 \newcommand*{\OrgLettColour}{purple}
```

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.

```
224 \newcommand\fs@orglett{\def\@fs@cfont{\bfseries}
225 \let\@fs@capt\floatc@ruled
226 \def\@fs@pre{\begingroup\color{\OrgLettColour}
227 \hrule height12pt depth0pt \kern2pt\endgroup}%
```

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

```
228 \def\@fs@mid{\vspace\belowcaptionskip\relax}%
229 \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
230 \hrule height1.5pt depth0pt\endgroup}%
231 \let\@fs@iftopcapt\iftrue}
```

Figures need to be slightly different, so an almost identical command is needed.

```
232 \newcommand\fs@orglettfig{\def\@fs@cfont{\bfseries}
233 \let\@fs@capt\floatc@plain
234 \def\@fs@pre{\begingroup\color{\OrgLettColour}
235 \hrule height12pt depth0pt \kern2pt\endgroup}%
236 \def\@fs@mid{\vspace\abovecaptionskip\relax}%
237 \def\@fs@post{\begingroup\color{\OrgLettColour}\kern2pt
```

```

238 \hrule height1.5pt depth0pt\endgroup}%
239 \let\@fs@iftopcapt\iffalse}

```

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

```

240 \floatstyle{orglettfig}
241 \restylefloat{figure}
242 \floatstyle{orglett}
243 \restylefloat{scheme}
244 \restylefloat{table}

```

The more usual style commands now occur.

```

245 \RequirePackage[labelfont=bf]{caption}
246 \DeclareCaptionLabelSeparator{perquad}{.\quad}
247 \captionsetup{labelsep=perquad}
248 \let\cst@emph\emph
249 \def\cst@hyph{}
250 \let\cst@super\textsuperscript
251 \cst@prefixtrue
252 \latinemphoff
253 \cst@notesbeforefalse

```

## 7.7 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 nothing 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.

```

254 \ProvidesFile{none.jdf}[2007/08/23 v1.0a]
255 \RequirePackage{caption}
256 \floatstyle{plain}
257 \restylefloat{table}
258 \labelformat{scheme}{#1}
259 \labelformat{figure}{#1}
260 \labelformat{table}{#1}
261 \floatplacement{table}{tbp}
262 \floatplacement{scheme}{tbp}
263 \floatplacement{figure}{tbp}

```

## Change History

v1.0		Fixed incorrect Latin formatting for <code>angew</code> option . . . . .	11
General: Initial release of package	. . . . .		1
v1.0a		No longer load <code>fixltx2e</code> pack- age . . . . .	5
General: Added <i>J. Phys. Chem.</i> style	12		
Added <i>Org. Lett.</i> style . . . . .	12		

# Index

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

<b>Symbols</b>	<code>\cst@prefixtrue</code> .	<code>\iPr</code> ..... 3
<code>\'</code> ..... 160	43, 186, 197, 215, 251	
	<code>\cst@radical</code> <u>44</u> , 56–61	<b>L</b>
	<code>\cst@refsection</code> .	<code>\latin</code> ... 3, <u>95</u> , 104–108
<code>\_</code> ..... 112, 114	..... <u>118</u> ,	<code>\latinemphoff</code> ...
	134, 140, 146, 151	<u>95</u> , 187, 201, 219, 252
<b>C</b>	<code>\cst@super</code> .....	<code>\latinemphon</code> ..... <u>95</u>
<code>\cmc</code> ..... 3, <u>31</u>	. 42, 49, 51, 177,	
<code>\cst@emph</code> .....	185, 196, 214, 250	<b>M</b>
. 40, 49, 51, 175,	<code>\cubiccentimeter</code> . <u>31</u>	<code>\mmHg</code> ..... 3, <u>31</u>
183, 194, 212, 248		<code>\Molar</code> ..... 3, <u>31</u>
<code>\cst@fg</code> ..... 16, 21, 24	<b>D</b>	<code>\molar</code> ..... 3, <u>31</u>
<code>\cst@hyph</code> .....	<code>\dm</code> ..... 35	<code>\mole</code> ..... 35
. 41, 49, 51, 176,		
184, 195, 213, 249	<b>E</b>	<b>N</b>
<code>\cst@language</code> ...	<code>\eg</code> ..... 3, <u>104</u>	<code>\nBu</code> ..... 3
..... <u>117</u> , 133, 145	<code>\etal</code> ..... 3, <u>104</u>	
<code>\cst@language@check</code>	<code>\etc</code> ..... 3, <u>104</u>	<b>O</b>
..... <u>129</u>	<b>F</b>	<code>\OrgLettColour</code> <u>223</u> ,
<code>\cst@name@refs</code> ..	<code>\fs@orglett</code> ..... 224	226, 229, 234, 237
..... <u>110</u> , 135, 141	<code>\fs@orglettfig</code> .. 232	
<code>\cst@namerefs</code> ...		<b>R</b>
.. 122, <u>129</u> , 157–160	<b>H</b>	<code>\relax</code> 14, 40, 42, 69, 71,
<code>\cst@notesbeforefalse</code>	<code>\Hz</code> ..... 3, <u>31</u>	97, 100, 101, 161,
..... 202, 220, 253		175, 185, 228, 236
<code>\cst@notesbeforetrue</code>	<b>I</b>	<b>S</b>
..... 109	<code>\iBu</code> ..... 3	<code>\sBu</code> ..... 3
<code>\cst@notesfalse</code> .. 68	<code>\ie</code> ..... 3, <u>104</u>	<code>\schemename</code> ..... 92
<code>\cst@notestruer</code> ... 62	<code>\ifcst@notes</code> .. 62, 131	<code>\standardstate</code> . 3, <u>37</u>
<code>\cst@option</code> ... 69,	<code>\ifcst@notesbefore</code>	
71, 73, 75, 161,	..... 109, 111	<b>T</b>
165, 167, 169, 171	<code>\ifcst@prefix</code> .. 43, 48	<code>\tBu</code> ..... 3
<code>\cst@prefixfalse</code> 178	<code>\invacuo</code> ..... 3, <u>104</u>	