

The ACM Transactions (`acmtrans`) bibliography style

Glenn Paulley Andrew W. Appel Rebecca L. Davies
 David M. Jones

30 October 1997
Version 1.5 β

Contents

1	Introduction	1
2	Further features of <code>acmtrans</code>	2
3	Modifications and bug fixes from <code>newapa.bst</code>	3
4	To-do list	3
5	The <code>docstrip</code> modules	4
6	The macros	4
6.1	The field names and useful utilities	4
6.2	The entry types	22
6.2.1	The <code>article</code> type	22
6.2.2	The <code>book</code> type	22
6.2.3	The <code>booklet</code> type	23
6.2.4	The <code>inbook</code> type	23
6.2.5	The <code>incollection</code> type	24
6.2.6	The <code>inproceedings</code> type	25
6.2.7	The <code>conference</code> type	27
6.2.8	The <code>manual</code> type	27
6.2.9	The <code>mastersthesis</code> type	27
6.2.10	The <code>misc</code> type	28
6.2.11	The <code>phdthesis</code> type	28
6.2.12	The <code>proceedings</code> type	28
6.2.13	The <code>techreport</code> type	29

©1995, all rights reserved. Copying of this file is authorized only if either (1) you make absolutely no changes to your copy, including name, or (2) if you do make changes, you name it something other than “`acmtrans.bst`”. There are undoubtedly bugs in this style. Please send any bug reports, suggestions for improvements, etc., to `jacm@theory.lcs.mit.edu`.

6.2.14	The <code>unpublished</code> type	29
6.2.15	The <code>default.type</code> type	29
6.3	Standard abbreviations	30
6.4	Sorting the entries	31

1 Introduction

This is the `acmtrans` \LaTeX style, which is based Glenn Paulley’s `chicago` bibliography style (for further information about the history of this code, please see the comments in `chicago.bst`). The `acmtrans` style was hacked by Andrew W. Appel and Rebecca L. Davies of Princeton University, and further modified and reworked by David M. Jones.

This style implements an “author-year” bibliography format, where in-text citations have the general form “[author name(s) year]” and the Reference list is sorted alphabetically by the author or whatever passes for author in the absence of one. The style also supports abbreviated author lists in citations.

This style must be supplemented by some \LaTeX macros to process the citation information generated by `acmtrans`. The citation information deposited in the bibliography has the following format:

```
\citeauthoryear{\full-author-info}{\abbrev-author-info}{\year}
```

The \LaTeX style should define something like the following:

```
\let\@internalcite\cite
\def\fullcite{\def\citeauthoryear##1##2##3{##1 ##3}\@internalcite}
\def\fullciteA{\def\citeauthoryear##1##2##3{##1}\@internalcite}
\def\shortcite{\def\citeauthoryear##1##2##3{##2 ##3}\@internalcite}
\def\shortciteA{\def\citeauthoryear##1##2##3{##2}\@internalcite}
\def\citeyear{\def\citeauthoryear##1##2##3{##3}\@internalcite}
```

Additional commands to manipulate different components of a citation can be defined so that, for example, you can list author’s names without parentheses if using a citation as a noun or object in a sentence. See the `acmart` document class for a more complete example.

In order to provide more flexibility for use with both \LaTeX 2.09 and \LaTeX 2 ϵ , `acmtrans` uses the following macros for all font changes:

`\bibsc` for small caps

`\bibemph` for italics without an italic correction

`\bibemphic` for italics with an italic correction at the end.

For \LaTeX 2 ϵ , the following is appropriate:

```
\let\bibsc\textsc
\let\bibemph\emph
\let\bibemphic\emph
```

For \LaTeX 2.09, the following should do:

```
\newcommand{\bibsc}[1]{\sc#1}
\newcommand{\bibemph}[1]{\em#1}
\newcommand{\bibemphic}[1]{\em#1/}
```

One further macro, `\bibyear`, is needed, to format the year. The following definition is used in the `acmart` class. See the comments before `output.year.check` below and before `\bibyear` in `acmart` for more information.

```
\newcommand{\bibyear}[2]{%
  \unskip\quad\ignorespaces#1\unskip
  \if#2. .\quad \else \quad#2 \fi
}
```

These TeX macro definitions are found in the `acmart` document class.

2 Further features of `acmtrans`

- full names used in citations, but abbreviated citations are available (see above)
- if an entry has a “month”, then the month and year are also printed as part of that bibitem.
- all conjunctions use “and” instead of “&”
- major modification from Chicago Manual of Style (13th ed.) is that only the first author in a reference appears last name first- additional authors appear as J. Q. Public.
- pages are listed as “pp. xx-xx” in all entry types except article entries.
- book, inbook, and manual use “location: publisher” (or organization) for address and publisher. All other types list publishers separately.
- “pp.” are used to identify page numbers for all entry types except articles.
- organization is used as a citation label if neither author nor editor is present (for manuals).
- “et al.” is used for long author and editor lists, or when “others” is used.

3 Modifications and bug fixes from `newapa.bst`

- added month, year to bib entries if month is present
- fixed bug with In proceedings, added necessary comma after title
- all conjunctions changed to “and” from “&”
- fixed bug with author labels in `my.full.label`: “et al.” now is generated when “others” is an author name
- major modification from Chicago Manual of Style (13th ed.) is that only the first author in a reference appears last name first—additional authors appear as J. Q. Public.
- pages are listed as “pp. xx-xx” in all entry types except article entries. Unnecessary (IMHO) “()” around page numbers were removed, and page numbers now don’t end with a period.

- created `chicago.sty` for use with this bibstyle (required).
- fixed bugs in `format.vol.num.pages` for missing volume, number, and/or pages. Renamed to `format.jour.vol`.
- fixed bug in formatting booktitles: additional period an error if book has a volume.
- fixed bug: editors usually given redundant period before next clause (`format.editors.dot`) removed.
- added label support for organizations, if both author and editor are missing (from `alpha.bst`). If organization is too long, then the key field is used for abbreviated citations.
- In proceedings or books of several volumes, no comma was written between the “Volume x” and the page numbers (this was intentional in `newapa.bst`). Fixed.
- Some journals may not have volumes/numbers, only month/year (eg. IEEE Computer). Fixed bug in article style that assumed volume/number was always present.

4 To-do list

Here are a few of the many things that need to be done to finish this style file.

- Figure out what, if any, copyright notice should cover this file.
- Decide what parts of the above documentation to keep. Sort out credit issues.
- Get style rules from ACM and implement them.
- Implement `format.proc.publisher` and sort out other issues mentioned in connection with `inproceedings`.
- Sort out `format.key` business.
- Munge `month` to standardize month abbreviations when authors fail to use the standard strings.
- Sort out journal abbreviations.
- Make `.bb1` file look more like what the ACM wants for its automatic conversion programs. (In fact, we could output SGML code directly for them, if they’d tell us what they want.)
- Rewrite `macrocode` to index BST macros.

5 The docstrip modules

This code uses the following modules:

<code>xref</code>	use crossrefs in the bibliography
<code>driver</code>	generate a driver file for the documentation

6 The macros

6.1 The field names and useful utilities

```
1 ENTRY
2 { address
3   author
4   booktitle
5   chapter
6   edition
7   editor
8   howpublished
9   institution
10  journal
11  key
12  month
13  note
14  number
15  organization
16  pages
17  publisher
```

In addition to the standard fields, we introduce three new fields, `pub-year`, `pub-month` and `pub-address`. These are for use in proceedings, where the `year`, `month` and `address` fields normally refer to the date and location of the conference rather than the publisher. For compatibility with the standard `BIBTEX` styles, we introduce new fields for the publisher information rather than the conference information, although it would probably be more logical for the existing fields always to refer to the publisher and to introduce new fields for the conference information.

```
18  pub-year
19  pub-month
20  pub-address
21  school
22  series
23  title
24  type
25  volume
26  year
27 }
28 {}
29 { label.year extra.label sort.year sort.label }
30
31 INTEGERS {
32   output.state
33   before.all
34   mid.sentence
35   after.sentence
36   after.block
37 }
```

`init.state.consts`

```
38 FUNCTION {init.state.consts}
39 { #0 'before.all :=
40   #1 'mid.sentence :=
41   #2 'after.sentence :=
```

```

42 #3 'after.block :=
43 }

```

output.nonnull

```

44 STRINGS { s t u }
45
46 FUNCTION {output.nonnull}
47 { 's :=
48   output.state mid.sentence =
49   { ", " * write$ }
50   { output.state after.block =
51     { add.period$ write$
52       newline$
53       "\newblock " write$
54     }
55     { output.state before.all =
56       'write$
57       { add.period$ " " * write$ }
58     if$
59   }
60   if$
61   mid.sentence 'output.state :=
62   }
63 if$
64 s
65 }

```

output.nonnull.colon Use a colon to separate output. Used only for address/publisher combination in book/inbook types, address/institution for manuals, and organization:publisher for proceedings (inproceedings).

```

66 FUNCTION {output.nonnull.colon}
67 { 's :=
68   output.state mid.sentence =
69   { ": " * write$ }
70   { output.state after.block =
71     { add.period$ write$
72       newline$
73       "\newblock " write$
74     }
75     { output.state before.all =
76       'write$
77       { add.period$ " " * write$ }
78     if$
79   }
80   if$
81   mid.sentence 'output.state :=
82   }
83 if$
84 s
85 }

```

output

```

86 FUNCTION {output}
87 { duplicate$ empty$

```

```

88     'pop$
89     'output.nonnull
90     if$
91 }

```

output.colon

```

92 FUNCTION {output.colon}
93 { duplicate$ empty$
94     'pop$
95     'output.nonnull.colon
96     if$
97 }

```

warn.null Issue a warning message if a field is empty.

```

98 FUNCTION {warn.null}
99 {
100     "empty " swap$ * " in " * cite$ * warning$
101 }

```

output.check

```

102 FUNCTION {output.check}
103 {
104     't :=
105     duplicate$
106     empty$
107     { pop$ t warn.null }
108     'output.nonnull
109     if$
110 }

```

field.or.null.check field.or.null.check is similar to field.or.null, but it also issues a warning if the field is null.

```

111 FUNCTION {field.or.null.check}
112 {
113     't :=
114     duplicate$
115     empty$
116     { pop$ "" t warn.null }
117     'skip$
118     if$
119 }

```

output.check.colon

```

120 FUNCTION {output.check.colon}
121 { 't :=
122     duplicate$ empty$
123     { pop$ t warn.null }
124     'output.nonnull.colon
125     if$
126 }

```

new.block

```

127 FUNCTION {new.block}

```

```

128 { output.state before.all =
129     'skip$
130     { after.block 'output.state := }
131   if$
132 }

```

`new.sentence`

```

133 FUNCTION {new.sentence}
134 {
135     output.state after.block =
136     'skip$
137     { output.state before.all =
138         'skip$
139         { after.sentence 'output.state := }
140     }
141   if$
142 }
143 }

```

`output.year.check` This is somewhat nasty. We need to put a quad space around the year. Unfortunately, the obvious thing (i.e., putting out “`\quad<year>\quad`”) won’t work because the next call to `output` will add a period *after* the second `\quad`. Because of the way `BIBTEX` writes things, it would be difficult to get the `\quad` in the right place, so instead we cheat by wrapping the year in the macro `\bibyear`, which is defined in `acmart` to look ahead one character and, if necessary, move the period before the second quad space.

This should be thought about some more.

```

144 FUNCTION {output.year.check}
145 { year empty$
146     { "year" warn.null }
147     { add.period$ write$
148         " \bibyear{" year * extra.label * "}" *
149         mid.sentence 'output.state :=
150     }
151   if$
152 }

```

`fin.entry`

```

153 FUNCTION {fin.entry}
154 { add.period$
155   write$
156   newline$
157 }

```

`not`

```

158 FUNCTION {not}
159 { { #0 }
160   { #1 }
161   if$
162 }

```

`and`

```

163 FUNCTION {and}

```



```
164 { 'skip$
165   { pop$ #0 }
166   if$
167 }
```

or

```
168 FUNCTION {or}
169 { { pop$ #1 }
170   'skip$
171   if$
172 }
```

new.block.checka

```
173 FUNCTION {new.block.checka}
174 { empty$
175   'skip$
176   'new.block
177   if$
178 }
```

new.block.checkb

```
179 FUNCTION {new.block.checkb}
180 { empty$
181   swap$ empty$
182   and
183   'skip$
184   'new.block
185   if$
186 }
```

new.sentence.checka

```
187 FUNCTION {new.sentence.checka}
188 { empty$
189   'skip$
190   'new.sentence
191   if$
192 }
```

new.sentence.checkb

```
193 FUNCTION {new.sentence.checkb}
194 { empty$
195   swap$ empty$
196   and
197   'skip$
198   'new.sentence
199   if$
200 }
```

field.or.null

```
201 FUNCTION {field.or.null}
202 { duplicate$ empty$
203   { pop$ "" }
204   'skip$
205   if$
206 }
```

`parenthesize` Put parentheses around the top string on the stack.

```
207 FUNCTION {parenthesize}
208 { duplicate$ empty$
209   { pop$ "" }
210   { " (" swap$ * ")" * }
211   if$
212 }
```

`emphasize` Emphasize the top string on the stack.

It's a shame that we can't assume people are using L^AT_EX 2_ε, since the `\emph` command would be much more convenient here. As it is, we have to try to guess whether or not we will need an italic correction, and use either `emphasize` or `emphasize.ic` as appropriate. As a partial solution, we use the macros `\bibemph` and `\bibemphic`, which should be defined by the style file. For L^AT_EX 2_ε, these can both be equivalent to `\emph`; for L^AT_EX 2.09, `\bibemphic` should add an italic correction and `\bibemph` should not.

```
213 FUNCTION {emphasize}
214 {
215   duplicate$ empty$
216   { pop$ "" }
217   { "\bibemph{" swap$ * "}" * }
218   if$
219 }
```

`emphasize.ic` Emphasize the top string on the stack, and add an italic correction.

```
220 FUNCTION {emphasize.ic}
221 {
222   duplicate$ empty$
223   { pop$ "" }
224   { "\bibemphic{" swap$ * "}" * }
225   if$
226 }
```

`smallcaps` Put the top string on the stack in small caps.

```
227 FUNCTION {smallcaps}
228 { duplicate$ empty$
229   { pop$ "" }
230   { "\bibsc{" swap$ * "}" * }
231   if$
232 }
```

`format.names` Format bibliographical entries with the last name first. All names are formatted in this routine.

Also moved “Junior” part to the very end, so that it appears after the first initial instead of before.

This function no longer automatically puts its output in smallcaps; that's left to the functions that call `format.names`, so we can use `format.names` to format “internal” editor names. – dmj, 4/28/96

```
233 INTEGERS { nameptr namesleft numnames }
234
235 FUNCTION {format.names}
236 { 's :=
```

```

237 #1 'nameptr := % nameptr = 1;
238 s num.names$ 'numnames := % numnames = num.name$(s);
239 numnames 'namesleft :=
240 { namesleft #0 > }
241 {
242 % nameptr #1 =
243 % {s nameptr "{vv~}{ll}{, f.}{, jj}" format.name$ 't := } {
244 s nameptr "{vv~}{ll}{, f.}{, jj}" format.name$ 't :=
245 % } if$
246 nameptr #1 >
247 { namesleft #1 >
248 { ", " * t * }
249 { numnames #2 >
250 { ", " * }
251 'skip$
252 if$
253 t "others" =
254 { " et~al." * }
255 { " and " * t * } % from Chicago Manual of Style
256 if$
257 }
258 if$
259 }
260 't
261 if$
262 nameptr #1 + 'nameptr := % nameptr += 1;
263 namesleft #1 - 'namesleft := % namesleft -= 1;
264 }
265 while$
266 }

```

my.full.label

```

267 FUNCTION {my.full.label}
268 { 's :=
269 #1 'nameptr := % nameptr = 1;
270 s num.names$ 'numnames := % numnames = num.name$(s);
271 numnames 'namesleft :=
272 { namesleft #0 > }
273 { s nameptr "{vv~}{ll}" format.name$ 't := % get the next name
274 nameptr #1 >
275 { namesleft #1 >
276 { ", " * t * }
277 { numnames #2 >
278 { ", " * }
279 'skip$
280 if$
281 t "others" =
282 { " et~al." * }
283 { " and " * t * } % from Chicago Manual of Style
284 if$
285 }
286 if$
287 }
288 't

```

```

289     if$
290     nameptr #1 + 'nameptr :=          % nameptr += 1;
291     namesleft #1 - 'namesleft :=     % namesleft -= 1;
292   }
293 while$
294 }

```

`format.names.fml` Format names in “familiar” format, with first initial followed by last name. Like `format.names`, ALL names are formatted.

Removed small caps from `format.names.fml` since it’s used for “internal” editor names, which should be in ordinary text. – dmj, 4/28/96

```

295 FUNCTION {format.names.fml}
296 { 's :=
297   #1 'nameptr :=          % nameptr = 1;
298   s num.names$ 'numnames := % numnames = num.name$(s);
299   numnames 'namesleft :=
300   { namesleft #0 > }
301   { s nameptr "{f.~}{vv~}{ll}{, ,jj}" format.name$ 't :=
302     nameptr #1 >
303     { namesleft #1 >
304       { " , " * t * }
305       { numnames #2 >
306         { " , " * }
307         'skip$
308         if$
309         t "others" =
310           { " et~al." * }
311           { " and " * t * }
312         if$
313       }
314     }
315   }
316   't
317   if$
318   nameptr #1 + 'nameptr :=          % nameptr += 1;
319   namesleft #1 - 'namesleft :=     % namesleft -= 1;
320 }
321 while$
322 }

```

`format.authors` Added `smallcaps`, which used to be inside `format.names`. – dmj, 4/28/96

```

323 FUNCTION {format.authors}
324 { author empty$
325   { "" }
326   { author format.names smallcaps }
327 if$
328 }

```

`format.key`

```

329 FUNCTION {format.key}
330 { empty$
331   { key field.or.null }
332   { "" }
333 if$

```

334 }

`format.editors.fml` Format editor names for use in the “in” types: inbook, incollection, inproceedings: first initial, then last names. When editors are the LABEL for an entry, then `format.editor` is used which lists editors by last name first.

Removed parens around “Ed.” since that seems to be desired by the ACM.
[8/18/95—dmj]

```
335 FUNCTION {format.editors.fml}
336 { editor empty$
337   { "" }
338   { editor format.names.fml
339     editor num.names$ #1 >
340     { ", Eds." * }
341     { ", Ed." * }
342     if$
343   }
344   if$
345 }
```

`format.editors` Format editor names for use in labels, last names first.

Removed parens around “Ed.” since that seems to be desired by the ACM.
[8/18/95—dmj]

```
346 FUNCTION {format.editors}
347 { editor empty$
348   { "" }
349   { editor format.names smallcaps
350     editor num.names$ #1 >
351     { ", Eds." * }
352     { ", Ed." * }
353     if$
354   }
355   if$
356 }
```

`format.title`

```
357 FUNCTION {format.title}
358 { title empty$
359   { "" }
360   { title "t" change.case$ }
361   if$
362 }
```

`n.dashify`

```
363 FUNCTION {n.dashify}
364 { 't :=
365   ""
366   { t empty$ not }
367   { t #1 #1 substring$ "--" =
368     { t #1 #2 substring$ "---" = not
369       { "---" *
370         t #2 global.max$ substring$ 't :=
371       }
372     { { t #1 #1 substring$ "-" = }
```

```

373     { "-" *
374       t #2 global.max$ substring$ 't :=
375     }
376     while$
377   }
378   if$
379 }
380 { t #1 #1 substring$ *
381   t #2 global.max$ substring$ 't :=
382 }
383   if$
384 }
385 while$
386 }

```

format.edition

```

387 FUNCTION {format.edition}
388 {
389   edition empty$
390   { "" }
391   { edition " ed." * }
392   if$
393 }

```

format.btitle

```

394 FUNCTION {format.btitle}
395 {
396   title emphasize "title" output.check
397   new.sentence
398   format.edition output
399   new.sentence
400 }

```

format.emphasize.booktitle

```

401 FUNCTION {format.emphasize.booktitle}
402 {
403   edition empty$
404   { booktitle emphasize.ic }
405   { booktitle empty$
406     See above.
407     { booktitle emphasize }
408     { booktitle emphasize.ic edition " ed." * parenthesize * }
409     if$
410   }
411 }

```

tie.or.space.connect Enhanced so that if the first string is empty, no extra space is put in. It's not clear that this is really useful, but it shouldn't hurt. – dmj, 4/28/96

```

412 FUNCTION {tie.or.space.connect}
413 {
414   swap$
415   duplicate$

```

```

416     empty$
417     'pop$
418     {
419         swap$
420         duplicate$
421         text.length$ #3 <
422         { "~" }
423         { " " }
424         if$
425         swap$ * *
426     }
427     if$
428 }

```

either.or.check

```

429 FUNCTION {either.or.check}
430 { empty$
431   'pop$
432   { "can't use both " swap$ * " fields in " * cite$ * warning$ }
433   if$
434 }

```

format.bvolume What should be done if there is both a volume and a number in a book? Currently, the number is silently ignored, and only the volume is used. Surely there should at least be some sort of warning message. Besides, if there's both a number and a series, the series gets printed twice. Must sort this out.

```

435 FUNCTION {format.bvolume}
436 {
437     volume empty$
438     { "" }
439     { series emphasize " vol." volume tie.or.space.connect * }
440     if$
441 }

```

format.number.series Should this be "Series no. 5"? If so, we should be able to merge this with format.bvolume.

```

442 FUNCTION {format.number.series}
443 {
444     volume empty$
445     { number empty$
446       { series field.or.null }
447       { series empty$
448         { "there's a number but no series in " cite$ * warning$ }
449         { series " no." number tie.or.space.connect * }
450         if$
451       }
452     }
453     if$
454     { "" }
455     if$
456 }

```

multi.page.check

```

457 INTEGERS { multiresult }
458
459 FUNCTION {multi.page.check}
460 { 't :=
461   #0 'multiresult :=
462   { multiresult not
463     t empty$ not
464     and
465   }
466   { t #1 #1 substring$
467     duplicate$ "-" =
468     swap$ duplicate$ "," =
469     swap$ "+" =
470     or or
471   { #1 'multiresult := }
472   { t #2 global.max$ substring$ 't := }
473   if$
474 }
475 while$
476 multiresult
477 }

```

format.pages gnp - removed ()

Removed "pp." since as far as I can tell, it's not wanted. – dmj, 4/28/96

```

478 FUNCTION {format.pages}
479 {
480   pages empty$
481     { "" }
482     { " " pages n.dashify * }
483   if$
484 }

```

format.jour.vol By Young (and Spencer) GNP - fixed bugs with missing volume, number, and/or pages

Format journal, volume, number, pages for article types.

```

485 FUNCTION {format.jour.vol}
486 {
487   journal "journal" field.or.null.check
488   volume empty$
489     { "volume" warn.null }
490     { volume tie.or.space.connect }
491   if$
492   emphasize.ic output
493   number field.or.null
494   month parenthesize
495   *
496   output
497   format.pages "pages" output.check
498 }

```

format.chapter.pages

```

499 FUNCTION {format.chapter.pages}
500 { chapter empty$

```



```

501 'format.pages
502 { type empty$
503   { "Chapter" } % gnp - changed to mixed case
504   { type "t" change.case$ }
505   if$
506   chapter tie.or.space.connect
507   pages empty$
508   {"page numbers missing in " cite$ * warning$}%gnp - added check
509   { ", " * format.pages * }
510   if$
511   }
512 if$
513 }

```

format.in.ed.booktitle I think this is backwards; editors should come after booktitle.

```

514 FUNCTION {format.in.ed.booktitle}
515 {
516   booktitle empty$
517   { "" }
518   {
519     "In " format.emphasize.booktitle *
520     editor empty$
521     'skip$
522     { ", " * format.editors.fml * }
523     if$
524   }
525   if$
526 }

```

format.thesis.type

```

527 FUNCTION {format.thesis.type}
528 { type empty$
529   'skip$
530   { pop$
531     type "t" change.case$
532   }
533   if$
534 }

```

format.thesis.title

```

535 %% FUNCTION {format.thesis.title}
536 %% {
537 %%   title field.or.null "t" change.case$
538 %% }

```

format.tr.number

```

539 FUNCTION {format.tr.number}
540 { type empty$
541   { "Technical Report" }
542   'type
543   if$
544   number empty$
545   { "t" change.case$ }
546   { number tie.or.space.connect }

```

```
547 if$
548 }
```

`format.crossref` Merged `format.article.crossref` and `format.incoll.inproc.crossref` into a single macro, `format.crossref`, which has proper spacing.

```
549 (*xref)
550 FUNCTION {format.crossref}
551 {
552   "See \citeN{" crossref * "}" *
553 }
554 \xref)
```

`format.crossref.editor`

```
555 (*xref)
556 FUNCTION {format.crossref.editor}
557 { editor #1 "{vv~}{ll}" format.name$
558   editor num.names$ duplicate$
559   #2 >
560   { pop$ " et~al." * }
561   { #2 <
562     'skip$
563     { editor #2 "{ff }{vv }{ll}{ jj}" format.name$ "others" =
564       { " et~al." * }
565       { " and " * editor #2 "{vv~}{ll}" format.name$ * }
566     if$
567   }
568   if$
569 }
570 if$
571 }
572 \xref)
```

`format.book.crossref`

```
573 (*xref)
574 FUNCTION {format.book.crossref}
575 { volume empty$
576   { "empty volume in " cite$ * "'s crossref of " * crossref * warning$
577     "In "
578   }
579   { "Volume" volume tie.or.space.connect % gnp - changed to mixed case
580     " of " *
581   }
582   if$
583   editor empty$
584   editor field.or.null author field.or.null =
585   or
586   { key empty$
587     { series empty$
588       { "need editor, key, or series for " cite$ * " to crossref " *
589         crossref * warning$
590         "" *
591       }
592       { series emphasize.ic }
593     if$
```

```

594   }
595   { key * }
596   if$
597   }
598   { format.crossref.editor * }
599   if$
600   " \citeN{" * crossref * }" *
601 }
602 </xref>

```

`format.lab.names` Determines “short” names for the abbreviated author information. “Long” labels are created in `calc.label`, using the routine `my.full.label` to format author and editor fields.

There are 4 cases for labels ($n = 3$ in the example):

1. one author: Foo
2. one to n : Foo, Bar and Baz
3. use of “and others”: Foo, Bar et al.
4. more than n : Foo et al.

```

603 FUNCTION {format.lab.names}
604 { 's :=
605   s num.names$ 'numnames :=
606   numnames #2 > % change number to number of others allowed before
607                 % forcing "et al".
608   { s #1 "{vv~}{ll}" format.name$ " et~al." * }
609   {
610     numnames #1 - 'namesleft :=
611     #2 'nameptr :=
612     s #1 "{vv~}{ll}" format.name$
613     { namesleft #0 > }
614     { nameptr numnames =
615       { s nameptr "{ff }{vv }{ll}{ jj}" format.name$ "others" =
616         { " et~al." * }
617         { " and " * s nameptr "{vv~}{ll}" format.name$ * }
618         if$
619       }
620       { ", " * s nameptr "{vv~}{ll}" format.name$ * }
621       if$
622       nameptr #1 + 'nameptr :=
623       namesleft #1 - 'namesleft :=
624     }
625     while$
626   }
627   if$
628 }

```

`author.key.label`

```

629 FUNCTION {author.key.label}
630 { author empty$
631   { key empty$
632     { "no key, author in " cite$ * warning$

```

```

633         cite$ #1 #3 substring$ }
634     'key
635     if$
636 }
637 { author format.lab.names }
638 if$
639 }

```

editor.key.label

```

640 FUNCTION {editor.key.label}
641 { editor empty$
642   { key empty$
643     { "no key, editor in " cite$ * warning$
644       cite$ #1 #3 substring$ }
645     'key
646     if$
647   }
648   { editor format.lab.names }
649   if$
650 }

```

author.key.org.label added - gnp. Provide label formatting by organization if author is null.

```

651 FUNCTION {author.key.org.label}
652 { author empty$
653   { organization empty$
654     { key empty$
655       { "no key, author or organization in " cite$ * warning$
656         cite$ #1 #3 substring$ }
657       'key
658       if$
659     }
660     'organization
661     if$
662   }
663   { author format.lab.names }
664   if$
665 }

```

editor.key.org.label added - gnp. Provide label formatting by organization if editor is null.

```

666 FUNCTION {editor.key.org.label}
667 { editor empty$
668   { organization empty$
669     { key empty$
670       { "no key, editor or organization in " cite$ * warning$
671         cite$ #1 #3 substring$ }
672       'key
673       if$
674     }
675     'organization
676     if$
677   }
678   { editor format.lab.names }
679   if$
680 }

```

author.editor.key.label

```
681 FUNCTION {author.editor.key.label}
682 { author empty$
683   { editor empty$
684     { key empty$
685       { "no key, author, or editor in " cite$ * warning$
686         cite$ #1 #3 substring$ }
687       'key
688       if$
689     }
690     { editor format.lab.names }
691   if$
692 }
693 { author format.lab.names }
694 if$
695 }
```

calc.label GNP: changed. Form label for BIB_TE_X entry. The classification of which fields are used for which type of entry (book, inbook, etc.) are taken from alpha.bst. The change here from newapa is to also include organization as a citation label if author or editor is missing. See also author.organization.sort, editor.organization.sort.

```
696 FUNCTION {calc.label}
697 { type$ "book" =
698   type$ "inbook" =
699   or
700   'author.editor.key.label
701   { type$ "proceedings" =
702     'editor.key.org.label
703     { type$ "manual" =
704       'author.key.org.label
705       'author.key.label
706       if$
707     }
708     if$
709   }
710   if$
711   author empty$ % generate the full label citation information.
712   { editor empty$
713     { organization empty$
714       { "no author, editor, or organization in " cite$ * warning$
715         "??" }
716       'organization
717       if$
718     }
719     { editor my.full.label }
720     if$
721   }
722   { author my.full.label }
723   if$

  leave label on the stack, to be popped when required.
724   "}{ " * swap$ * "}{ " *
725   % year field.or.null purify$ #-1 #4 substring$ *
```

```

save the year for sort processing afterwards (adding a, b, c, etc.)
726 year field.or.null purify$ #-1 #4 substring$
727 'label.year :=
728 }

```

output.bibitem

```

729 FUNCTION {output.bibitem}
730 { newline$
731 "\bibitem[\protect\citeauthoryear{" write$
732 calc.label write$
733 sort.year write$
734 "]}{" write$
735 cite$ write$
736 "}" write$
737 newline$
738 ""
739 before.all 'output.state :=
740 }

```

6.2 The entry types

6.2.1 The article type

article

```

741 FUNCTION {article}
742 {
743   output.bibitem
744   format.authors "author" output.check
745   author format.key output           % added
746   output.year.check                   % added
747   new.block
748   format.title "title" output.check
749   new.block
750   (*xref)
751   crossref missing$
752   {
753     \xref)
754     format.jour.vol
755   (*xref)
756   }
757   {
758     format.crossref output.nonnull
759     format.pages output
760   }
761   if$
762   \xref)
763   new.block
764   note output
765   fin.entry
766 }

```

Some but not all types contain the following line. It's questionable whether it belongs there at all, but if it does, shouldn't it be in *all* the types?

6.2.2 The book type

book

```
767 FUNCTION {book}
768 {
769     output.bibitem
770     author empty$
771     { format.editors "author and editor" output.check }
772     {
773         format.authors output.nonnull
774 < *xref >
775         crossref missing$
776         {
777 < /xref >
778             "author and editor" editor either.or.check
779 < *xref >
780             }
781             'skip$
782             if$
783 < /xref >
784         }
785         if$
786         output.year.check      % added
787         new.block
788         format.btitle % "title" output.check
789 < *xref >
790         crossref missing$
791         {
792 < /xref >
793             format.bvolume output
794             new.block
795             format.number.series output
796             new.sentence
797             publisher "publisher" output.check
798             address output
799 < *xref >
800             }
801             {
802                 new.block
803                 format.book.crossref output.nonnull
804             }
805             if$
806 < /xref >
807         new.block
808         note output
809         fin.entry
810 }
```

6.2.3 The booklet type

booklet

```

811 FUNCTION {booklet}
812 { output.bibitem
813   format.authors output
814   author format.key output          % added
815   output.year.check                 % added
816   new.block
817   format.title "title" output.check
818   new.block
819   howpublished output
820   address output
821   new.block
822   note output
823   fin.entry
824 }

```

6.2.4 The inbook type

inbook

```

825 FUNCTION {inbook}
826 {
827   output.bibitem
828   author empty$
829   { format.editors "author and editor" output.check }
830   { format.authors output.nonnull
831     crossref missing$
832     { "author and editor" editor either.or.check }
833     'skip$
834     if$
835   }
836   if$
837   output.year.check          % added
838   new.block
839   format.btitle % "title" output.check
840 <*xref>
841   crossref missing$
842   {
843 </xref>
844     format.bvolume output
845     format.chapter.pages "chapter and pages" output.check
846     new.block
847     format.number.series output
848     new.sentence
849     publisher "publisher" output.check.colon
850     address output
851 <*xref>
852   }
853   {
854     format.chapter.pages "chapter and pages" output.check
855     new.block
856     format.book.crossref output.nonnull
857   }
858   if$
859 </xref>
860   new.block

```



```

861     note output
862     fin.entry
863 }

```

6.2.5 The incollection type

incollection

```

864 FUNCTION {incollection}
865 { output.bibitem
866   format.authors "author" output.check
867   author format.key output      % added
868   output.year.check             % added
869   new.block
870   format.title "title" output.check
871   new.block
872   < *xref >
873   crossref missing$
874   {
875     < /xref >
876     format.in.ed.booktitle "booktitle" output.check
877     format.bvolume output
878     format.number.series output
879     format.chapter.pages output % gnp - was special.output.nonnull
880 %                               left out comma before page numbers
881     new.sentence
882     address output
883     publisher "publisher" output.check.colon
884   < *xref >
885   }
886   {
887     format.crossref output.nonnull
888     format.chapter.pages output
889   }
890   if$
891 < /xref >
892   new.block
893   note output
894   fin.entry
895 }

```

6.2.6 The inproceedings type

format.date

```

896 FUNCTION {format.date}
897 { year empty$
898   { month empty$
899     { "" }
900     { "there's a month but no year in " cite$ * warning$
901       month
902     }
903   } if$
904 }
905 { month empty$

```

```

906         'year
907         { month " " * year * }
908     if$
909 }
910 if$
911 }

```

format.proc.loc.date

```

912 FUNCTION {format.proc.loc.date}
913 {
914     address field.or.null
915     format.date
916     duplicate$ empty$
917     'pop$
918     { swap$
919         duplicate$ empty$
920         'pop$
921         { ", " * swap$ * }
922         if$
923     }
924     if$
925     parenthesize
926     *
927 }

```

inproceedings Added `format.proc.loc.date` to format proceedings location and date. Should also add something like `format.proc.publisher` to handle the `pub-year`, `pub-month` and `pub-address` fields, but should first find out from ACM how such things should be handled. In particular, if year of publication is different from year of conference, which should be used in citation (presumably the year of publication).

A related problem is to decide exactly what the `year` refers to. In the standard styles, if there is a `year` but no `address`, the year is assumed to refer to the date of publication, while if there is an `address`, the address and date are assumed to refer to the conference. For true compatibility with the standard styles, we should probably preserve this distinction.

```

928 FUNCTION {inproceedings}
929 {
930     output.bibitem
931     format.authors "author" output.check
932     author format.key output          % added
933     output.year.check                  % added
934     new.block
935     format.title "title" output.check
936     new.block
937     <*/xref>
938     crossref missing$
939     {
940     </xref>
941         format.in.ed.booktitle "booktitle" output.check
942         format.bvolume output
943         format.number.series output
944     %     address output
945         format.proc.loc.date %output

```

```

946 %      format.pages output
947      new.sentence
948      organization output
949      publisher output %.colon
950      format.pages output
951 < *xref>
952     }
953     {
954       format.crossref output.nonnull
955       format.pages output
956     }
957   if$
958 < /xref>
959   new.block
960   note output
961   fin.entry
962 }

```

6.2.7 The conference type

conference

```

963 FUNCTION {conference} { inproceedings }

```

6.2.8 The manual type

manual

```

964 FUNCTION {manual}
965 { output.bibitem
966   author empty$
967   { editor empty$
968     { organization "organization" output.check
969       organization format.key output } % if all else fails, use key
970     { format.editors "author and editor" output.check }
971     if$
972   }
973   { format.authors output.nonnull }
974   if$
975   output.year.check          % added
976   new.block
977   format.btitle % "title" output.check
978   organization address new.block.checkb

  Reversed the order of "address" and "organization", added the ":".

979   address output
980   organization "organization" output.check.colon
981 %   address output
982 %   ":" output
983 %   organization output
984   new.block
985   note output
986   fin.entry
987 }

```

6.2.9 The mastersthesis type

mastersthesis

```
988 FUNCTION {mastersthesis}
989 { output.bibitem
990   format.authors "author" output.check
991   author format.key output          % added
992   output.year.check                 % added
993   new.block
994   format.title "title" output.check
995   new.block
996   "Master's thesis" format.thesis.type output.nonnull
997   school "school" output.check
998   address output
999   new.block
1000  note output
1001  fin.entry
1002 }
```

6.2.10 The misc type

misc

```
1003 FUNCTION {misc}
1004 { output.bibitem
1005   format.authors output
1006   author format.key output          % added
1007   output.year.check                 % added
1008   title howpublished new.block.checkb
1009   format.title output
1010   new.block
1011   howpublished output
1012   new.block
1013   note output
1014   fin.entry
1015 }
```

6.2.11 The phdthesis type

phdthesis

```
1016 FUNCTION {phdthesis}
1017 { output.bibitem
1018   format.authors "author" output.check
1019   author format.key output          % added
1020   output.year.check                 % added
1021   new.block
1022   format.btitle % "title" output.check
1023   new.block
1024   "Ph.\ D. thesis" format.thesis.type output.nonnull
1025   school "school" output.check
1026   address output
1027   new.block
1028   note output
1029   fin.entry
1030 }
```

6.2.12 The proceedings type

proceedings

```
1031 FUNCTION {proceedings}
1032 { output.bibitem
1033   editor empty$
1034   { organization output
1035     organization format.key output } % gnp - changed from
1036   { format.editors output.nonnull } % author format.key
1037   if$
1038 % author format.key output          % gnp - removed (should be either
1039 %                                     editor or organization
1040   output.year.check                 % added (newapa)
1041   new.block
1042   format.btitle % "title" output.check
1043   format.bvolume output
1044   format.number.series output
1045 % address output
1046   format.proc.loc.date % output
1047   new.sentence
1048   organization output
1049   publisher output.colon
1050   new.block
1051   note output
1052   fin.entry
1053 }
```

6.2.13 The techreport type

techreport

```
1054 FUNCTION {techreport}
1055 { output.bibitem
1056   format.authors "author" output.check
1057   author format.key output          % added
1058   output.year.check                 % added
1059   new.block
1060   format.title "title" output.check
1061   new.block
1062   format.tr.number
1063     month empty$
1064     'skip$
1065     { month parenthesize * }
1066     if$
1067   output.nonnull
1068   institution "institution" output.check
1069   address output
1070   new.block
1071   note output
1072   fin.entry
1073 }
```

6.2.14 The unpublished type

unpublished

```
1074 FUNCTION {unpublished}
1075 { output.bibitem
1076   format.authors "author" output.check
1077   author format.key output           % added
1078   output.year.check                 % added
1079   new.block
1080   format.title "title" output.check
1081   new.block
1082   note "note" output.check
1083   fin.entry
1084 }
```

6.2.15 The default.type type

default.type

```
1085 FUNCTION {default.type} { misc }
```

6.3 Standard abbreviations

The following abbreviations should all be checked to make sure they correspond to what the ACM wants.

```
1086 MACRO {jan} {"Jan."}
1087
1088 MACRO {feb} {"Feb."}
1089
1090 MACRO {mar} {"Mar."}
1091
1092 MACRO {apr} {"Apr."}
1093
1094 MACRO {may} {"May"}
1095
1096 MACRO {jun} {"June"}
1097
1098 MACRO {jul} {"July"}
1099
1100 MACRO {aug} {"Aug."}
1101
1102 MACRO {sep} {"Sept."}
1103
1104 MACRO {oct} {"Oct."}
1105
1106 MACRO {nov} {"Nov."}
1107
1108 MACRO {dec} {"Dec."}
1109
1110 MACRO {acmcs} {"ACM Comput. Surv."}
1111
1112 MACRO {acta} {"Acta Inf."}
1113
1114 MACRO {ai} {"Artif. Intell."}
```

```

1115
1116 MACRO {cacm}      {"Commun. ACM"}
1117
1118 MACRO {ibmjrd}    {"IBM J. Res. Devel."}
1119
1120 MACRO {ibmsj}    {"IBM Syst. J."}
1121
1122 MACRO {ieeese}    {"IEEE Trans. Softw. Eng."}
1123
1124 MACRO {ieeetc}    {"IEEE Trans. Comput."}
1125
1126 MACRO {ieeetcad} {"IEEE Trans. Comput. Aided Des. Integr. Circuits"}
1127
1128 MACRO {ipl}       {"Inf. Process. Lett."}
1129
1130 MACRO {ic}        {"Inf. Comput."}
1131
1132 MACRO {jacm}      {"J. ACM"}
1133
1134 MACRO {jcass}     {"J. Comput. Syst. Sci."}
1135
1136 MACRO {scp}       {"Sci. Comput. Program."}
1137
1138 MACRO {sicomp}    {"SIAM J. Comput."}
1139
1140 MACRO {tocs}      {"ACM Trans. Comput. Syst."}
1141
1142 MACRO {tods}      {"ACM Trans. Database Syst."}
1143
1144 MACRO {tog}       {"ACM Trans. Graph."}
1145
1146 MACRO {toms}      {"ACM Trans. Math. Softw."}
1147
1148 MACRO {toois}     {"ACM Trans. Off. Inf. Syst"}
1149
1150 MACRO {tois}      {"ACM Trans. Inf. Syst."}
1151
1152 MACRO {toplas}    {"ACM Trans. Program. Lang. Syst."}
1153
1154 MACRO {tcs}       {"Theor. Comput. Sci."}
1155
1156 READ

```

6.4 Sorting the entries

sortify

```

1157 FUNCTION {sortify}
1158 {
1159     purify$
1160     "1" change.case$
1161 }

```

chop.word

```

1162 INTEGERS { len }
1163
1164 FUNCTION {chop.word}
1165 { 's :=
1166   'len :=
1167   s #1 len substring$ =
1168   { s len #1 + global.max$ substring$ }
1169   's
1170   if$
1171 }

```

sort.format.names

```

1172 FUNCTION {sort.format.names}
1173 { 's :=
1174   #1 'nameptr :=
1175   ""
1176   s num.names$ 'numnames :=
1177   numnames 'namesleft :=
1178   { namesleft #0 > }
1179   { nameptr #1 >
1180     { " " * }
1181     'skip$
1182     if$
1183     s nameptr "{vv{ }}{ll{ }}{ f{ }}{ jj{ }}" format.name$ 't :=
1184     nameptr numnames = t "others" = and
1185     { " et~al" * }
1186     { t sortify * }
1187     if$
1188     nameptr #1 + 'nameptr :=
1189     namesleft #1 - 'namesleft :=
1190   }
1191   while$
1192 }

```

sort.format.title

```

1193 FUNCTION {sort.format.title}
1194 { 't :=
1195   "A " #2
1196   "An " #3
1197   "The " #4 t chop.word
1198   chop.word
1199   chop.word
1200   sortify
1201   #1 global.max$ substring$
1202 }

```

author.sort

```

1203 FUNCTION {author.sort}
1204 { author empty$
1205   { key empty$
1206     { "to sort, need author or key in " cite$ * warning$
1207     " " }
1208     { key sortify }
1209   if$

```



```

1210     }
1211     { author sort.format.names }
1212   if$
1213 }

```

editor.sort

```

1214 FUNCTION {editor.sort}
1215 { editor empty$
1216   { key empty$
1217     { "to sort, need editor or key in " cite$ * warning$
1218       ""
1219     }
1220     { key sortify }
1221   if$
1222 }
1223 { editor sort.format.names }
1224 if$
1225 }

```

author.editor.sort

```

1226 FUNCTION {author.editor.sort}
1227 {
1228   author empty$

```

Removed warning generated by lack of author since there's nothing wrong with having an editor and no author. – dmj, 4/26/96

```

1229   { % "missing author in " cite$ * warning$
1230     editor empty$
1231     { key empty$
1232       { "to sort, need author, editor, or key in "
1233         cite$ * warning$
1234         ""
1235       }
1236       {
1237         key sortify
1238       }
1239     if$
1240   }
1241   {
1242     editor sort.format.names
1243   }
1244   if$
1245 }
1246 {
1247   author sort.format.names
1248 }
1249 if$
1250 }

```

author.organization.sort added - GNP. Stack author or organization for sorting (from alpha.bst). Unlike alpha.bst, we need entire names, not abbreviations

```

1251 FUNCTION {author.organization.sort}
1252 { author empty$
1253   { organization empty$

```

```

1254     { key empty$
1255         { "to sort, need author, organization, or key in "
1256             cite$ * warning$
1257             ""
1258         }
1259         { key sortify }
1260     if$
1261 }
1262 { organization sortify }
1263 if$
1264 }
1265 { author sort.format.names }
1266 if$
1267 }

```

`editor.organization.sort` added - GNP. Stack editor or organization for sorting (from `alpha.bst`). Unlike `alpha.bst`, we need entire names, not abbreviations

```

1268 FUNCTION {editor.organization.sort}
1269 { editor empty$
1270     { organization empty$
1271         { key empty$
1272             { "to sort, need editor, organization, or key in "
1273                 cite$ * warning$
1274                 ""
1275             }
1276             { key sortify }
1277         if$
1278         }
1279         { organization sortify }
1280     if$
1281 }
1282 { editor sort.format.names }
1283 if$
1284 }

```

`presort` Presort creates the bibentry's label via a call to `calc.label`, and then sorts the entries based on entry type. `Chicago.bst` adds support for including organizations as the sort key; the following is stolen from `alpha.bst`.

```

1285 FUNCTION {presort}
1286 { calc.label sortify % recalculate bibitem label
1287   year field.or.null purify$ #-1 #4 substring$ * % add year
1288   " "
1289   *
1290   type$ "book" =
1291   type$ "inbook" =
1292   or
1293   'author.editor.sort
1294   { type$ "proceedings" =
1295     'editor.organization.sort
1296     { type$ "manual" =
1297       'author.organization.sort
1298       'author.sort
1299     if$

```

```

1300     }
1301     if$
1302   }
1303   if$
1304   #1 entry.max$ substring$           % added for newapa
1305   'sort.label :=                     % added for newapa
1306   sort.label                         % added for newapa
1307   *
1308   "  "
1309   *
1310   title field.or.null
1311   sort.format.title
1312   *
1313   #1 entry.max$ substring$
1314   'sort.key$ :=
1315 }

1316 ITERATE {presort}
1317
1318 SORT    % by label, year, author/editor, title
1319

```

init.extra.label.stuff

```

1320 STRINGS { last.label next.extra }
1321
1322 INTEGERS { last.extra.num }
1323
1324 FUNCTION {init.extra.label.stuff}
1325 { #0 int.to.chr$ 'last.label :=
1326   "" 'next.extra :=
1327   #0 'last.extra.num :=
1328 }

```

forward.pass Pass through all entries, comparing current entry to last one. Need to concatenate year to the stack (done by calc.label) to determine if two entries are the same (see presort)

```

1329 FUNCTION {forward.pass}
1330 { last.label
1331   calc.label year field.or.null purify$ #-1 #4 substring$ * % add year
1332   #1 entry.max$ substring$ = % are they equal?
1333   { last.extra.num #1 + 'last.extra.num :=
1334     last.extra.num int.to.chr$ 'extra.label :=
1335   }
1336   { "a" chr.to.int$ 'last.extra.num :=
1337     "" 'extra.label :=
1338     calc.label
1339     year field.or.null purify$ #-1 #4 substring$ * % add year
1340     #1 entry.max$ substring$ 'last.label := % assign to last.label
1341   }
1342   if$
1343 }

```

reverse.pass

```

1344 FUNCTION {reverse.pass}

```

```

1345 { next.extra "b" =
1346     { "a" 'extra.label := }
1347     'skip$
1348 if$
1349 label.year extra.label * 'sort.year :=
1350 extra.label 'next.extra :=
1351 }

1352 EXECUTE {init.extra.label.stuff}
1353
1354 ITERATE {forward.pass}
1355
1356 REVERSE {reverse.pass}

bib.sort.order

1357 FUNCTION {bib.sort.order}
1358 { sort.label
1359     " "
1360     *
1361     year field.or.null sortify
1362     *
1363     " "
1364     *
1365     title field.or.null
1366     sort.format.title
1367     *
1368     #1 entry.max$ substring$
1369     'sort.key$ :=
1370 }

1371 ITERATE {bib.sort.order}
1372
1373 SORT % by sort.label, year, title --- giving final bib. order.

begin.bib

1374 FUNCTION {begin.bib}
1375 { preamble$ empty$
1376     'skip$
1377     { preamble$ write$ newline$ }
1378 if$
1379 "\begin{thebibliography}{" write$ newline$
1380 }

1381 EXECUTE {begin.bib}
1382
1383 EXECUTE {init.state.consts}
1384
1385 ITERATE {call.type$}

end.bib

1386 FUNCTION {end.bib}
1387 { newline$
1388     "\end{thebibliography}" write$ newline$
1389 }

1390 EXECUTE {end.bib}

```