

rsc — BibTeX styles for Royal Society of Chemistry and Wiley journals*

Joseph Wright[†]

Released 2008/08/27

Abstract

The rsc package provides BibTeX style files to produce bibliographies in accordance with the guidelines of the Royal Society of Chemistry and Wiley chemistry-related journals. The styles require the use of natbib and mciteplus. In addition, a short LaTeX package is included; this provides a convenient user interface to the customisation hooks made available by the BibTeX styles.

Contents	4	The LaTeXsource	2
1 Introduction	1	5 The BibTeX source	4
2 Using the styles	1	6 Change History	32
3 Customising the styles: the LaTeXpackage	2	7 References	32

1 Introduction

Although synthetic chemists do not, in the main, use LaTeX for the preparation of journal articles, it would be nice to be able to use it for reports. The package achemso provides for a BibTeX style and other support for reports in the style of the American Chemical Society. The aim of the rsc package is to provide similar support for the style favoured by the Royal Society of Chemistry. The package also provides support for the style used in Wiley-published journals.

As of version 3, this bundle requires the use of natbib and mciteplus. These two packages make creating bibliographies much easier for chemists. As both require correctly-designed BibTeX styles, providing versions supporting different combinations had become error-prone and confusing.

2 Using the styles

The styles are used in the normal way for BibTeX styles, as the argument to the \bibliographystyle macro. As both natbib and mciteplus are required, these must of course be loaded. Citations can then be given as normal:

*This file describes version v3.0, last revised 2008/08/27.

[†]E-mail: joseph.wright@morningstar2.co.uk

Some standard citations.^{1,2}

A multiple citation.³

Some natbib-specific features: Abernethy *et al.*¹ have said something, as reported by Cotton *et al.*, in 1992.

Some standard citations

```
\cite{Abernethy2003,Cotton1999}.\
```

A multiple citation

```
\cite{Arduengo1992,*Arduengo1994}.\
```

Some `\textsf{natbib}`-specific features:

```
\citet{Abernethy2003} have said something,  
as reported by \citeauthor{Cotton1999},  
in \citeyear{Arduengo1992}.
```

To make life slightly easier for the user, a small \LaTeX package accompanies the `BibTeX` files. It automatically loads the support packages, and handles customisation (see the next section).

```
\documentclass{article}  
\usepackage{rsc}  
\begin{document}  
% Document body here  
\bibliographystyle{rsc} % or angew  
\bibliography{your-bibtex-database}  
\end{document}
```

3 Customising the styles: the \LaTeX package

Both of the styles provide a small number of control hooks to modify the output behaviour. These can be altered directly, using a mechanism similar to that used by `IEEEtrans`. However, for most users, this is overly complex. A short \LaTeX package is therefore provided, which provides high-level access to the customisation.

`maxnames` The `maxnames` option sets whether and when a long list of authors is truncated, and takes an integer value. Giving the value 0 means that no truncation occurs, otherwise the list is truncated if the number of names exceeds the given value. The `usetitle` and `usedoi` options are simple Boolean switches. The `usetitle` option sets whether to include the titles of journal articles in the bibliography. The default is `false`; the only RSC journal using article titles is *Photochemical & Photobiological Sciences*. The `usedoi` option forces the use of DOI information in cases where it would not normally be present (for example journal articles with page ranges available). This is intended to make it easier for readers to find journal information.

`super` The `super` Boolean option turns on superscript citations, and is `true` by default. It also causes the `natmove` package to be loaded if available (`natmove` is part of the `achemso` bundle). This ensures that citations appear *after* punctuation even if given before in the text. This documentation has been compiled using `natmove`: see the input of the example citations in the previous section.

4 The \LaTeX source

The package starts with the usual identification.

```
1 \NeedsTeXFormat{LaTeX2e}  
2 \ProvidesPackage{rsc}  
3 [2008/08/27 v3.0 Support for RSC and Wiley journals]
```

```

4 \RequirePackage{mciteplus,kvoptions}

\ifrsc@super To make life easier for the user, the control values for the bibliography are
\ifrsc@usetitle converted into LATEX package options using kvoptions.
\ifrsc@usedoi
\rsc@maxauthors
5 \SetupKeyvalOptions{
6   family = rsc,
7   prefix = rsc@}
8 \DeclareBoolOption{super}
9 \DeclareBoolOption{usetitle}
10 \DeclareBoolOption{usedoi}
11 \DeclareStringOption{maxauthors}
12 \setkeys{rsc}{
13   maxauthors = 0,
14   usetitle = false,
15   usedoi = false,
16   super = true}
17 \ProcessKeyvalOptions{rsc}

\rsc@prefix Some set up work is carried out for writing to the control bibliography.
\rsc@msg
18 \newcommand*{\rsc@prefix}{rsc-}
19 \edef\rsc@msg{%
20   This is an auxiliary file used by the 'rsc' package.^^J%
21   This file may safely be deleted. It will be recreated as
22   required.^^J}

\rsc@out Assuming files are being written, the control values are placed in the control
\rsc@stream BibTEX database.
23 \if@filesw
24   \newwrite\rsc@out
25   \newcommand*\rsc@stream{\rsc@prefix\jobname.bib}%
26   \immediate\openout\rsc@out\rsc@stream\relax
27   \immediate\write\rsc@out{\rsc@msg}%
28   \edef\@tempa#1#2{\space\space#1\space=\space"#2",^^J}%
29   \immediate\write\rsc@out{%
30     @Control\string{rsc-control,^^J%
31     \@tempa{ctrl-use-title}{\ifrsc@usetitle yes\else no\fi}%
32     \@tempa{ctrl-et al-number}{\rsc@maxauthors}%
33     \@tempa{ctrl-use-doi-all}{\ifrsc@usedoi yes\else no\fi}%
34     \string}^^J}%
35 \fi

\rsc@bibliography Finally, the control database and a suitable citation are added to the document
\bibliography body. The citation has to come right at the end of the \document macro.
36 \AtBeginDocument{
37   \let\rsc@bibliography\bibliography
38   \renewcommand*\bibliography[1]{%
39     \rsc@bibliography{\rsc@prefix\jobname,#1}}
40 \g@addto@macro{\document}{%
41   \if@filesw
42     \immediate\write\@auxout{%
43       \string\citation\string{rsc-control\string}}%
44   \fi}

```

Citation support is loaded.

```
45 \ifrsc@super
46   \RequirePackage[sort&compress,numbers,super]{natbib}
47   \IfFileExists{natmove.sty}
48     {\RequirePackage{natmove}}
49     {}
50 \else
51   \RequirePackage[sort&compress,numbers]{natbib}
52 \fi
```

5 The Bib_T_EX source

Very little of the Bib_T_EX source is very exciting, with most of it taken from the standard classes, the natbib and mciteplus classes or *Tame the BeaST*.

```
53 ENTRY
54 { address
55   author
56   booktitle
57   chapter
```

The next three entries are used to allow formatting control, in the style of biblatex.

```
58   ctrl-use-title
59   ctrl-et al-number
60   ctrl-use-doi-all
61   doi
62   edition
63   editor
64   howpublished
65   institution
66   journal
67   key
68   note
69   number
70   organization
71   pages
72   publisher
73   school
74   series
75   title
76   type
77   url
78   volume
79   year
80 }
81 {}
82 { label
83   extra.label
84   short.list
85 }
86
87 INTEGERS { output.state before.all mid.sentence }
88 INTEGERS { after.sentence after.block after.item }
89 INTEGERS { author.or.editor }
```

```

90
91 FUNCTION {init.state.consts}
92 { #0 'before.all :=
93   #1 'mid.sentence :=
94   #2 'after.sentence :=
95   #3 'after.block :=
96   #4 'after.item :=
97 }
98

```

Using a trick from `biblatex`, some integers are made available to control some formatting aspects without needing a new style.

```

99 INTEGERS { is.use.title etal.number use.doi.all }
100

```

A simple Boolean switch for controlling title output.

```

101 FUNCTION {default.is.use.title} { #0 }
102

```

The number of names that force “*et al.*” to be used If set to 0, all names are displayed

```

103 FUNCTION {default.etal.number} { #0 }
104

```

A simple Boolean switch for controlling DOI printing for normal entries.

```

105 FUNCTION {default.use.doi.all} { #0 }
106
107 FUNCTION {add.comma}
108 { ", " * }
109
110 STRINGS { s t }
111
112 FUNCTION {output.nonnull}
113 { 's :=
114   output.state mid.sentence =
115     { add.comma write$ }
116     { output.state after.block =
117       { add.comma write$
118         newline$
119         "\newblock " write$
120       }
121       { output.state before.all =
122         'write$
123         { output.state after.item =
124           { " " * write$ }
125           { add.period$ " " * write$ }
126         if$
127       }
128       if$
129     }
130     if$
131   mid.sentence 'output.state :=
132 }
133 if$
134 s
135 }

```

```

136
137 FUNCTION {output}
138 { duplicate$ empty$
139   'pop$
140   'output.nonnull
141   if$
142 }
143
144 FUNCTION {output.check}
145 { 't :=
146   duplicate$ empty$
147   { pop$ "Empty " t * " in " * cite$ * warning$ }
148   'output.nonnull
149   if$
150 }
151

```

Some functions taken more or less verbatim from mciteplus.

```

152 INTEGERS { would.add.period.textlen }
153
154 FUNCTION {would.add.period}
155 { duplicate$
156   add.period$
157   text.length$
158   'would.add.period.textlen :=
159   duplicate$
160   text.length$
161   would.add.period.textlen =
162   { #0 }
163   { #1 }
164   if$
165 }
166
167 FUNCTION {fin.entry}
168 { would.add.period
169   { "\relax" * write$ newline$
170     "\mciteBstWouldAddEndPuncttrue" write$ newline$
171     "\mciteSetBstMidEndSepPunct{\mcitedefaultmidpunct}"
172     write$ newline$
173     "{\mcitedefaultendpunct}{\mcitedefaultseppunct}\relax"
174   }
175   { "\relax" * write$ newline$
176     "\mciteBstWouldAddEndPunctfalse" write$ newline$
177     "\mciteSetBstMidEndSepPunct{\mcitedefaultmidpunct}"
178     write$ newline$
179     "{}{\mcitedefaultseppunct}\relax"
180   }
181   if$
182   write$
183   newline$
184   "\EndOfBibitem" write$
185 }
186
187 FUNCTION {not}
188 { { #0 }

```

```

189     { #1 }
190   if$
191 }
192
193 FUNCTION {and}
194 {   'skip$
195     { pop$ #0 }
196   if$
197 }
198
199 FUNCTION {or}
200 {   { pop$ #1 }
201     'skip$
202   if$
203 }
204
205 FUNCTION {field.or.null}
206 { duplicate$ empty$
207     { pop$ "" }
208     'skip$
209   if$
210 }
211
212 FUNCTION {emphasize}
213 { duplicate$ empty$
214     { pop$ "" }
215     { "\emph{" swap$ * "}" * }
216   if$
217 }
218
219 FUNCTION {boldface}
220 { duplicate$ empty$
221     { pop$ "" }
222     { "\textbf{" swap$ * "}" * }
223   if$
224 }
225
226 FUNCTION {paren}
227 { duplicate$ empty$
228     { pop$ "" }
229     { "(" swap$ * ")" * }
230   if$
231 }
232

```

Using independent strings makes editing easier. Here, a number of variations between the RSC and Wiley styles show up.

```

233 FUNCTION {bbl.and}
234 { "and" }
235
236 FUNCTION {bbl.chapter}
237 {rsc}{ "ch." }
238 {angew}{ "Chapter" }
239

```

```

240 FUNCTION {bbl.doi}
241  $\langle$ rsc $\rangle$ { "DOI:" }
242  $\langle$ angew $\rangle$ { "DOI" }
243
244 FUNCTION {bbl.editor}
245  $\langle$ rsc $\rangle$ { "ed." }
246  $\langle$ angew $\rangle$ { "Ed.:" }
247
248  $\langle$ angew $\rangle$ FUNCTION {bbl.editors}
249  $\langle$ angew $\rangle$ { "Eds.:" }
250
251 FUNCTION {bbl.edition}
252  $\langle$ rsc $\rangle$ { "edn." }
253  $\langle$ angew $\rangle$ { "ed." }
254
255 FUNCTION {bbl.etal}
256 { "et~al." emphasize }
257
258 FUNCTION {bbl.in}
259 { "in" }
260
261 FUNCTION {bbl.inpress}
262 { "in press" }
263
264
265 FUNCTION {bbl.msc}
266  $\langle$ rsc $\rangle$ { "M.Sc. thesis" }
267  $\langle$ angew $\rangle$ { "MSc thesis" }
268
269 FUNCTION {bbl.page}
270 { "p." }
271
272 FUNCTION {bbl.pages}
273 { "pp." }
274
275 FUNCTION {bbl.phd}
276  $\langle$ rsc $\rangle$ { "Ph.D. thesis" }
277  $\langle$ angew $\rangle$ { "PhD thesis" }
278
279 FUNCTION {bbl.submitted}
280 { "submitted for publication" }
281
282 FUNCTION {bbl.techreport}
283 { "Technical Report" }
284
285 FUNCTION {bbl.volume}
286  $\langle$ rsc $\rangle$ { "vol." }
287  $\langle$ angew $\rangle$ { "Vol." }
288
289 FUNCTION {bbl.first}
290 { "1st" }
291
292 FUNCTION {bbl.second}
293 { "2nd" }

```



```

294
295 FUNCTION {bbl.third}
296 { "3rd" }
297
298 FUNCTION {bbl.fourth}
299 { "4th" }
300
301 FUNCTION {bbl.fifth}
302 { "5th" }
303
304 FUNCTION {bbl.st}
305 { "st" }
306
307 FUNCTION {bbl.nd}
308 { "nd" }
309
310 FUNCTION {bbl.rd}
311 { "rd" }
312
313 FUNCTION {bbl.th}
314 { "th" }
315
316 FUNCTION {eng.ord}
317 { duplicate$ "1" swap$ *
318   #-2 #1 substring$ "1" =
319     { bbl.th * }
320     { duplicate$ #-1 #1 substring$
321       duplicate$ "1" =
322         { pop$ bbl.st * }
323         { duplicate$ "2" =
324           { pop$ bbl.nd * }
325           { "3" =
326             { bbl.rd * }
327             { bbl.th * }
328             if$
329           }
330         if$
331       }
332     if$
333   }
334   if$
335 }
336

```

A search and replace system taken from *Tame the BeaST*.

```

337 INTEGERS{ 1 }
338
339 FUNCTION{string.length}
340 { #1 '1 :=
341   { duplicate$ duplicate$ #1 1 substring$ = not }
342   { 1 #1 + '1 := }
343   while$
344   pop$ 1
345 }
346

```

```

347 STRINGS{replace find text}
348
349 INTEGERS{find_length}
350
351 FUNCTION{find.replace}
352 { 'replace :=
353   'find :=
354   'text :=
355   find string.length 'find_length :=
356   ""
357   { text empty$ not }
358   { text #1 find_length substring$ find =
359     { replace *
360       text #1 find_length + global.max$ substring$ 'text :=
361     }
362     { text #1 #1 substring$ *
363       text #2 global.max$ substring$ 'text :=
364     }
365     if$
366   }
367   while$
368 }
369

```

Also some checks on whether a “string” is really a number.

```

370 FUNCTION {chr.to.value}
371 { chr.to.int$ #48 -
372   duplicate$ duplicate$
373   #0 < swap$ #9 > or
374   { #48 + int.to.chr$
375     " is not a number..." *
376     warning$
377     pop$ #0
378   }
379   {}
380   if$
381 }
382
383 FUNCTION{is.a.digit}
384 { duplicate$ "" =
385   {pop$ #0}
386   {chr.to.int$ #48 - duplicate$
387     #0 < swap$ #9 > or not}
388   if$
389 }
390
391 FUNCTION{is.a.number}
392 {
393   { duplicate$ #1 #1 substring$ is.a.digit }
394   {#2 global.max$ substring$}
395   while$
396   "" =
397 }
398
399 FUNCTION {extract.num}

```

```

400 { duplicate$ 't :=
401   "" 's :=
402   { t empty$ not }
403   { t #1 #1 substring$
404     t #2 global.max$ substring$ 't :=
405     duplicate$ is.a.number
406       { s swap$ * 's := }
407       { pop$ "" 't := }
408   if$
409 }
410 while$
411 s empty$
412 'skip$
413 { pop$ s }
414 if$
415 }
416
417 FUNCTION {bibinfo.check}
418 { swap$
419   duplicate$ missing$
420   { pop$ pop$
421     ""
422   }
423   { duplicate$ empty$
424     {
425       swap$ pop$
426     }
427     { swap$
428       pop$
429     }
430   if$
431 }
432 if$
433 }
434
435 FUNCTION {convert.edition}
436 { extract.num "1" change.case$ 's :=
437   s "first" = s "1" = or
438   { bbl.first 't := }
439   { s "second" = s "2" = or
440     { bbl.second 't := }
441     { s "third" = s "3" = or
442       { bbl.third 't := }
443       { s "fourth" = s "4" = or
444         { bbl.fourth 't := }
445         { s "fifth" = s "5" = or
446           { bbl.fifth 't := }
447           { s #1 #1 substring$ is.a.number
448             { s eng.ord 't := }
449             { edition 't := }
450           if$
451         }
452       if$
453     }

```

```

454             if$
455         }
456     if$
457 }
458 if$
459 }
460 if$
461 t
462 }
463
464 FUNCTION {tie.or.space.connect}
465 { duplicate$ text.length$ #3 <
466   { "~" }
467   { " " }
468   if$
469   swap$ * *
470 }
471
472 FUNCTION {space.connect}
473 { " " swap$ * * }
474
475 INTEGERS { nameptr namesleft numnames }
476
477 FUNCTION {format.names}
478 { 's :=
479   #1 'nameptr :=
480   s num.names$ 'numnames :=
481   numnames 'namesleft :=

```

Names are truncated if there are more than `etal.number`, and it is not set to zero.

```

482   numnames etal.number > etal.number #0 > and
483   { s #1 "{f.~}{vv~}{ll}{, jj}" format.name$ 't :=
484     t bbl.etal space.connect
485   }
486   {
487     { namesleft #0 > }
488     { s nameptr "{f.~}{vv~}{ll}{, jj}" format.name$ 't :=
489       nameptr #1 >
490       { namesleft #1 >
491         { add.comma t * }
492         { numnames #2 >
493           { " " * }
494           'skip$
495         if$
496         t "others" =
497           { bbl.etal space.connect }
498 (rsc)           { bbl.and space.connect t space.connect }
499 (angew)         { add.comma t * }
500             if$
501         }
502             if$
503         }
504         't

```

```

505         if$
506         nameptr #1 + 'nameptr :=
507         namesleft #1 - 'namesleft :=
508         }
509     while$
510 }
511 if$
512 }
513
514 FUNCTION {format.authors}
515 { author empty$
516   { "" }
517   { #1 'author.or.editor :=
518     author format.names }
519   if$
520 }
521

```

The `strip.comma` function is used to “unwind” the comma added by `output.nonnull`, so that editors in brackets are easy to do. This is needed for the Wiley style.

```

522 FUNCTION {strip.comma}
523 { duplicate$
524   string.length 'find_length :=
525   duplicate$
526   find_length #1 - #1 substring$
527   ", " =
528   { #1 find_length #2 - substring$
529     " " *
530   }
531   'skip$
532   if$
533 }
534
535 FUNCTION {format.editors}
536 { editor empty$
537   { "" }
538   { #2 'author.or.editor :=
539     bbl.editor " " * *
540     editor format.names
541     strip.comma
542     editor num.names$ #1 >
543     { bbl.editors }
544     { bbl.editor }
545     if$
546     " " *
547     editor format.names * paren
548   }
549   if$
550 }
551
552 FUNCTION {format.doi}
553 { use.doi.all
554   { doi empty$

```

```

555         { bbl.doi doi tie.or.space.connect output }
556         'skip$
557     if$
558 }
559 'skip$
560 if$
561 }
562

```

`n.separate` is given in the `custom-bib` package. The following function is modified to separate out a number of any length into digit groups, *e.g.* 12345 turns into 12 345, 1234567 in 1 234 567 but 1234 or a12345 are left alone.

```

563 FUNCTION {n.separate.multi}
564 { 't :=
565   ""
566   #0 'numnames :=

```

This is the key line: the test is that the sting is both at least 5 digits long and is an integer.

```

567   t text.length$ #4 > t is.a.number and
568   {
569     { t empty$ not }
570     { t #-1 #1 substring$ is.a.number
571       { numnames #1 + 'numnames := }
572       { #0 'numnames := }
573     if$
574     t #-1 #1 substring$ swap$ *
575     t #-2 global.max$ substring$ 't :=
576     numnames #4 =
577     { duplicate$ #1 #1 substring$ swap$
578       #2 global.max$ substring$
579       "\," swap$ * *
580       #1 'numnames :=
581     }
582     'skip$
583   if$
584   }
585   while$
586 }
587 { t swap$ * }
588 if$
589 }
590
591 FUNCTION {format.bvolume}
592 { volume empty$
593   { "" }
594   { bbl.volume volume tie.or.space.connect }
595   if$
596 }
597
598 FUNCTION {format.title.noemph}
599 { 't :=
600   t empty$
601   { "" }
602   { t }

```

```

603   if$
604 }
605
606 FUNCTION {format.title}
607 { 't :=
608   t empty$
609     { "" }
610     { t emphasize }
611   if$
612 }
613
614 FUNCTION {format.url}
615 { url empty$
616   { "" }
617   { "\url{" url * "}" * }
618   if$
619 }
620
621 FUNCTION {format.title.vol}
622 { 't :=
623   t empty$
624     { "" }
625     { t emphasize }
626   if$
627   volume empty$
628     'skip$
629     { format.bvolume emphasize
630       swap$ add.comma swap$ *
631     }
632   if$
633 }
634
635 FUNCTION {format.full.names}
636 { 's :=
637   #1 'nameptr :=
638   s num.names$ 'numnames :=
639   numnames 'namesleft :=
640     { namesleft #0 > }
641     { s nameptr
642       "{vv~}{ll}" format.name$ 't :=
643       nameptr #1 >
644       {
645         namesleft #1 >
646         { ", " * t * }
647         {
648           numnames #2 >
649             { ", " * }
650             'skip$
651           if$
652           t "others" =
653             { bbl.etal * }
654             { bbl.and space.connect t space.connect }
655           if$
656         }

```

```

657         if$
658     }
659     't
660     if$
661     nameptr #1 + 'nameptr :=
662     namesleft #1 - 'namesleft :=
663 }
664 while$
665 }
666
667 FUNCTION {author.editor.full}
668 { author empty$
669   { editor empty$
670     { "" }
671     { editor format.full.names }
672     if$
673   }
674   { author format.full.names }
675   if$
676 }
677
678 FUNCTION {author.full}
679 { author empty$
680   { "" }
681   { author format.full.names }
682   if$
683 }
684
685 FUNCTION {editor.full}
686 { editor empty$
687   { "" }
688   { editor format.full.names }
689   if$
690 }
691
692 FUNCTION {make.full.names}
693 { type$ "book" =
694   type$ "inbook" =
695   or
696   'author.editor.full
697   { type$ "proceedings" =
698     'editor.full
699     'author.full
700     if$
701   }
702   if$
703 }
704
705 FUNCTION {output.bibitem} { newline$
706   "\bibitem[" write$
707   label write$
708   ")" make.full.names duplicate$ short.list =
709   { pop$ }
710   { * }

```



```

711   if$
712   "]"{" * write$
713   cite$ write$
714   }" write$
715   newline$
716   ""
717   before.all 'output.state :=
718 }
719
720 FUNCTION {n.dashify} { 't :=
721   ""
722   { t empty$ not }
723   { t #1 #1 substring$ "-" =
724     { t #1 #2 substring$ "--" = not
725       { "--" *
726         t #2 global.max$ substring$ 't :=
727       }
728       { { t #1 #1 substring$ "-" = }
729         { "-" *
730           t #2 global.max$ substring$ 't :=
731         }
732         while$
733       }
734       if$
735     }
736     { t #1 #1 substring$ *
737       t #2 global.max$ substring$ 't :=
738     }
739     if$
740   }
741   while$
742 }
743
744 FUNCTION {format.date}
745 { year empty$
746   { "" }
747   (rsc) 'year
748   (angew) { year boldface }
749   if$
750 }
751
752 FUNCTION {format.bdate}
753 { year empty$
754   { "There's no year in " cite$ * warning$ }
755   (rsc) 'year
756   (angew) { year boldface }
757   if$
758 }
759
760 FUNCTION {either.or.check}
761 { empty$
762   'pop$
763   { "Can't use both " swap$ * " fields in "
764     * cite$ * warning$ }

```

```

765  if$
766 }
767
768 FUNCTION {format.edition}
769 { edition duplicate$ empty$
770   'skip$
771   { convert.edition
772     bbl.edition bibinfo.check
773     " " * bbl.edition *
774   }
775   if$
776 }
777
778 INTEGERS { multiresult }
779
780 FUNCTION {multi.page.check}
781 { 't :=
782   #0 'multiresult :=
783   { multiresult not
784     t empty$ not
785     and
786   }
787   { t #1 #1 substring$
788     duplicate$ "-" =
789     swap$ duplicate$ "," =
790     swap$ "+" =
791     or or
792     { #1 'multiresult := }
793     { t #2 global.max$ substring$ 't := }
794     if$
795   }
796   while$
797   multiresult
798 }
799
800 FUNCTION {format.pages}
801 { pages empty$
802   { "" }
803   { pages multi.page.check
804     { bbl.pages pages n.dashify tie.or.space.connect }
805     { bbl.page pages tie.or.space.connect }
806     if$
807   }
808   if$
809 }
810
811 FUNCTION {format.pages.required}
812 { pages empty$
813   { ""
814     "There are no page numbers for " cite$ * warning$
815     output
816   }
817   { pages multi.page.check
818     { bbl.pages pages n.dashify tie.or.space.connect }

```

```

819     { bbl.page pages tie.or.space.connect }
820     if$
821     }
822     if$
823 }
824
825 FUNCTION {format.pages.nopp}
826 { pages empty$
827   { ""
828     "There are no page numbers for " cite$ * warning$
829     output
830   }
831   { pages multi.page.check
832     { pages n.dashify space.connect }
833     { pages space.connect }
834     if$
835   }
836   if$
837 }
838
839 FUNCTION {format.pages.patent}
840 { pages empty$
841   { "There is no patent number for " cite$ * warning$ }
842   { pages multi.page.check
843     { pages n.dashify }
844     (rsc) { pages n.separate.multi }
845     (angew) { pages }
846     if$
847   }
848   if$
849 }
850
851 FUNCTION {format.vol.pages}
852 (rsc){ volume boldface field.or.null
853 (angew){ volume emphasize field.or.null
854   duplicate$ empty$
855   { pop$ format.pages.required }
856   { add.comma pages n.dashify * }
857   if$
858 }
859
860 FUNCTION {format.chapter.pages}
861 { chapter empty$
862   'format.pages
863   { type empty$
864     { bbl.chapter }
865     { type "1" change.case$ }
866     if$
867     chapter tie.or.space.connect
868     pages empty$
869     'skip$
870     { add.comma format.pages * }
871     if$
872   }

```

```

873   if$
874 }
875
876 FUNCTION {format.title.in}
877 { 's :=
878 angew after.item 'output.state :=
879   s empty$
880   { "" }
881   { editor empty$
882     { bbl.in s format.title space.connect }
883     { bbl.in s format.title space.connect
884       add.comma format.editors *
885     }
886     if$
887   }
888   if$
889 }
890
891 FUNCTION {format.title.vol.in}
892 { 's :=
893   after.item 'output.state :=
894   s empty$
895   { "" }
896   { editor empty$
897     { bbl.in s format.title.vol space.connect }
898     { bbl.in s format.title.vol space.connect
899       add.comma format.editors *
900     }
901     if$
902   }
903   if$
904 }
905
906 FUNCTION {format.pub.address}
907 { publisher empty$
908   { "" }
909   { address empty$
910     { publisher }
911     { publisher add.comma address *}
912     if$
913   }
914   if$
915 }
916
917 FUNCTION {empty.misc.check}
918 { author empty$ title empty$ howpublished empty$
919   year empty$ note empty$ url empty$
920   and and and and
921   { "all relevant fields are empty in " cite$ * warning$ }
922   'skip$
923   if$
924 }
925
926 FUNCTION {empty.doi.note}

```

```

927 { doi empty$ note empty$ and
928   { "Need either a note or DOI for " cite$ * warning$ }
929   'skip$
930   if$
931 }
932
933 FUNCTION {format.thesis.type}
934 { type empty$
935   'skip$
936   { pop$
937     type emphasize
938   }
939   if$
940 }
941
942 FUNCTION {article}
943 { output.bibitem
944   format.authors "author" output.check

```

To allow titles to appear in article citations without needing a separate Bib_{TEX} style file.

```

945   is.use.title
946   { title format.title.noemph "title" output.check }
947   'skip$
948   if$
949 <rc> journal emphasize "journal" output.check
950 <angew>
951 <angew> journal ", " "" find.replace emphasize
952 <angew> "journal" output.check
953 <angew> after.item 'output.state :=
954   format.date "year" output.check
955   volume empty$
956   { "" format.pages.nopp output }
957   { format.vol.pages output }
958   if$
959   format.doi
960   fin.entry
961 }
962
963 FUNCTION {book}
964 { output.bibitem
965   author empty$
966 <rc> { title format.title "title" output.check
967 <angew> { title format.title.vol "title" output.check
968   editor empty$
969     { "Need either an author or editor for "
970       cite$ * warning$ }
971     { "" format.editors * "editor" output.check }
972     if$
973   }
974   { format.authors output
975     "author and editor" editor either.or.check
976 <rc> title format.title "title" output.check
977 <angew> title format.title.vol "title" output.check

```

```

978     }
979   if$
980     format.pub.address "publisher" output.check
981     format.edition output
982     format.bdate "year" output.check
983 (rsc) format.bvolume output
984   pages empty$
985     'skip$
986     { format.pages output }
987   if$
988     format.doi
989   fin.entry
990 }
991
992 FUNCTION {booklet}
993 { output.bibitem
994   format.authors output
995   title format.title "title" output.check
996   howpublished output
997   address output
998   format.date output
999   format.doi
1000   fin.entry
1001 }
1002
1003 FUNCTION {inbook}
1004 { output.bibitem
1005   author empty$
1006 (rsc) { title format.title "title" output.check
1007 (angew) { title format.title.vol "title" output.check
1008   editor empty$
1009   { "Need at least an author or an editor for "
1010     cite$ * warning$ }
1011   { "" format.editors * "editor" output.check }
1012   if$
1013   }
1014   { format.authors output
1015 (rsc) title format.title.in "title" output.check
1016 (angew) title format.title.vol.in "title" output.check
1017   }
1018   if$
1019     format.pub.address "publisher" output.check
1020     format.edition output
1021     format.bdate "year" output.check
1022 (rsc) format.bvolume output
1023     format.chapter.pages "chapter and pages" output.check
1024     format.doi
1025   fin.entry
1026 }
1027
1028 FUNCTION {incollection}
1029 { output.bibitem
1030   author empty$
1031   { booktitle

```

```

1032 <rsc>      format.title
1033 <angew>      format.title.vol
1034      "booktitle" output.check
1035      editor empty$
1036      { "Need at least an author or an editor for "
1037        cite$ * warning$ }
1038      { "" format.editors * "editor" output.check }
1039      if$
1040    }
1041    { format.authors output
1042      booktitle
1043 <rsc>      format.title
1044 <angew>      format.title.vol
1045      "booktitle" output.check
1046    }
1047    if$
1048    format.pub.address "publisher" output.check
1049    format.edition output
1050    format.bdate "year" output.check
1051 <rsc> format.bvolume output
1052 format.chapter.pages "chapter and pages" output.check
1053 format.doi
1054 fin.entry
1055 }
1056
1057 FUNCTION {inpress}
1058 { output.bibitem
1059   format.authors "author" output.check
1060   journal emphasize "journal" output.check
1061   bbl.inpress output
1062   format.doi
1063   fin.entry
1064 }
1065
1066 FUNCTION {inproceedings}
1067 { output.bibitem
1068   format.authors "author" output.check
1069 <rsc> booktitle format.title.noemph "booktitle" output.check
1070 <angew> booktitle format.title "booktitle" output.check
1071   address output
1072   format.date "year" output.check
1073   pages empty$
1074   'skip$
1075   { format.pages output }
1076   if$
1077   format.doi
1078   fin.entry
1079 }
1080
1081 FUNCTION {manual}
1082 { output.bibitem
1083   author empty$
1084   { organization empty$
1085     'skip$

```

```

1086         { organization output
1087           address output
1088         }
1089         if$
1090     }
1091     { format.authors output }
1092     if$
1093 <rsc> title format.title "title" output.check
1094 <angew> title format.title.noemph "title" output.check
1095     author empty$
1096     { organization empty$
1097       { address output }
1098       'skip$
1099     if$
1100     }
1101     { organization output
1102       address output
1103     }
1104     if$
1105     format.edition output
1106     format.date output
1107     format.doi
1108     fin.entry
1109 }
1110
1111 FUNCTION {mastersthesis}
1112 { output.bibitem
1113   format.authors "author" output.check
1114 <rsc> bbl.msc emphasize format.thesis.type output
1115 <angew> bbl.msc format.thesis.type output
1116   school "school" output.check
1117   address output
1118   format.date "year" output.check
1119   format.doi
1120   fin.entry
1121 }
1122
1123 FUNCTION {misc}
1124 { output.bibitem
1125   format.authors output
1126   title empty$
1127   'skip$
1128   { title format.title output }
1129   if$
1130   howpublished output
1131 <rsc> year output
1132 <angew> format.date output
1133   format.url output
1134   note output
1135   format.doi
1136   fin.entry
1137   empty.misc.check
1138 }
1139

```



```

1140 FUNCTION {patent}
1141 { output.bibitem
1142   organization empty$
1143   { format.authors "author and organization" output.check }
1144   { author empty$
1145     { organization }
1146     { format.authors organization paren space.connect }
1147     if$
1148     "author and organization" output.check
1149   }
1150   if$
1151   journal emphasize "journal" output.check
1152   format.pages.patent "pages" output.check
1153   format.date "year" output.check
1154   format.doi
1155   fin.entry
1156 }
1157
1158 FUNCTION {phdthesis}
1159 { output.bibitem
1160   format.authors "author" output.check
1161   (rsc) bbl.phd emphasize format.thesis.type output
1162   (angew) bbl.phd format.thesis.type output
1163   school "school" output.check
1164   address output
1165   format.date "year" output.check
1166   format.doi
1167   fin.entry
1168 }
1169
1170 FUNCTION {proceedings}
1171 { output.bibitem
1172   (rsc) title format.title.noemph "title" output.check
1173   (angew) title format.title "title" output.check
1174   address output
1175   format.date "year" output.check
1176   pages empty$
1177   'skip$
1178   { format.pages output }
1179   if$
1180   format.doi
1181   fin.entry
1182 }
1183
1184 FUNCTION {techreport}
1185 { output.bibitem
1186   format.authors "author" output.check
1187   title format.title "title" output.check
1188   institution
1189   type empty$
1190   'bbl.techreport
1191   'type
1192   if$
1193   space.connect

```

```

1194 number empty$
1195 { "t" change.case$ }
1196 { number tie.or.space.connect }
1197 if$
1198 output
1199 format.pub.address output
1200 format.date "year" output.check
1201 format.doi
1202 fin.entry
1203 }
1204
1205 FUNCTION {unpublished}
1206 { output.bibitem
1207   format.authors "author" output.check
1208   journal empty$
1209   'skip$
1210   { journal emphasize "journal" output.check }
1211   if$
1212   doi empty$
1213   { note output }
1214   {
1215     (rsc) format.date output
1216     bbl.doi doi tie.or.space.connect output
1217   }
1218   if$
1219   fin.entry
1220   empty.doi.note
1221 }
1222

```

Using the same mechanism as in *biblatex*, control of output can be achieved using a special entry type. The auxiliary macros come from *biblatex* and *Tame the Beast*.

```

1223 INTEGERS { a b }
1224
1225 FUNCTION {mult}
1226 { 'a :=
1227   'b :=
1228   b #0 <
1229     { #-1 #0 b - 'b := }
1230     { #1 }
1231   if$
1232   #0
1233   { b #0 > }
1234   { a +
1235     b #1 - 'b :=
1236   }
1237   while$
1238   swap$
1239   'skip$
1240   { #0 swap$ - }
1241   if$
1242 }
1243
1244 FUNCTION {str.to.int.aux}

```

```

1245 { {duplicate$ empty$ not}
1246   { swap$ #10 mult 'a :=
1247     duplicate$ #1 #1 substring$
1248     chr.to.value a +
1249     swap$
1250     #2 global.max$ substring$
1251   }
1252   while$
1253   pop$
1254 }
1255
1256 FUNCTION {str.to.int}
1257 { duplicate$ #1 #1 substring$ "-" =
1258   {#1 swap$ #2 global.max$ substring$}
1259   {#0 swap$}
1260   if$
1261   #0 swap$ str.to.int.aux
1262   swap$
1263   {#0 swap$ -}
1264   {}
1265   if$
1266 }
1267
1268 FUNCTION {yes.no.to.int}
1269 { "1" change.case$ duplicate$
1270   "yes" =
1271   { pop$ #1 }
1272   { duplicate$ "no" =
1273     { pop$ #0 }
1274     { "unknown Boolean " quote$ * swap$ * quote$ *
1275       " in " * cite$ * warning$
1276       #0
1277     }
1278     if$
1279   }
1280   if$
1281 }
1282
1283 FUNCTION {Control}
1284 { ctrl-use-title
1285   empty$
1286   { skip$ }
1287   { ctrl-use-title
1288     yes.no.to.int
1289     'is.use.title := }
1290   if$
1291   ctrl-et al-number
1292   empty$
1293   { skip$ }
1294   { ctrl-et al-number
1295     str.to.int
1296     'etal.number := }
1297   if$
1298   ctrl-use-doi-all

```

```

1299 empty$
1300 { skip$ }
1301 { ctrl-use-doi-all
1302   yes.no.to.int
1303   'use.doi.all := }
1304 if$
1305 }
1306
1307 FUNCTION {conference} {inproceedings}
1308
1309 FUNCTION {other} {patent}
1310
1311 FUNCTION {default.type} {misc}
1312
1313 MACRO {jan} {"January"}
1314 MACRO {feb} {"February"}
1315 MACRO {mar} {"March"}
1316 MACRO {apr} {"April"}
1317 MACRO {may} {"May"}
1318 MACRO {jun} {"June"}
1319 MACRO {jul} {"July"}
1320 MACRO {aug} {"August"}
1321 MACRO {sep} {"September"}
1322 MACRO {oct} {"October"}
1323 MACRO {nov} {"November"}
1324 MACRO {dec} {"December"}
1325
1326 READ
1327

```

Set up the special control entries.

```

1328 FUNCTION {initialize.controls}
1329 { default.is.use.title 'is.use.title :=
1330   default.etal.number 'etal.number :=
1331   default.use.doi.all 'use.doi.all :=
1332 }
1333
1334 INTEGERS { len }
1335
1336 FUNCTION {chop.word}
1337 { 's :=
1338   'len :=
1339   s #1 len substring$ =
1340   { s len #1 + global.max$ substring$ }
1341   's
1342   if$
1343 }
1344
1345 FUNCTION {format.lab.names}
1346 { 's :=
1347   s #1 "{vv~}{ll}" format.name$
1348   s num.names$ duplicate$
1349   #2 >
1350   { pop$ bbl.etal space.connect }
1351   { #2 <

```

```

1352         'skip$
1353         { s #2 "{ff }{vv }{ll}{ jj}" format.name$ "others" =
1354             { bbl.etal space.connect }
1355             { bbl.and space.connect s #2 "{vv~}{ll}"
1356               format.name$ space.connect }
1357         if$
1358     }
1359     if$
1360 }
1361 if$
1362 }
1363
1364 FUNCTION {author.key.label}
1365 { author empty$
1366     { key empty$
1367         { cite$ #1 #3 substring$ }
1368         'key
1369         if$
1370     }
1371     { author format.lab.names }
1372     if$
1373 }
1374
1375 FUNCTION {author.editor.key.label}
1376 { author empty$
1377     { editor empty$
1378         { key empty$
1379             { cite$ #1 #3 substring$ }
1380             'key
1381             if$
1382         }
1383         { editor format.lab.names }
1384         if$
1385     }
1386     { author format.lab.names }
1387     if$
1388 }
1389
1390 FUNCTION {author.key.organization.label}
1391 { author empty$
1392     { key empty$
1393         { organization empty$
1394             { cite$ #1 #3 substring$ }
1395             { "The " #4 organization chop.word #3
1396               text.prefix$ }
1397             if$
1398         }
1399         'key
1400         if$
1401     }
1402     { author format.lab.names }
1403     if$
1404 }
1405

```

```

1406 FUNCTION {editor.key.organization.label}
1407 { editor empty$
1408   { key empty$
1409     { organization empty$
1410       { cite$ #1 #3 substring$ }
1411       { "The " #4 organization chop.word #3
1412         text.prefix$ }
1413       if$
1414     }
1415     'key
1416   if$
1417 }
1418 { editor format.lab.names }
1419 if$
1420 }
1421
1422 FUNCTION {calc.short.authors}
1423 { type$ "book" =
1424   type$ "inbook" =
1425   or
1426   'author.editor.key.label
1427   { type$ "proceedings" =
1428     'editor.key.organization.label
1429     { type$ "manual" =
1430       'author.key.organization.label
1431       'author.key.label
1432     if$
1433   }
1434   if$
1435 }
1436 if$
1437 'short.list :=
1438 }
1439
1440 FUNCTION {calc.label}
1441 { calc.short.authors
1442   short.list
1443   "("
1444   *
1445   year duplicate$ empty$
1446   short.list key field.or.null = or
1447   { pop$ "" }
1448   'skip$
1449   if$
1450   *
1451   'label :=
1452 }
1453
1454 ITERATE {calc.label}
1455
1456 STRINGS { longest.label last.label next.extra }
1457
1458 INTEGERS { longest.label.width last.extra.num number.label }
1459

```

```

1460 FUNCTION {initialize.longest.label}
1461 { "" 'longest.label :=
1462   #0 int.to.chr$ 'last.label :=
1463   "" 'next.extra :=
1464   #0 'longest.label.width :=
1465   #0 'last.extra.num :=
1466   #0 'number.label :=
1467 }
1468
1469 FUNCTION {forward.pass}
1470 { last.label label =
1471   { last.extra.num #1 + 'last.extra.num :=
1472     last.extra.num int.to.chr$ 'extra.label :=
1473   }
1474   { "a" chr.to.int$ 'last.extra.num :=
1475     "" 'extra.label :=
1476     label 'last.label :=
1477   }
1478   if$
1479   number.label #1 + 'number.label :=
1480 }
1481
1482 EXECUTE {initialize.longest.label}
1483
1484 ITERATE {forward.pass}
1485
1486 FUNCTION {begin.bib}
1487 { preamble$ empty$
1488   'skip$
1489   { preamble$ write$ newline$ }
1490   if$
1491   "\ifx\mcitethebibliography\mciteundefinedmacro"
1492   write$ newline$
1493   " \PackageError"
1494   <rsc> "{rsc.bst}"
1495   <angew> "{angew.bst}"
1496   "{mciteplus.sty has not been loaded}" * *
1497   write$ newline$
1498   " {This bibstyle requires the use of the"
1499   " mciteplus package.}" *
1500   write$ newline$
1501   "\fi"
1502   write$ newline$
1503   "\begin{mcitethebibliography}{ "
1504   number.label int.to.str$ * "}" *
1505   write$ newline$
1506   "\providecommand{\natexlab}[1]{#1}"
1507   write$ newline$
1508   "\mciteSetBstSublistMode{f}"
1509   write$ newline$
1510   "\mciteSetBstMaxWidthForm{subitem}"
1511   write$ newline$
1512   <rsc> "{ (\emph{\alph{mcitesubitemcount}}) }"
1513   <angew> "{ \alph{mcitesubitemcount}} "

```

```

1514 write$ newline$
1515 "\mciteSetBstSublistLabelBeginEnd"
1516 write$ newline$
1517 "{\mcitemaxwidthsubitemform\space}"
1518 write$ newline$
1519 "{\relax}{\relax}"
1520 write$ newline$
1521 }
1522
1523 EXECUTE {begin.bib}
1524
1525 EXECUTE {init.state.consts}
1526
1527 ITERATE {call.type$}
1528
1529 FUNCTION {end.bib}
1530 { newline$
1531   "\end{mcitethebibliography}" write$ newline$
1532 }
1533
1534 EXECUTE {end.bib}

```

6 Change History

v1.0	v2.0
General: Initial release of packaged	General: First re-write of package . 1
rsc.bst 1	v3.0
	General: Second re-write of package 1

7 References

- [1] C. D. Abernethy, G. M. Codd, M. D. Spicer and M. K. Taylor, *J. Am. Chem. Soc.*, 2003, **125**, 1128–1129.
- [2] F. A. Cotton, G. Wilkinson, C. A. Murillio and M. Bochmann, *Advanced Inorganic Chemistry*, Wiley, Chichester, 6th edn., 1999.
- [3] (a) A. J. Arduengo, III, H. V. R. Dias, R. L. Harlow and M. Kline, *J. Am. Chem. Soc.*, 1992, **114**, 5530–5534; (b) A. J. Arduengo, III, S. F. Gamper, J. C. Calabrese and F. Davidson, 1994, **116**, 4391–4394.