

%

*%A printer friendly PDF version of this page is available [abbrv.pdf \(36Kb\)](#)*

## **BibTeX standard bibliography style ‘abbrv’**

% version 0.99a for BibTeX versions 0.99a or later, LaTeX version 2.09.

% Copyright (C) 1985, 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

% btxbst.doc, plain.bst, unsrt.bst, alpha.bst, and abbrv.bst.

% This restriction helps ensure that all standard styles are identical.

% The file btxbst.doc has the documentation for this style.

ENTRY

{ address

author

booktitle

chapter

edition

editor

howpublished

institution

journal

key

month

note

number

organization

pages

publisher

school

series

title

type

volume

year

}

{ }

{ label }

INTEGERS { output.state before.all mid.sentence after.sentence after.block }

FUNCTION { init.state.consts }

{ #0 'before.all :=

#1 'mid.sentence :=

#2 'after.sentence :=

#3 'after.block :=

}

STRINGS { s t }

```

FUNCTION {output.nonnull}
{ 's :=
output.state mid.sentence =
{ ", " * write$ }
{ output.state after.block =
{ add.period$ write$
newline$
"\newblock " write$
}
{ output.state before.all =
'write$
{ add.period$ " " * write$ }
if$
}
if$
mid.sentence 'output.state :=
}
if$
s
}
FUNCTION {output}
{ duplicate$ empty$
'pop$
'output.nonnull
if$
}
FUNCTION {output.check}
{ 't :=
duplicate$ empty$
{ pop$ "empty " t * " in " * cite$ * warning$ }
'output.nonnull
if$
}
FUNCTION {output.bibitem}
{ newline$
"\bibitem{" write$
cite$ write$
"}" write$
newline$
""
before.all 'output.state :=
}
FUNCTION {fin.entry}
{ add.period$
write$
newline$
}
FUNCTION {new.block}

```

```

{ output.state before.all =
'skip$
{ after.block 'output.state := }
if$
}
FUNCTION {new.sentence}
{ output.state after.block =
'skip$
{ output.state before.all =
'skip$
{ after.sentence 'output.state := }
if$
}
if$
}
FUNCTION {not}
{ { #0 }
{ #1 }
if$
}
FUNCTION {and}
{ 'skip$
{ pop$ #0 }
if$
}
FUNCTION {or}
{ { pop$ #1 }
'skip$
if$
}
FUNCTION {new.block.checka}
{ empty$
'skip$
'new.block
if$
}
FUNCTION {new.block.checkb}
{ empty$
swap$ empty$
and
'skip$
'new.block
if$
}
FUNCTION {new.sentence.checka}
{ empty$
'skip$
'new.sentence

```

```

if$
}
FUNCTION {new.sentence.checkb}
{ empty$
swap$ empty$
and
'skip$
'new.sentence
if$
}
FUNCTION {field.or.null}
{ duplicate$ empty$
{ pop$ "" }
'skip$
if$
}
FUNCTION {emphasize}
{ duplicate$ empty$
{ pop$ "" }
{"{\em " swap$ * "}" * }
if$
}
INTEGERS { nameptr namesleft numnames }
FUNCTION {format.names}
{ 's :=
#1 'nameptr :=
s num.names$ 'numnames :=
numnames 'namesleft :=
{ namesleft #0 > }
{ s nameptr "{f.~}{vv~}{ll}{, jj}" format.name$ 't :=
nameptr #1 >
{ namesleft #1 >
{ ", " * t * }
{ numnames #2 >
{ ", " * }
'skip$
if$
t "others" =
{ " et~al." * }
{ " and " * t * }
if$
}
}
if$
}
't
if$
nameptr #1 + 'nameptr :=
namesleft #1 - 'namesleft :=

```

```

}
while$
}
FUNCTION {format.authors}
{ author empty$
{ "" }
{ author format.names }
if$
}
FUNCTION {format.editors}
{ editor empty$
{ "" }
{ editor format.names
editor num.names$ #1 >
{ ", editors" * }
{ ", editor" * }
if$
}
if$
}
FUNCTION {format.title}
{ title empty$
{ "" }
{ title "t" change.case$ }
if$
}
FUNCTION {n.dashify}
{ 't :=
""

{ t empty$ not }
{ t #1 #1 substring$ "-" =
{ t #1 #2 substring$ "--" = not
{ "--" *
t #2 global.max$ substring$ 't :=
}
{ { t #1 #1 substring$ "-" = }
{ "-" *
t #2 global.max$ substring$ 't :=
}
while$
}
if$
}
{ t #1 #1 substring$ *
t #2 global.max$ substring$ 't :=
}
if$
}

```

```

while$
}
FUNCTION {format.date}
{ year empty$
{ month empty$
{ "" }
{ "there's a month but no year in " cite$ * warning$
month
}
}
if$
}
{ month empty$
`year
{ month " " * year * }
if$
}
if$
}
FUNCTION {format.btitle}
{ title emphasize
}
FUNCTION {tie.or.space.connect}
{ duplicate$ text.length$ #3 <
{ "~" }
{ " " }
if$
swap$ * *
}
FUNCTION {either.or.check}
{ empty$
`pop$
{ "can't use both " swap$ * " fields in " * cite$ * warning$ }
if$
}
FUNCTION {format.bvolume}
{ volume empty$
{ "" }
{ "volume" volume tie.or.space.connect
series empty$
`skip$
{ " of " * series emphasize * }
if$
"volume and number" number either.or.check
}
if$
}
FUNCTION {format.number.series}
{ volume empty$

```

```

{ number empty$
{ series field.or.null }
{ output.state mid.sentence =
{ "number" }
{ "Number" }
if$
number tie.or.space.connect
series empty$
{ "there's a number but no series in " cite$ * warning$ }
{ " in " * series * }
if$
}
if$
}
{ "" }
if$
}
FUNCTION {format.edition}
{ edition empty$
{ "" }
{ output.state mid.sentence =
{ edition "l" change.case$ " edition" * }
{ edition "t" change.case$ " edition" * }
if$
}
if$
}
INTEGERS { multiresult }
FUNCTION {multi.page.check}
{ 't :=
#0 'multiresult :=
{ multiresult not
t empty$ not
and
}
{ t #1 #1 substring$
duplicate$ "-" =
swap$ duplicate$ "," =
swap$ "+" =
or or
{ #1 'multiresult := }
{ t #2 global.max$ substring$ 't := }
if$
}
while$
multiresult
}
FUNCTION {format.pages}

```

```

{ pages empty$
{ "" }
{ pages multi.page.check
{ "pages" pages n.dashify tie.or.space.connect }
{ "page" pages tie.or.space.connect }
if$
}
if$
}
FUNCTION {format.vol.num.pages}
{ volume field.or.null
number empty$
'skip$
{ "(" number * ")" * *
volume empty$
{ "there's a number but no volume in " cite$ * warning$ }
'skip$
if$
}
if$
pages empty$
'skip$
{ duplicate$ empty$
{ pop$ format.pages }
{ ":" * pages n.dashify * }
if$
}
if$
}
FUNCTION {format.chapter.pages}
{ chapter empty$
'format.pages
{ type empty$
{ "chapter" }
{ type "I" change.case$ }
if$
chapter tie.or.space.connect
pages empty$
'skip$
{ ", " * format.pages * }
if$
}
if$
}
FUNCTION {format.in.ed.booktitle}
{ booktitle empty$
{ "" }
{ editor empty$

```



```

{ "In " booktitle emphasize * }
{ "In " format.editors * ", " * booktitle emphasize * }
if$
}
if$
}
FUNCTION {empty.misc.check}
{ author empty$ title empty$ howpublished empty$
month empty$ year empty$ note empty$
and and and and and
key empty$ not and
{ "all relevant fields are empty in " cite$ * warning$ }
'skip$
if$
}
FUNCTION {format.thesis.type}
{ type empty$
'skip$
{ pop$
type "t" change.case$
}
if$
}
FUNCTION {format.tr.number}
{ type empty$
{ "Technical Report" }
'type
if$
number empty$
{ "t" change.case$ }
{ number tie.or.space.connect }
if$
}
FUNCTION {format.article.crossref}
{ key empty$
{ journal empty$
{ "need key or journal for " cite$ * " to crossref " * crossref *
warning$
""
}
{ "In {\em " journal * "V}" * }
if$
}
{ "In " key * }
if$
" \cite{" * crossref * "}" *
}
FUNCTION {format.crossref.editor}

```

```

{ editor #1 "{vv~}{ll}" format.name$
editor num.names$ duplicate$
#2 >
{ pop$ " et~al." * }
{ #2 <
'skip$
{ editor #2 "{ff }{vv }{ll}{jj}" format.name$ "others" =
{ " et~al." * }
{ " and " * editor #2 "{vv~}{ll}" format.name$ * }
if$
}
if$
}
if$
}
FUNCTION {format.book.crossref}
{ volume empty$
{ "empty volume in " cite$ * "'s crossref of " * crossref * warning$
"In "
}
{ "Volume" volume tie.or.space.connect
" of " *
}
if$
editor empty$
editor field.or.null author field.or.null =
or
{ key empty$
{ series empty$
{ "need editor, key, or series for " cite$ * " to crossref " *
crossref * warning$
"" *
}
{ "{\em " * series * "V}" * }
if$
}
{ key * }
if$
}
{ format.crossref.editor * }
if$
" \cite{" * crossref * }" *
}
FUNCTION {format.incoll.inproc.crossref}
{ editor empty$
editor field.or.null author field.or.null =
or
{ key empty$

```

```

{ booktitle empty$
{ "need editor, key, or booktitle for " cite$ * " to crossref " *
crossref * warning$
""
}
{ "In {\em " booktitle * "V}" * }
if$
}
{ "In " key * }
if$
}
{ "In " format.crossref.editor * }
if$
"\cite{" * crossref * "}" *
}
FUNCTION {article}
{ output.bibitem
format.authors "author" output.check
new.block
format.title "title" output.check
new.block
crossref missing$
{ journal emphasize "journal" output.check
format.vol.num.pages output
format.date "year" output.check
}
{ format.article.crossref output.nonnull
format.pages output
}
if$
new.block
note output
fin.entry
}
FUNCTION {book}
{ output.bibitem
author empty$
{ format.editors "author and editor" output.check }
{ format.authors output.nonnull
crossref missing$
{ "author and editor" editor either.or.check }
'skip$
if$
}
if$
new.block
format.btitle "title" output.check
crossref missing$

```

```

{ format.bvolume output
new.block
format.number.series output
new.sentence
publisher "publisher" output.check
address output
}
{ new.block
format.book.crossref output.nonnull
}
if$
format.edition output
format.date "year" output.check
new.block
note output
fin.entry
}
FUNCTION {booklet}
{ output.bibitem
format.authors output
new.block
format.title "title" output.check
howpublished address new.block.checkb
howpublished output
address output
format.date output
new.block
note output
fin.entry
}
FUNCTION {inbook}
{ output.bibitem
author empty$
{ format.editors "author and editor" output.check }
{ format.authors output.nonnull
crossref missing$
{ "author and editor" editor either.or.check }
'skip$
if$
}
if$
new.block
format.btitle "title" output.check
crossref missing$
{ format.bvolume output
format.chapter.pages "chapter and pages" output.check
new.block
format.number.series output

```

```

new.sentence
publisher "publisher" output.check
address output
}
{ format.chapter.pages "chapter and pages" output.check
new.block
format.book.crossref output.nonnull
}
if$
format.edition output
format.date "year" output.check
new.block
note output
fin.entry
}
FUNCTION {incollection}
{ output.bibitem
format.authors "author" output.check
new.block
format.title "title" output.check
new.block
crossref missing$
{ format.in.ed.booktitle "booktitle" output.check
format.bvolume output
format.number.series output
format.chapter.pages output
new.sentence
publisher "publisher" output.check
address output
format.edition output
format.date "year" output.check
}
{ format.incoll.inproc.crossref output.nonnull
format.chapter.pages output
}
}
if$
new.block
note output
fin.entry
}
FUNCTION {inproceedings}
{ output.bibitem
format.authors "author" output.check
new.block
format.title "title" output.check
new.block
crossref missing$
{ format.in.ed.booktitle "booktitle" output.check

```

```

format.bvolume output
format.number.series output
format.pages output
address empty$
{ organization publisher new.sentence.checkb
organization output
publisher output
format.date "year" output.check
}
{ address output.nonnull
format.date "year" output.check
new.sentence
organization output
publisher output
}
if$
}
{ format.incoll.inproc.crossref output.nonnull
format.pages output
}
if$
new.block
note output
fin.entry
}
FUNCTION {conference} { inproceedings }
FUNCTION {manual}
{ output.bibitem
author empty$
{ organization empty$
'skip$
{ organization output.nonnull
address output
}
if$
}
{ format.authors output.nonnull }
if$
new.block
format.btitle "title" output.check
author empty$
{ organization empty$
{ address new.block.checka
address output
}
'skip$
if$
}

```

```

{ organization address new.block.checkb
organization output
address output
}
if$
format.edition output
format.date output
new.block
note output
fin.entry
}
FUNCTION {mastersthesis}
{ output.bibitem
format.authors "author" output.check
new.block
format.title "title" output.check
new.block
"Master's thesis" format.thesis.type output.nonnull
school "school" output.check
address output
format.date "year" output.check
new.block
note output
fin.entry
}
FUNCTION {misc}
{ output.bibitem
format.authors output
title howpublished new.block.checkb
format.title output
howpublished new.block.checka
howpublished output
format.date output
new.block
note output
fin.entry
empty.misc.check
}
FUNCTION {phdthesis}
{ output.bibitem
format.authors "author" output.check
new.block
format.btitle "title" output.check
new.block
"PhD thesis" format.thesis.type output.nonnull
school "school" output.check
address output
format.date "year" output.check

```

```

new.block
note output
fin.entry
}
FUNCTION {proceedings}
{ output.bibitem
editor empty$
{ organization output }
{ format.editors output.nonnull }
if$
new.block
format.btitle "title" output.check
format.bvolume output
format.number.series output
address empty$
{ editor empty$
{ publisher new.sentence.checka }
{ organization publisher new.sentence.checkb }
organization output
}
if$
publisher output
format.date "year" output.check
}
{ address output.nonnull
format.date "year" output.check
new.sentence
editor empty$
'skip$
{ organization output }
if$
publisher output
}
if$
new.block
note output
fin.entry
}
FUNCTION {techreport}
{ output.bibitem
format.authors "author" output.check
new.block
format.title "title" output.check
new.block
format.tr.number output.nonnull
institution "institution" output.check
address output
format.date "year" output.check

```



```

new.block
note output
fin.entry
}
FUNCTION {unpublished}
{ output.bibitem
format.authors "author" output.check
new.block
format.title "title" output.check
new.block
note "note" output.check
format.date output
fin.entry
}
FUNCTION {default.type} { misc }
MACRO {jan} {"Jan."}
MACRO {feb} {"Feb."}
MACRO {mar} {"Mar."}
MACRO {apr} {"Apr."}
MACRO {may} {"May"}
MACRO {jun} {"June"}
MACRO {jul} {"July"}
MACRO {aug} {"Aug."}
MACRO {sep} {"Sept."}
MACRO {oct} {"Oct."}
MACRO {nov} {"Nov."}
MACRO {dec} {"Dec."}
MACRO {acmcs} {"ACM Comput. Surv."}
MACRO {acta} {"Acta Inf."}
MACRO {cacm} {"Commun. ACM"}
MACRO {ibmjrd} {"IBM J. Res. Dev."}
MACRO {ibmsj} {"IBM Syst.~J."}
MACRO {ieeese} {"IEEE Trans. Softw. Eng."}
MACRO {ieeetc} {"IEEE Trans. Comput."}
MACRO {ieeetcad}
{"IEEE Trans. Comput.-Aided Design Integrated Circuits"}
MACRO {ipl} {"Inf. Process. Lett."}
MACRO {jacm} {"J.~ACM"}
MACRO {jcss} {"J.~Comput. Syst. Sci."}
MACRO {scp} {"Sci. Comput. Programming"}
MACRO {sicomp} {"SIAM J. Comput."}
MACRO {tocs} {"ACM Trans. Comput. Syst."}
MACRO {tods} {"ACM Trans. Database Syst."}
MACRO {tog} {"ACM Trans. Gr."}
MACRO {toms} {"ACM Trans. Math. Softw."}
MACRO {toois} {"ACM Trans. Office Inf. Syst."}
MACRO {toplas} {"ACM Trans. Prog. Lang. Syst."}
MACRO {tcs} {"Theoretical Comput. Sci."}

```

```

READ
FUNCTION {sortify}
{ purify$
"l" change.case$
}
INTEGERS { len }
FUNCTION {chop.word}
{ 's :=
'len :=
s #1 len substring$ =
{ s len #1 + global.max$ substring$ }
's
if$
}
FUNCTION {sort.format.names}
{ 's :=
#1 'nameptr :=
""

s num.names$ 'numnames :=
numnames 'namesleft :=
{ namesleft #0 > }
{ nameptr #1 >
{ " " * }
'skip$
if$
s nameptr "{vv{ }}{ll{ }}{ f{ }}{ jj{ }}" format.name$ 't :=
nameptr numnames = t "others" = and
{ "et al" * }
{ t sortify * }
if$
nameptr #1 + 'nameptr :=
namesleft #1 - 'namesleft :=
}
while$
}
FUNCTION {sort.format.title}
{ 't :=
"A " #2
"An " #3
"The " #4 t chop.word
chop.word
chop.word
sortify
#1 global.max$ substring$
}
FUNCTION {author.sort}
{ author empty$
{ key empty$

```

```

{ "to sort, need author or key in " cite$ * warning$
""
}
{ key sortify }
if$
}
{ author sort.format.names }
if$
}
FUNCTION {author.editor.sort}
{ author empty$
{ editor empty$
{ key empty$
{ "to sort, need author, editor, or key in " cite$ * warning$
""
}
{ key sortify }
if$
}
{ editor sort.format.names }
if$
}
{ author sort.format.names }
if$
}
FUNCTION {author.organization.sort}
{ author empty$
{ organization empty$
{ key empty$
{ "to sort, need author, organization, or key in " cite$ * warning$
""
}
{ key sortify }
if$
}
{ "The " #4 organization chop.word sortify }
if$
}
{ author sort.format.names }
if$
}
FUNCTION {editor.organization.sort}
{ editor empty$
{ organization empty$
{ key empty$
{ "to sort, need editor, organization, or key in " cite$ * warning$
""
}
}
}

```

```

{ key sortify }
if$
}
{ "The " #4 organization chop.word sortify }
if$
}
{ editor sort.format.names }
if$
}
FUNCTION {presort}
{ type$ "book" =
type$ "inbook" =
or
'author.editor.sort
{ type$ "proceedings" =
'editor.organization.sort
{ type$ "manual" =
'author.organization.sort
'author.sort
if$
}
if$
}
if$
" "
*
year field.or.null sortify
*
" "
*
title field.or.null
sort.format.title
*
#1 entry.max$ substring$
'sort.key$ :=
}
ITERATE {presort}
SORT
STRINGS { longest.label }
INTEGERS { number.label longest.label.width }
FUNCTION {initialize.longest.label}
{ "" 'longest.label :=
#1 'number.label :=
#0 'longest.label.width :=
}
FUNCTION {longest.label.pass}
{ number.label int.to.str$ 'label :=
number.label #1 + 'number.label :=

```

```

label width$ longest.label.width >
{ label 'longest.label :=
label width$ 'longest.label.width :=
}
'skip$
if$
}
EXECUTE {initialize.longest.label}
ITERATE {longest.label.pass}
FUNCTION {begin.bib}
{ preamble$ empty$
'skip$
{ preamble$ write$ newline$ }
if$
"\begin{thebibliography}{ " longest.label * " }" * write$ newline$
}
EXECUTE {begin.bib}
EXECUTE {init.state.consts}
ITERATE {call.type$}
FUNCTION {end.bib}
{ newline$
"\end{thebibliography}" write$ newline$
}
EXECUTE {end.bib}

```