

```

ENTRY  address author booktitle chapter edition editor howpublished institution journal key note
number organization pages publisher school series title type volume year  label extra.label sort.label
INTEGERS  output.state before.all mid.sentence after.sentence after.block docomma givetitles
FUNCTION init.state.consts 0 'before.all := 1 'mid.sentence := 2 'after.sentence := 3 'after.block :=
0 'docomma :=
FUNCTION init.toggle.switches 0 'givetitles :=
STRINGS  s t
FUNCTION output.nonnull 's := output.state mid.sentence =
docomma 0 = " " * write", " * write ifoutput.stateafter.block = add.period writenewline "" writeoutput.statebefore
add.period"" * write ifif mid.sentence 'output.state := ifs
FUNCTION output duplicateempty 'pop'output.nonnullif
FUNCTION output.check 't := duplicateempty t "title" = poppop "empty" t * " in " * cite*warning
if'output.nonnullif
FUNCTION output.year.check year empty"emptyyearin"cite ", using label: " extra.label * * * warningwrite
" (" extra.label * ") " *
label "" extra.label "" * * * 'label :=
mid.sentence 'output.state := write
" (" year * "" extra.label "" * * * ") " *
mid.sentence 'output.state := if
FUNCTION output.bibitem newline["write label write"]"write citewrite "" writenewline "" before.all
'output.state :=
FUNCTION fin.entry add.periodwrite newline
FUNCTION new.block output.state before.all = 'skipafter.block'output.state :=if
FUNCTION new.sentence output.state after.block = 'skipoutput.statebefore.all =' skip after.sentence
'output.state := ifif
FUNCTION not 0 1 if
FUNCTION and 'skippop 0 if
FUNCTION or pop1'skip if
FUNCTION new.block.checkb emptyswap emptyand'skip 'new.block if
FUNCTION field.or.null duplicateempty pop""'skip if
FUNCTION emphasize duplicateempty pop""""swap * "" * if
INTEGERS nameptr namesleft numnames
FUNCTION format.names 's := 1 'nameptr := s num.names'numnames := numnames'namesleft :=
namesleft0 > snameptr"vv ll, jj, f."format.name 't := nameptr 1 ; namesleft 1 ; ", " * t * numnames
2 ; " " * 'skipif t "others" = " et al." * " and " * t * ifif 't ifnameptr1 + 'nameptr := namesleft1 -'
namesleft := while
FUNCTION format.authors author empty""authorformat.namesif
FUNCTION format.key emptykeyfield.or.null""if
FUNCTION format.editors editor empty""editorformat.nameseditornum.names 1 ; ", eds" * ",
ed." * ifif
FUNCTION special.change.case
title 1 1 substring
title "l" change.case
2 global.maxsubstring
*
FUNCTION format.title title empty""titleif
FUNCTION n.dashify 't := "" t emptynott1substring "-" = t 1 2 substring"--" = not" --" * t2global.max
substring't := t1substring "-" = "-" * t 2 global.maxsubstring 't := whileif t 1 1 substring*t2global.max
substring't := if while
FUNCTION format.btitle title emphasize
FUNCTION tie.or.space.connect duplicatetext.length 3 ; " " " " ifswap * *
FUNCTION either.or.check empty'pop "can't use both" swap*"fieldsin" * cite * warningif

```

FUNCTION format.bvolume volume empty"" *vol." volume", " * tie.or.space.connectseriesempty 'skip" of" * series*
"vol. and num." number either.or.check if
 FUNCTION format.number.series volume empty *numberempty* series field.or.null output.state mid.sentence
 = "number" "Number" if *numbertie.or.space.connectseriesempty* "there's a number but no series in "
*cite*warning* " in " * series * if *if* " if
 FUNCTION format.edition edition empty"" *output.statemid.sentence = edition"l" change.case* " edi-
 tion" * edition "t" change.case *edition" * if* if
 INTEGERS multiresult
 FUNCTION multi.page.check 't := 0 'multiresult := multiresult not t empty *notandt11substring*
 duplicate" - " = *swap* duplicate", " = *swap* "+" = or or 1 'multiresult := t 2 global.maxsubstring 't :=
ifwhile multiresult
 FUNCTION format.pages pages empty"" *pagesmulti.page.check"pp." pagesn.dashifytie.or.space.connect"p." pages*
 if
 FUNCTION format.vol.num.pages "" volume "" * * field.or.null
 pages empty' *skip* duplicate *empty* pop *format.pages", " * pagesn.dashify*if* if
 FUNCTION format.chapter.pages chapter empty' *format.pagetypeempty* "chapter" type "l" change.case if
 chapter tie.or.space.connect pages empty' *skip* ", " * format.pages * if *if*
 FUNCTION format.in.ed.booktitle booktitle empty"" *editoreempty* "In " booktitle emphasize * "In
 " booktitle emphasize * ", " * "(" format.editors ")," * * * if *if*
 FUNCTION format.thesis.type type empty' *skip* po *type"t" change.case* if
 FUNCTION format.tr.number type empty' *TechnicalReport" 'typeif* number empty" *t" change.case*
 number tie.or.space.connect if
 FUNCTION format.article.crossref "In" " " * crossref * "" *
 FUNCTION format.book.crossref volume empty" *emptyvolumein"cite* * "'s crossref of " * crossref *
 warning" *In" " Volume"volumetie.or.space.connect" of" *if* "" * crossref * "" *
 FUNCTION format.incoll.inproc.crossref "In" " " * crossref * "" *
 FUNCTION article cite" *TitlesOn" = ' skip* output.bibitem format.authors "author" output.check
 author format.key output output.year.check new.block format.title "title" output.check new.block crossref
 missing
 journal -1 1 substring"." = *journaljournal"" *if* emphasize "journal" output.check
 format.vol.num.pages output format.article.crossref output.nonnull format.pages output if *fin.entryif*
 FUNCTION book output.bibitem author empty *format.editors"authorandeditor" output.checkeditor* format.key
 "author and editor" editor either.or.check 'skip *if* if *output.year.checknew.blockformat.btitle" title" output.check1' docomma*
crossrefmissing format.bvolume output new.block format.number.series output format.edition output pub-
 lisher "publisher" output.check address output new.block format.book.crossref output.nonnull if *new.blocknoteoutputfin.*
 FUNCTION booklet output.bibitem format.authors output author format.key output output.year.check
 new.block format.title "title" output.check new.block howpublished output address output new.block note
 output fin.entry
 FUNCTION inbook output.bibitem author empty *format.editors"authorandeditor" output.checkeditor* format.key
 "author and editor" editor either.or.check 'skip *if* if *output.year.checknew.blockformat.btitle" title" output.checkcrossref*
 format.bvolume output format.chapter.pages "chapter and pages" output.check new.block format.number.series
 output new.sentence 0 'docomma := address ":" * output publisher "publisher" output.check 1 'docomma
 := format.chapter.pages "chapter and pages" output.check new.block format.book.crossref output.nonnull
 if *format.editionoutputnew.blocknoteoutputfin.entry*
 FUNCTION incollection output.bibitem format.authors "author" output.check author format.key out-
 put output.year.check new.block format.title "title" output.check new.block crossref missing *format.in.ed.booktitle" bookti*
 new.block note output fin.entry
 FUNCTION inproceedings output.bibitem format.authors "author" output.check author format.key
 output output.year.check new.block format.title "title" output.check new.block crossref missing *format.in.ed.booktitle" bo*
 output " , " * output if
 organization output 0 'docomma := publisher "publisher" output.check 1 'docomma :=
 address output 0 'docomma := format.incoll.inproc.crossref output.nonnull format.pages output if *new.blocknoteoutp*
 FUNCTION conference inproceedings

FUNCTION manual output.bibitem format.authors output author format.key output output.year.check
new.block format.btitle "title" output.check organization address new.block.checkb organization output ad-
dress output format.edition output new.block note output fin.entry

FUNCTION mastersthesis output.bibitem format.authors "author" output.check author format.key
output output.year.check new.block format.title "title" output.check new.block "Master's thesis" format.thesis.type
output.nonnull school "school" output.check address output new.block note output fin.entry

FUNCTION misc output.bibitem format.authors output author format.key output output.year.check
new.block format.title output new.block howpublished output new.block note output fin.entry

FUNCTION phdthesis output.bibitem format.authors "author" output.check author format.key output
output.year.check new.block format.btitle "title" output.check new.block "PhD thesis," format.thesis.type
output.nonnull school "school" output.check address output new.block note output fin.entry

FUNCTION proceedings output.bibitem format.editors output editor format.key output output.year.check
new.block format.btitle "title" output.check format.bvolume output format.number.series output address
output new.sentence organization output publisher output new.block note output fin.entry

FUNCTION techreport output.bibitem format.authors "author" output.check author format.key out-
put output.year.check new.block format.title "title" output.check new.block format.tr.number output.nonnull
institution "institution" output.check address output new.block note output fin.entry

FUNCTION unpublished output.bibitem format.authors "author" output.check author format.key out-
put output.year.check new.block

title empty""*title"t" change.case if" title" output.check*

new.block note "note" output.check fin.entry

FUNCTION default.type misc

MACRO jan "January"

MACRO feb "February"

MACRO mar "March"

MACRO apr "April"

MACRO may "May"

MACRO jun "June"

MACRO jul "July"

MACRO aug "August"

MACRO sep "September"

MACRO oct "October"

MACRO nov "November"

MACRO dec "December"

MACRO acmcs "ACM Computing Surveys"

MACRO acta "Acta Informatica"

MACRO cacm "Communications of the ACM"

MACRO ibmjrd "IBM Journal of Research and Development"

MACRO ibmsj "IBM Systems Journal"

MACRO ieese "IEEE Transactions on Software Engineering"

MACRO ieetc "IEEE Transactions on Computers"

MACRO ieetcad "IEEE Transactions on Computer-Aided Design of Integrated Circuits"

MACRO ipl "Information Processing Letters"

MACRO jacm "Journal of the ACM"

MACRO jcsc "Journal of Computer and System Sciences"

MACRO scp "Science of Computer Programming"

MACRO sicomp "SIAM Journal on Computing"

MACRO tocs "ACM Transactions on Computer Systems"

MACRO tods "ACM Transactions on Database Systems"

MACRO tog "ACM Transactions on Graphics"

MACRO toms "ACM Transactions on Mathematical Software"

MACRO toois "ACM Transactions on Office Information Systems"

MACRO toplas "ACM Transactions on Programming Languages and Systems"

```

MACRO tcs "Theoretical Computer Science"
READ
FUNCTION check.for.titlecommand
cite"TitlesOn" = "TitlesWillBePrinted"warning 1 'givetitles := 'skipif EXECUTE init.toggle.switches
ITERATE check.for.titlecommand
FUNCTION sortify purify"l"change.case
INTEGERS len
FUNCTION chop.word 's := 'len := s 1 len substring= slen1 + global.max substring'sif
FUNCTION format.lab.names 's := s 1 "vv ll" format.namesnum.names duplicate2 > pop " et al." *
2 | 'skips2"ffvlljj"format.name "others" = " et al." * " and " * s 2 "vv ll" format.name*if ifif
FUNCTION author.key.label author emptykeyempty cite13substring 'key ifauthorformat.lab.namesif

FUNCTION author.editor.key.label author emptyeditoreempty key emptycite 1 3 substring'keyif ed-
itor format.lab.names ifauthorformat.lab.namesif
FUNCTION editor.key.label editor emptykeyempty cite13substring 'key ifeditorformat.lab.namesif
FUNCTION calc.label type"book" = type "inbook" = or 'author.editor.key.label type"proceedings" =
editor.key.label'author.key.labelif if'," * yearfield.or.nullpurify -1 4 substring*'label :=
FUNCTION sort.format.names 's := 1 'nameptr := "" s num.names'numnames := numnames'namesleft :=
namesleft0 > nameptr1 > ""*'skip ifnameptr"vllfjj"format.name 't := nameptr numnames = t "oth-
ers" = and "et al" * t sortify * ifnameptr1 +' nameptr := namesleft1 -' namesleft := while
FUNCTION sort.format.title 't := "A " 2 "An " 3 "The " 4 t chop.word chop.word chop.word sortify
1 global.maxsubstring
FUNCTION author.sort author emptykeyempty "to sort, need author or key in " cite*warning ""
key sortify ifauthorsort.format.namesif
FUNCTION author.editor.sort author emptyeditoreempty key empty"tosort, needauthor, editor, orkeyin" cite
* warning""keysortifyif editor sort.format.names ifauthorsort.format.namesif
FUNCTION editor.sort editor emptykeyempty "to sort, need editor or key in " cite*warning "" key
sortify ifeditorsort.format.namesif
FUNCTION presort calc.label label sortify " " * type"book" = type "inbook" = or 'author.editor.sort
type"proceedings" = ' editor.sort'author.sortif iflentry.max substring'sort.label := sort.label*" "*titlefield.or.nullsort
lentry.max substring'sort.key :=
ITERATE presort
SORT
STRINGS last.label next.extra
INTEGERS last.extra.num
FUNCTION initialize.extra.label.stuff 0 int.to.chr'last.label := ""'next.extra := 0'last.extra.num :=
FUNCTION forward.pass
last.label label = last.extra.num 1 + 'last.extra.num := last.extra.num int.to.chr'extra.label := "a"chr.to.int
'last.extra.num :=
year empty"a"'extra.label :=""'extra.label :=if
label 'last.label := if
FUNCTION reverse.pass next.extra "b" = "a" 'extra.label := 'skipif
extra.label "" = 'skiplabel""extra.label"" * * 'label :=if
extra.label 'next.extra :=
FUNCTION bib.sort.order sort.label " " * year duplicateempty popcite 'skipif
* " " *
cite*
1 entry.maxsubstring 'sort.key:=
EXECUTE initialize.extra.label.stuff ITERATE bib.sort.order SORT ITERATE forward.pass REVERSE
reverse.pass
FUNCTION begin.bib preambleempty 'skippreamble writenewline if'thebibliography"write newline
EXECUTE begin.bib
EXECUTE init.state.consts

```

```
ITERATE call.type  
FUNCTION end.bib newline"
```