BIB – A Program for Formatting Bibliographies

Timothy A. Budd
revised by A. Dain Samples, May 1990

Bib is a program for collecting and formatting reference lists in documents. It is a preprocessor to the nroff/troff typesetting systems, (much like the *tbl* [7] and *eqn* [3] programs) and an alternative to the *refer* [6] and *tib* [2] bibliography programs. *Bib* takes two inputs: a document to be formatted and a library of references. Imprecise citations in the source document are replaced by more conventional citation strings, the appropriate references are selected from the reference file, and commands are generated to format both citation and the referenced item in the bibliography.

Bib has been enhanced to promote sharing of bibliographic references from the *refer*, *bib*, and Tib tools. *Bib* now accepts the Tib style of macro invocation, i.e. surrounding names that are to be expanded with vertical bars. If you or your group are not contemplating moving to the TeX or LaTeX [4, 5] text processors, then you need not worry about any of the references to Tib in this document. However, if you are considering such a move, and are worried about how you will convert your bibliographic databases to the BibTeX format, then the current version of *bib*, in conjunction with the existing Tib software[1] will make life a lot easier for you.

Description

An imprecise citation is a list of words surrounded by the characters [. and .]. Words (which are truncated to six letters) in the imprecise citation are matched against entries in the reference file, and if an entry is found that matches all words, that reference is used. For example:

In Brooks' interesting book [. brooks mythical.] various reasons ...

Multiple citations are indicated by simply placing a comma in the imprecise citation:

In [.kernig tools, kernig elements.], Kernighan and Plauger have ...

Embedded newlines, tabs and extra blanks within the imprecise citation are ignored.

Judicious use of the K (keyword) field in references in the database can simplify citations considerably. Also additional information can be placed into citations by surrounding text with curly braces. The additional information is inserted verbatim into the citation, e.g. [1, Chapter 6]. Note that it may be desirable to use non-breakable spaces, in order that the citation not be split across a line boundary by *troff*, for example:

For a description of LR parsing, see [.dragon {,\ Chapter 6}.] by Aho and Ullman.

The angle brackets can be used as alternatives to the curly braces.

For a description of LR parsing, see [.dragon <,\ Chapter 6>.] by Aho and Ullman.

1. Tib is available, for a handling charge, from James C. Alexander at the Dept. of Mathematics, University of Maryland, College Park, MD 20742. It is also available from **eneevax.umd.edu:pub/tib** via *ftp*.

An alternative citation style can be used by surrounding the imprecise citation with {. and .}. Most document styles just give the raw citation, without the braces, in this case. This is useful, for example, to refer to citations in running text.

For a discussion of this point, see reference {.dragon.}.

The algorithm used by *bib* scans the source input in two passes. In the first pass, references are collected and the location of citations marked. In the second pass, these marks are replaced by the appropriate citation, and the entire list of references is dumped following a call on the macro .[]. This macro is left untouched. However, this can be altered to achieve other typographic effects.

An exception to this process is made in those instances where references are indicated in footnotes. In this case the macro that generates the reference is placed immediately after each line in which the reference is cited.

Creating the Database

Reference files are prepared for *bib* using *invert*. By default *invert* places an inverted index for the reference list in the file INDEX. Unless the user specifies an alternative (see the −p switch described below), this is the first file searched by *bib* in attempting to locate a reference. If the entry is not found in the user's file, a standard system-wide index is searched. If the entry is still not found in the system file, a warning message is produced and a blank citation is generated.

The format for entries in the reference file is described more fully in the section 'Reference File Formats'.

Since the user's index is searched before the system index, if the user wants to alter a specific entry in the system index (say to change the name W. E. Howden to William E. Howden, for example) it is a simple matter to copy the system information into a private database and make the changes locally. *Bib* issues messages warning the user if there are multiple entries in a reference file that match an imprecise reference, or if there are multiple index files that match an imprecise reference. (Note that previous versions of *bib* stopped searching after the first match was found. Current users of *bib* may see warning messages now that they did not receive before.) The 'duplicate matches' warning message can be controlled by making the citations more precise.

Citation formats are either determined by explicit switch settings or, more generally, by using a predefined formatting style. In the latter form, usage looks something like:

bib −t*style* [files]

where *style* is a citation style. Currently the following citation styles are available:

astro	astrophysical journal style references
compsurv	Computing Surveys style references
foot	footnoted references.
hnf	Hanson Normal Form
jrnl	lists references in alphabetical and date order by journal; see Miscellaneous Tools section.
lib	useful for printing the entire bibliography; see Miscellaneous Tools section
list	a useful format for listrefs, which see.
llist	another useful format for printing everything about your bibliography database; see Miscellaneous Tools section
opena	same as stda, but using an open format.
openn	same as stdsn, only using an open reference format (each major entry is on a new line[1]).
spe	format used by the journal *Software—Practice and Experience*.

1. The open reference format is adapted from *A Handbook for Scholars*, by Mary-Claire van Leunen, published by Knopf, 1978.

spe2 a second format for *Software—Practice and Experience*.

stda standard alphabetic format, where citations are three letters followed by the last two digits of the date; for papers with a single author, the letters are the first three letters of the authors last name (e.g. Knu); in papers with two authors the first two letters are from the first author followed by one letter from the second (e.g. HoU); etc. (see the section **Citation Templates** under **Reference Format Desiner's Guide** for more details).

stdafull standard alphabetic format, with full references.

stdn standard numeric citation; reference entries are listed in order of mention.

stdsn same as stdn, but references are sorted by senior author followed by date.

supn same as stdn, but using superscripts.

It is possible to alter slightly the format of standard styles. For example, to generate references in standard numeric style with first names abbreviated:

bib −tstdn −aa ...

If two reference items create the same citation string (this can happen if two works by the same authors published in a single year are referred to in one paper) a disambiguating final letter is added to the citation (i.e., Knu79 becomes Knu79a and Knu79b). This can be altered by using the F field (see the section on Reference File Formats).

For the purposes of sorting by author, the last name is taken to be the last word of the name field. This means some care must be taken when names contain embedded blanks, such as in 'Hartley Rogers, Jr.' or 'Mary-Claire van Leunen'. In these cases a concealed space (\ ) should be used, as in 'Hartley Rogers,\ Jr.'.

bib knows very little about *troff* usage or syntax. This can sometimes be useful. For example, to cause an entry to appear in a reference list without having it explicitly cited in the text the citation can be placed in a *troff* comment.

```
.\" [.imprecise citation.]
```

It is also possible to embed *troff* commands within a reference definition. See 'Abbreviations' in the section 'Reference Format Designer's Guide' for an example. However, be aware that unbridled use of such embedded processor-specific commands makes it more difficult to convert to other processors later. In the section on Miscellaneous Tools, we discuss ways to avoid such specificity.

In some styles (superscripts) periods and commas should precede the citation while spaces follow. In other styles (brackets) these rules are reversed. If a period, comma or space immediately precedes a citation, it will be moved to the appropriate location for the particular reference style being used. This movement is not done for citations given in the alternative style (i.e. {.dragon.}).

The following is a complete list of options for *bib*:

−aa reduce author's first names to abbreviations.

−ar*num* reverse the first *num* author's names. If *num* is omitted all names are reversed.

−ax print authors last names in Caps-Small Caps style. For example Budd becomes BUDD.

−c*str* build citations according to the template *str*; see the section **Citation Templates** under **Reference Format Desiner's Guide** for the format of the string and its effect.

−d *dir*

−d*dir* change the base directory in which files are sought. It is initially /usr/new/lib/bmac.

−ea abbreviate editors' names

−ex places editors' names in Caps-Small Caps style. (see −x)

−er*num* reverse the first *num* editors' names. If *num* is omitted all editors' names are reversed.

−f instead of dumping references following the call on .[], dump each reference immediately following the line on which the citation is placed (used for footnoted references).

−h hyphenate runs of three or more contiguous references in the citation string. (eg 2,3,4,5 becomes 2-5). This is most useful for numeric citation styles, but works generally. The −h option implies the −o option.

−i file

−ifile include and process the indicated file. This is useful for including a private file of string definitions.

−n*str* turn off the indicated options. *str* must be composed of the characters *afhoRrvx*.

−o sort contiguous citations according to the order given by the reference list. (This option defaults on).

−p *file*

−p*file* instead of searching the file INDEX, search the indicated reference file(s) before searching the system file. Multiple files are separated by commas.

−R print a warning each time there is an attempt to redefine a name. (No warning is the default.)

−r*num* synonym for −ar.

−s*str* sort references according to the template *str*.

−t *type*

−t*type* use the standard macros and switch settings to generate citations and references in the indicated style.

−Tib use the Tib macro conventions. See the discussion in the Reference Format Designer's Guide.

−Tibx creates the file .bib.m4.in in the current directory. This file contains macro definitions that when applied to a database file, converts calls on macros to the form expected by the Tib bibliography preprocessor; more information is in the section Miscellaneous Tools.

−v **[[I'm not sure what this does; it is related to the −f option, but it apparently suppresses the printing of certain information.]]**

−x synonym for −ax.

When a file is to be included during normal *bib* processing (options −i and −t, and the style file command I) *bib* searches a specific set of directories and filenames. For example, if **-i myfile** is specified on the invocation line, *bib* attempts to open, in order, the following files until one is found.

1. *./myfile*
2. **BMACLIB***/myfile*, where **BMACLIB** is defined by the −d option (default: /usr/new/lib/bmac).
3. **BMACLIB***/tibmacs/myfile* if −Tib was specified, otherwise **BMACLIB***/bibmacs/myfile*
4. *./bib.myfile*
5. **BMACLIB***/bib.myfile.*

If none are found, an error message is issued, and execution halts.

Reference File Formats

A reference file contains any number of reference items. Reference items are separated by one or more blank lines. There are no restrictions placed on the order of reference items in a file, although the user will find that imposing some order simplifies updates.

A reference item is a collection of field tags and values. A field tag is a percent sign followed by a single letter. Currently, the following field tags are recognized:

A	Author's name †
B	Title of book containing item
C	City of publication
D	Date
E	Editor(s) of book containing item †
F	Caption
G	Government (NTIS) ordering number
I	Issuer (publisher)
J	Journal name
K	Keys for searching § †
N	Issue number
O	Other information § †
P	Page(s) of article
R	Technical report number
S	Series title
T	Title
V	Volume number
W	Where the item can be found locally § †
X	Annotations § †

Fields marked with † are *accumulated* fields and can be repeated as necessary, but for all others only the last occurence of the field in any reference will be used. Those fields marked with § are ignored by most styles designed for publication, but can have additional information and are available to database listing styles and other software tools. A field can be as long as necessary and can extend onto new lines. No continuation characters are necessary: lines that do not begin with a percent sign or a period and are not blank are treated as continuations of the previous line. The order of fields is irrelevant, except that accumulated fields are listed in the order of occurrence.

The format of the reference file for *bib* is similar to that used by *refer* except that *bib* has the following additional capabilities:

1. An F field, if present, overrides whatever citation string would otherwise be constructed.
2. Certain defined names can be used, and will be expanded differently by different document styles. For example, the string CACM is expanded into 'Communications of the ACM' by some document styles, 'Comm. ACM' by others, and 'Comm. of the Assoc. of Comp. Mach.' by yet others. Appendix 1 lists some of the currently recognized names.
3. The program automatically abbreviates names, reverses names, and hyphenates strings of contiguous references, if requested.
4. A reference can have more than one editor field, and editor's names can be abbreviated, reversed, and/or printed in cap/small caps style, independent of any processing done to authors names.
5. Comments (lines with '#' in the first column) can be interspersed throughout the reference file. They are not used as sources of keywords (use the %O or %X fields for that).

The format of the reference file for *bib* is also similar to that used by *tib*, which shares a lot of source code with *bib*, with the following exceptions:

1. *Bib* does not recognize field tags *M*, *Z*, *a* through *z* and \, while *tib* does.
2. *Tib* currently does not recognize field tag *X*, and *bib* does.

3. *Tib* currently does not have any form of comment lines in the reference file, while *bib* marks comment lines with a ’#’ at the beginning of a line.

4. *Troff* commands may be inserted verbatim into *bib* references. In *tib* the ‘%\’ field tag is used to insert TeX commands.

There may be (and probably are) other differences between *bib*, *tib*, and *refer* not documented here.

Generally a reference falls into one of several basic categories: book, journal article, conference paper, article in a book, compilations, technical report, PhD thesis, etc. An example of each and a brief comment is given below. With less standard references (Archival Sources, Correspondence, Government Documents, Newspapers) generally some experimentation is necessary.

Books

A book is something with a publisher that isn’t a journal article or a technical report. Generally, books also have authors and titles and dates of publication (although some don’t). For books not published by a major publishing house it is also helpful to give a city for the publisher. Some government documents also qualify as books, so a book may have a government ordering number.

It is conventional that the authors names appear in the reference in the same form as on the title page of the book. Note also that string definitions are provided for most of the major publishing houses (PRHALL for Prentice-Hall, for example). The string definition may include the city as part of the definition, depending on the database in use.

```
%A	R. E. Griswold
%A	J. F. Poage
%A	I. P. Polonsky
%T	The SNOBOL4 Programming Language
%I	PRHALL
%D	second edition 1971
```

Sometimes a book (particularly old books) will have no listed publisher. The reference entry must still have an I field.

```
%A	R. Colt Hoare
%T	A Tour through the Island of Elba
%I	(no listed publisher)
%C	London
%D	1814
```

If a reference database contains entries from many people (such as a departmental-wide database), the W field can be used to indicate where the referenced item can be found; using the initials of the owner, for example. Any entry style can take a W field, since this field is not used in formatting the reference.

The K field is used to define general subject categories for an entry. This is useful in locating all entries pertaining to a specific subject area. Note the use of the backslash to indicate the last name is Van Tassel, and not simply Tassel.

```
%A	Dennie Van\ Tassel
%T	Program Style, Design, Efficiency,
Debugging and Testing
%I	PRHALL
%D	1978
%W	tab
%K	testing debugging
```

Journal article

The only requirement for a journal article is that it have a journal name and a volume number. Usually journal articles also have authors, titles, page numbers, and a date of publication. They may also have numbers, and, less frequently, a publisher. (Generally, publishers are only listed for obscure journals).

Note that string names (such as CACM for *Communications of the ACM*) are defined for most major journals. There are also string names for the months of the year, so that months can be abbreviated to the first three (capital) letters. Note also in this example the use of the %K field to define a short name (hru) that can be used as a shorthand in an imprecise citation. (This is to be contrasted with BibTeX which not only *requires* user selected abbreviations, but also requires that they all be distinct from one another.)

```
%A  M. A. Harrison
%A  W. L. Ruzzo
%A  J. D. Ullman
%T  Protection in Operating Systems
%J  CACM
%V  19
%N  8
%P  461-471
%D  AUG 1976
%K  hru
```

Article in conference proceedings

An article from a conference is printed as though it were a journal article and the journal name was the name of the conference. Note that string names (SOSP, The Symposium on Operating System Principles) are also defined for the major conferences.

```
%A  M. Bishop
%A  L. Snyder
%T  The Transfer of Information and Authority
in a Protection System
%J  Proceedings of the 7th SOSP
%P  45-54
%D  1979
```

Article in book

An article in a book has two titles, the title of the article and the title of the book. The first goes into the T field and the second into the B field. Similarly the author of the article goes into the A field and the editor of the book goes into the E field.

```
%A  John B. Goodenough
%T  A Survey of Program Testing Issues
%B  Research Directions in Software Technology
%E  Peter Wegner
%I  MIT Press
%P  316-340
%D  1979
```

If a work as more than one editor, they each get their own %E field.

```
%A  R. J. Lipton
%A  L. Snyder
%T  On Synchronization and Security
%E  Richard A. DeMillo
%E  David P. Dobkin
```

```
%E	Anita K. Jones
%E	Richard J. Lipton
%B	Foundations of Secure Computation
%P	367-388
%I	ACPRESS
%D	1978
```

Sometimes the book is part of a multi-volume series, and hence may contain a volume field and/or a series name.

```
%A	C.A.R. Hoare
%T	Procedures and parameters: An axiomatic approach
%B	Symposium on semantics of algorithmic languages
%E	E. Engeler
%P	102-116
%S	Lecture Notes in Mathematics
%V	188
%I	Springer-Verlag
%C	Berlin-Heidelberg-New York
%D	1971
```

In any reference format, the O field can be used to give additional information. This is frequently used, for example, for secondary references.

```
%A	A. Girard
%A	J-C Rault
%T	A Programming Technique for Software Reliability
%B	Symposium on Software Reliability
%I	IEEE
%C	Montvale, New Jersey
%D	1977
%O	(Discussed in Glib [32])
```

Compilations

A compilation is the work of several authors gathered together by an editor into a book. The reference format is the same as for a book, with the editor(s) taking the place of the author.

```
%E	R. A. DeMillo
%E	D. P. Dobkin
%E	A. K. Jones
%E	R. J. Lipton
%T	Foundations of Secure Computation
%I	ACPRESS
%D	1978
```

Technical Reports

A technical report must have a report number. They usually have authors, titles, dates and an issuing institution (the I field is used for this). They may also have a city and a government issue number. Again string values (UATR for ‘University of Arizona Technical Report’) will frequently simplify typing references.

```
%A	T. A. Budd
%T	An APL Complier
%R	UATR 81-17
%C	Tucson, Arizona
%D	1981
```

If the institution name is not part of the technical report number, then the institution should be given separately.

```
%A	Douglas Baldwin
%A	Frederick Sayward
%T	Heuristics for Determining Equivalence of Program Mutations
%R	Technical Report Number 161
%I	Yale University
%D	1979
```

PhD Thesis

A PhD thesis is listed as if it were a book, and the institution granting the degree the publisher.

```
%A	Martin Brooks
%T	Automatic Generation of Test Data for
Recursive Programs Having Simple Errors
%I	PhD Thesis, Stanford University
%D	1980
```

Some authors prefer to treat Master's and Bachelor theses similarly, although most references on style instruct say to treat a Master's degree as an article or as a report.

```
%A	A. Snyder
%T	A Portable Compiler for the Language C
%R	Master's Thesis
%I	M.I.T.
%D	1974
```

Miscellaneous

A miscellaneous object is something that does not fit into any other form. It can have any of the the following fields; an author, a title, a date, page numbers, and, most generally, other information (the O field).

Any reference item can contain an F field, and the corresponding text will override whatever citation would otherwise be constructed.

```
%F	BHS--
%A	Timothy A. Budd
%A	Robert Hess
%A	Frederick G. Sayward
%T	User's Guide for the EXPER Mutation Analysis system
%O	(Yale university, memo)
```

Reference Format Designer's Guide

This section need only be read by those users who wish to write their own formatting macro packages.

The information necessary for generating citations and references of a particular style is contained in a *format file*. A format file consists of two parts; a sequence of format commands, which are read and interpreted by *bib*, and a sequence of text lines (usually *troff* macro definitions) which are merely copied to output. The format file name is always prefixed by 'bib.'. Thus the format file for a standard document type, such as stdn, is found in a file called bib.stdn.

Each formatting command is distinguished by a single letter, which must be the first character on a line. The formatting commands in a database file are similar to the command line options for *bib*. The legal commands, and their arguments, are as follows:

#*text* — A line beginning with a sharp sign is a comment, and all remaining text on the line is ignored.

A — The A command controls how authors' names are to be formatted. It can be followed by the following character sequences:

- A — Authors' names are to be abbreviated (see the section **Abbreviations**, below).
- R*num* — The first *num* authors' names are to be reversed. If *num* is omitted, all authors' names are reversed.
- X — Authors' names are to be printed in Caps-Small Caps style.

E — The E command is equivalent to the A command, except that it controls the formatting of editors' names.

F — The F command indicates that references are to be dumped immediately after a line containing a citation, such as when the references are to be placed in footnotes.

S *template* — The S command indicates references are to be sorted before being dumped. The comparison used in sorting is based on the *template*. See the discussion in the section **Sort Template** for an explanation of templates.

C *template* — The *template* is used as a model in constructing citations. See the discussion below in the section **Citation Templates**.

D *word definition* — The word-definition pair is placed into a table. Before each reference is dumped it is examined for the occurrence of these words. Any occurrence of a word from this table is replaced by the definition, which is then rescanned for other words. Words are limited to alphanumeric characters, ampersand and underscore. Definitions can extend over multiple lines by ending lines with a backslash (\). The backslash will be removed, and the definition, including the newline and the next line, will be entered into the table. This is useful for including several fields as part of a single definition (city names can be included as part of a definition for a publishing house, for example).

Bib has been enhanced to recognize macro calls where the macro name is surrounded by vertical |bars|. This enhancement was implemented to to provide a little more compatibility between *bib* and Tib, a preprocessor that uses a database format very similar to *bib*'s. To have *bib* recognize only macro names surrounded by vertical bars, invoke the −Tib option.

I *filename* — The indicated file is included at the current point. The included file may contain other formatting commands.

H — Three or more contiguous citations that refer to adjacent items in the reference list are replaced by a hyphenated string. For example, the citation 2,3,4,5 would be replaced by 2-5. This is most useful with numeric citations. The H option implies the O option.

O — Contiguous citations are sorted according to the order given by the reference list.

R *number* — The first *number* author's names are reversed on output (i.e. T. A. Budd becomes Budd, T. A.). If number is omitted all names are reversed.

T *str* The *str* is a list of field names. Each time a definition string for a named field is produced, a second string containing just the last character will also be generated. See the section **Trailing Characters** below.

X Authors' last names are to be printed in Caps/Small Caps format (i.e., Budd becomes BUDD).

**** NOTE ** The first line encountered in the format input that does not match a format command causes that line, and all subsequent lines, to be immediately copied to the output without further processing.**

File Naming Conventions

Standard database format files are kept in a standard library area. The string BMACLIB in bib.h points to this directory (/usr/new/lib/bmac in the distribution). In addition, this name is always defined when reading format files, and is defined with the l] macro when processing with *troff*. The first command output by *bib* defines the string l] to be this standard macro database directory. This allows macro files to be independent of where they are actually stored. There are three types of files:

bib.xxx These files contain bib commands to format documents in the xxx style.

bmac.xxx These files are the *troff* macros to actually implement a style. They are generally not examined by *bib* at all, but are processed by troff in response to a .so command.

bibinc.xxx These files contain information (such as definitions) used by more than one style database. The two usual files are bibinc.fullnames and bibinc.shortnames. Both of these include files bibinc.Xlocal (where X is full or short, respectively). Due to the way *bib* searches for files, if the user has these in his local directory, they will be read just before the contents of bibinc.Xnames are processed.

Troff Naming Conventions

There is a simple naming convention for strings, registers and macros used by *bib* during processing by *trof*. All strings, registers and macros are denoted by two character names containing either a left or right brace. The following are general rules:

[*x* If *x* is alphanumeric, the string contains the value of a reference field. If *x* is nonalphanumeric, this is a formatting string preceding a citation.

]*x* If *x* is alphanumeric, this is the final character from a reference field. If *x* is nonalphanumeric, the string is formatting information within a citation.

x[Strings in this format, where *x* is can be any character, are defined by the specific macro package in use and are not specified by *bib*.

x] If *x* is nonalphanumeric these strings represent formatting commands following citations (the inverse of [*x* commands). Other strings represent miscellaneous formatting commands, such as the space between leading letters in abbreviated names.

There are two such macro names to be particularly aware of: .s[and .e[. The first is called at the beginning of formatting a reference and the latter is called at the end of the formatting. The user can have some control over the formatting of references by redefining these macros.

It might be noted here that the best way to understand this stuff is to look at some existing definition files, and start making small changes here and there.

Sort Templates

The sort template is used in comparing two references to generate the sorted reference list. The sort template is a sequence of sort objects. Each sort object consists of an optional negative sign, followed by a field character, followed by an optional signed size. The leading negative sign, if present, specifies the sort is to be in decreasing order, rather than increasing. The field character indicates which field in the reference is to be compared. The entire field is used, except in the case of the 'A' field, in which case only the senior author's last name is used. A positive number following the field character indicates that only the first n characters are to be examined in the comparison. The negative value indicates only the last *n* characters. Thus, for example, the template AD−2 indicates that sorting is to be done by the senior author followed by the last two characters of the date.

The sort algorithm is stable, so that two documents which compare equally will be listed in citation order.

Note that in sorting, citation construction, and elsewhere, if an author field is not present the senior editor will be used. If neither author nor editor fields are present the institution name will be used.

Citation Templates

A citation template is similar to a sort template, with the following additions:

0 suppresses all printing. **[[True? can anyone tell me what this does?]]**

1 refers to the number which represents the position of the reference in the reference list (after sorting).

2 generates a three character sequence: if the paper being referenced has only one author, this is the first three characters of the author's last name; for two author papers, this is the first two characters of the senior author, followed by the first character of the second author; for papers with three or more authors the first letter of the first three authors is used.

3 used to specify a format consisting of the authors' last names, or the senior author followed by the text 'et al' if more than four authors are listed. This is the Astrophysical Journal style of citation.

4 the Computing Surveys style of citation.

8 full alphabetic.

9 last name of senior author.

Each object can be followed by either of the letters 'u' or 'l' and the field will be printed in all upper or all lower case, respectively.

If necessary for disambiguating, the character '@' can be used as a separator between objects in the citation template. Any text which should be inserted into the citation uninterpreted should be surrounded by either '{..}' or '<..>' pairs.

Citation Formatting in Troff

In the output, each citation is surrounded by the strings *([[and *(]] (or in the alternative style *([{ and *(}]). Multiple citations are separated by the string *(],. The text portion of a format file should contain *troff* definitions for these strings to achieve the appropriate typographic effect.

Citations that are preceded by a period, comma, space or other punctuation are surrounded by string values for formatting the punctuation in the appropriate location. Again, *troff* commands should be given to insure the appropriate values are produced.

The following table summarizes the string values that must be defined to handle citations.

[[	]]	Standard citation beginning and ending
{[	}]	Alternate citation beginning and ending
[.	.]	Period before and after citation
[,	,]	Comma before and after citation
[?	?]	Question mark before and after citation
[!	!]	Exclamation Point before and after citation
[:	:]	Colon before and after citation
[;	;]	Semi-Colon before and after citation
["	"]	Double Quote before and after citation
['	']	Single Quote before and after citation
[<	>]	Space before and after citation
],		Multiple citation separator
]-		Separator for a range of citations

Name Formatting in Troff

Authors' (and editors') names can be abbreviated, reversed, and/or printed in Caps-small Caps format. In producing the string values for an author, formatting strings are inserted to give the macro writer greater flexibility in producing the final output. Currently the following strings are used:

a]	gap between successive initials
b]	comma between last name and initial in reversed text
c]	comma between authors
n]	*and* between two authors
m]	*and* between last two authors
p]	period following initial

For example, suppose the name ‘William E. Howden’ is abbreviated and reversed. It will come out looking like

```
Howden\*(b]W\*(p]\*(a]E\*(p]
```

Reference Formatting in Troff

The particular style used in printing references is decided by macros passed to *troff*. Basically, for each reference, *bib* generates a sequence of string definitions, one for each field in the reference, followed by a call on the formatting macro. For example an entry which in the reference file looks like:

```
%A	M. A. Harrison
%A	W. L. Ruzzo
%A	J. D. Ullman
%T	Protection in Operating Systems
%J	CACM
%V	19
%N	8
%P	461-471
%D	1976
%K	hru
```

is converted into the following sequence of *troff* commands

```
.[−
.ds [F 1
.ds [A M\*(p]\*(a]A\*(p] Harrison
.as [A \*(c]W\*(p]\*(a]L\*(p] Ruzzo
.as [A \*(m]J\*(p]\*(a]D\*(p] Ullman
.ds [T Protection in Operating Systems
.ds [J Communications of the ACM
.ds [V 19
.ds [N 8
.nr [P 1
.ds [P 461-471
.ds [D 1976
.ds [K hru
.][
```

Note that the commands are preceded by a call on the macro ‘.[−’. This can be used by the macro routines for initialization, for example to delete old string values. (For some reason, the ending macro .e[also calls .[which seems redundant.)

The string [F is the citation string used in the document. Note that the string CACM has been expanded.

The strings c], n] and m] are used to separate authors. c] separates the initial authors in multi-author documents (it is usually a comma with some space before and after), n] separates authors in two author documents (usually ‘ and ’), and m] separates the last two authors in multi-author documents (either ‘ and ’ or ‘, and ’).

If abbreviation is specified, the string a] is used to separate initials in the author’s first name, and the string p] determines the punctuation to use (usually a period).

The *bib* system provides minimal assistance in deciding format types. For example note that the number register [P has been set of 1, to indicate that the article is on more than one page. Similarly, in documents with editors, the register [E is set to the number of editors.

Trailing Characters

There is a problem with fields that end with punctuation characters causing multiple occurrences of those characters to be printed. For example, suppose author fields are terminated with a period, as in T. A. Budd. If names are reversed, this could be printed as Budd, T. A.. Even if names are not reversed, abbreviations, such as in Jr. can cause problems.

To avoid this problem *bib*, if instructed, generates the last character from a particular field as a separate string. The string name is a right brace followed by the field character (e.g.]A for the author field). Macro packages should test this value before generating punctuation.

Abbreviations

The algorithm used to generate abbreviations from first names is fairly simple: Each word in the first name field that begins with a capital is reduced to that capital letter followed by a period. In some cases, this may not be sufficient. For example, suppose Ole-Johan Dahl should be abbreviated ‘O−J. Dahl’. The only way to achieve this (short of editing the output) is to include *troff* commands in the reference file that alter the strings produced by *bib*, as in the following

```
...
%A Ole-Johan Dahl
.ds [A O−J. Dahl
...
```

In fact, if absolutely necessary, any *troff* commands can be entered in the middle of a reference entry, and the commands are copied uninterpreted to the output. For example, the user may wish to have a switch indicating whether the name is to be abbreviated or not:

```
...
%A Ole-Johan Dahl
.if \n(i[ .ds [A O−J. Dahl
...
```

However, keep in mind the restrictions this imposes on the portability and convertibility of the database.

An Example

Figure 1 shows the format file for the standard alphabetic format. The I command includes a file of definitions for common strings, such as dates and journal names. A portion of this file is shown in figure 2. The sort command indicates that sorting is to be done by senior author, followed by the last two digits of the date. The citation template indicates that citations will be a three character sequence followed by the last two characters of the date (e.g. AHU79) (described in the section on citations above). Author and editor names will be abbreviated. The .so command causes *troff* to read a basic set of macro definitions for formatting the references; the beginning of this file is shown in figure 3; note that a no-op has been inserted into the definition string for BIT in order to avoid further expansion when the definition is rescanned. Finally, the *troff* macro s[is redefined to turn off hyphenation across line-boundaries, and to make sure the citation is put in the bibliography list indented 10 ens and surrounded by square brackets.

```
#  standard alphabetic format
I bibinc.shortnames
SAD−2
C2D−2
AA
EA
 .so \*(l]/bmac.std
 .de s[\" start reference by turning off hyphenation
 .nh
 .IP [\\*([F] 10n     \" indent the citation
 ..
```

Figure 1

```
D ACTA Acta Informatica
D BIT B\&IT
D CACM Communications of the ACM
 ...
D JAN Jan.
 ...
D DEC Dec.
```

Figure 2

```
.\" standard format troff commands
.\" citation formatting strings
.ds [[ [
.ds ]] ]
.ds ], ,\|
.ds ]- -
.ds [. " \&
.ds .] .
.ds [, " \&
.ds ,] ,
  ...
.\" reference formating strings
.ds a] " \&
.ds b] , \&
.ds c] , \&
.ds n] "\& and \&
.ds m] "\& and \&
.ds e] \\fIet al.\\fP
.ds p] .
  ...
.\" reference formating macros
.de s[   \" start reference
.nh
.IP "[\*([F]" 5m
..
.de e[   \" end reference
.[–
..
.de []   \" start to display collected references
.LP
..
```

Figure 3

```
.de ][   \" choose format
.ie !"\\*([J"" \{\
.   ie !"\\*([V"" .nr t[ 1    \" journal
.   el            .nr t[ 5    \" conference paper
.\}
.el .ie !"\\*([B"" .nr t[ 3    \" article in book
.el .ie !"\\*([R"" .nr t[ 4    \" technical report
.el .ie !"\\*([I"" .nr t[ 2    \" book
.el               .nr t[ 0    \" other
.\\n(t[[
..
```

Figure 4

On the basis of some simple rules (the presence or absence of certain fields) the document is identified as one of five different types, and a call made on a different macro for each type. This is shown in figure 4. Note that how the reference is printed (as a book, a journal article reference, etc.) is not determined by *bib*, but by the .][macro invoked by *troff*.

Finally figure 5 shows the macro for one of those different types, in this case the formatting macro for references to books.

```
.de 2[ \" book
.s[
.ie !"\\*([A"" \\*([A,
.el .if !"\\*([E"" \{\
.	ie \\n([E-1 \\*([E, eds.,
.	el \\*([E, ed.,\}
.if !"\\*([T"" \\fI\\*([T\\fP,
.rm a[
.if !"\\*([I"" .ds a[ \\*([I
.if !"\\*([S"" , \\*([S\c	\" book in a series
.if !"\\*([C"" \{\
.	if !"\\*(a["" .as a[ , \\&
.	as a[ \\*([C\}
.if !"\\*([D"" \{\
.	if !"\\*(a["" .as a[ , \\&
.	as a[ \\*([D\}
\\*(a[.
.if !"\\*([G"" Gov. ordering no. \\*([G.
.if !"\\*([O"" \\*([O.
.e[
..
```

Figure 5

Miscellaneous Tools

Bibinc

Some editors require journal names in a reference to be spelled out completely, while other journals expect a standardized shortened version. *Bib* is able to accomodate these requirements by reading different different definition files. For example, standard open alphabetic format (see **BMACLIB**/*bib.opena*) expects a definition file called *bibinc.fullnames*, while **BMACLIB**/*bib.spe* reads *bibinc.shortnames*. Maintaining these two files and making sure they are consistent with one another can be tedious. A program is included with *bib* to ease this problem. It allows the user to keep a single file with all definitions, and to have those definitions sent to the appropriate file(s). In what follows we assume that the name of this common file, the input file to *bibinc*, is called *bibinc.names*.

The format of the lines in *bibinc.names* is formally:

```
<inputline> ::= '+'<char1> <filename>
            | '+'<char1> '+'<char2> '+'<char3> ...
            | '+'<char> <name> <stuff>
            | '+'<char><stuff>
            | '?'<name>
```

In all cases the '+' or '?' must be the first character of a line to be recognized, and there can be no spaces between it and the following character. Continuation lines are all lines from the preceding '+' to the following '+'. Any and all whitespace at the beginning of a continuation line is discarded.

The first form defines <char1> to be the character that denotes an outputfile named <filename>. It must be the first occurence of <char1> following a '+' in the file.

The second form defines <char1> to be the character that denotes several previously defined outputfiles. It must be the first occurence of '+'<char1> in the file.

The third form says that the following line is to be written to the file or files denoted by <char> as:

```
D <name> <stuff>
```

The fourth form allows the inclusion of arbitrary stuff into the file. Note that there is no space between the <stuff> and <char> in the fourth form.

The fifth form is used to provide some mechanism to select lines to be processed according to whether one is intending to use *tib* or *bib*, and whether one is intending to use the TeX or the *troff* processors. The problem to be solved is that we want to be able to create files for three possibilities:

(1) the user is using bib-style macros with troff,

(2) the user is using tib-style macros with troff,

(3) the user is using tib-style macros with TeX or LaTeX.

Therefore, the user can type the invocation line as:

```
bibinc bib troff  <bibinc.names
bibinc bib        <bibinc.names     (troff implied)
bibinc tib troff  <bibinc.names
bibinc tib tex    <bibinc.names
bibinc tex        <bibinc.names     (tib implied)
```

(Note that 'bibinc bib tex' is illegal: not a supported combination.) If a line of the form "?tib" is encountered in *bibinc.names*, then the lines following that line are processed only if "tib" was specified or implied on the invocation line. This restriction remains true until a line beginning "?bib" or "?" is encountered. Likewise, "?troff" will permit the processing of the following lines only if troff was specified or implied on the invocation line, and this remains true until "?tex" or "?" is encountered.

More on Tib style macro invocation

Tib macro mechanism is quite different from the *bib* macro style. In *bib*, you define and use a macro as follows:

```
D macro expansion text
        :
%A A. Nonymous
%T macro on macro
```

The title will be expanded to ‘expansion text on expansion text’. In *tib*, macro calls are always enclosed in vertical bars:

```
D macro expansion text
    :
%A A. Nonymous
%T The macro as |macro|
```

The title will be expanded to ‘The macro as expansion text’, therewith demonstrating a major benefit of using the vertical |bars|. Here is another benefit of the bars:

```
D u_um \"{u}
D Karlsruhe Karlsr|u_um|he
```

It’s easy now to get the diacritical marks right, and still have a recognizable word to act as a key in the INDEX. It is possible to acheive this same effect with *bib*-style macros by using the non-printing space ‘\&’.

```
D u_um \*(:u
D Karlsruhe Karlsr\&u_um\&he
```

A Bibinc Example

The |Karlsruhe| example has a problem if your database is being used for both TeX and ditroff: bib/tib macros have no conditionals, and you have to choose either the TeX code or the troff code for special characters. Bibinc has a mechanism to solve this. Let’s assume you have the following in a file called bibinc.names:

```
================================= bibinc.names
# first line of bibinc.names
#
# first use of a letter after ‘+’ constitutes its definition
#
#  define the letter F to mean ’write this line into the fullnames file’
?bib
?troff
+F bibinc.fullnames
+S bibinc.shortnames
+B +F +S        # both
?tib
?troff
+F bibinc-t.fullnames
+S bibinc-t.shortnames
?tex
+F tibinc.fullnames
+S tibinc.shortnames
?               # turns off specialization
+B +F +S        # both
#
?tex
+B a_um \"{a}
```

```
+B o_sl \o
+B o_um \"{o}
+B u_um \"{u}
+B TCOLADA $\mbox{TCOL}_\mbox{Ada}$
+B dash --
+B Rn  $\mbox{R}^\mbox{n}$
+B AMP \\&
?troff          # either bib or tib
+B a_um \*(:a
+B o_sl \*(/o
+B o_um \*(:o
+B u_um \*(:u
+B TCOLADA TCOL\dAda\u
+B dash \(hy
+B Rn  \*(Rn
+B AMP &
?
+B fuer f|u_um|r
#
+B IFI   Institut |fuer| Informatik
+B Universitat Universit|a_um|t
+B KARLSRUHE Karlsr|u_um|he
+F Karlsruhe  |IFI|, |Universitat| |KARLSRUHE|
   %C |KARLSRUHE|, West Germany        # note that leading blanks are removed
+S Karlsruhe  |IFI|, |Universitat| |KARLSRUHE|
+F SIGPLAN SIG\&PLAN Notices
+S SIGPLAN SIG\&PLAN
+B GUNS Smith |AMP| Wesson
# last line of bibinc.names
================================== bibinc.names
```

This input to bibinc will select the correct expansion for u_um depending on whether -Tib was specified on its invocation line or not. That is, with the following invocation:

```
% bibinc troff tib <bibinc.names
```

two files will be written, bibinc-t.fullnames and bibinc-t.shortnames:

```
================================== bibinc-t.fullnames
D a_um \*(:a
D o_sl \*(/o
D o_um \*(:o
D u_um \*(:u
D TCOLADA TCOL\dAda\u
D dash \(hy
D Rn  \*(Rn
D AMP &
D fuer f|u_um|r
D IFI   Institut |fuer| Informatik
D Universitat Universit|a_um|t
D KARLSRUHE Karlsr|u_um|he
D Karlsruhe  |IFI|, |Universitat| |KARLSRUHE|\
%C |KARLSRUHE|, West Germany
D SIGPLAN SIGPLAN Notices
================================== bibinc-t.fullnames
```

and

```
================================== bibinc-t.shortnames
D a_um \*(:a
D o_sl \*(/o
D o_um \*(:o
D u_um \*(:u
D TCOLADA TCOL\dAda\u
D dash \(hy
D Rn  \*(Rn
D AMP &
D fuer f|u_um|r
D IFI   Institut |fuer| Informatik
D Universitat Universit|a_um|t
D KARLSRUHE Karlsr|u_um|he
D Karlsruhe  |IFI|, |Universitat| |KARLSRUHE|
D SIGPLAN SIGPLAN
================================== bibinc-t.shortnames
```

Users of bib/ditroff will have to specify -Tib on their invocations of bib to use these files.

The command:

```
%bibinc bib troff <bibinc.names
```

produces

```
================================== bibinc.fullnames
D a_um \*(:a
D o_sl \*(/o
D o_um \*(:o
D u_um \*(:u
D TCOLADA TCOL\dAda\u
D dash \(hy
D Rn  \*(Rn
D AMP &
D fuer f\&u_um\&r
D IFI   Institut fuer Informatik
D Universitat Universit\&a_um\&t
D KARLSRUHE Karlsr\&u_um\&he
D Karlsruhe  IFI, Universitat KARLSRUHE\
%C KARLSRUHE, West Germany
D SIGPLAN SIG\&PLAN Notices
================================== bibinc.fullnames
```

and

```
================================== bibinc.shortnames
D a_um \*(:a
D o_sl \*(/o
D o_um \*(:o
D u_um \*(:u
D TCOLADA TCOL\dAda\u
D dash \(hy
D Rn  \*(Rn
D AMP &
D fuer f\&u_um\&r
D IFI   Institut fuer Informatik
D Universitat Universit\&a_um\&t
D KARLSRUHE Karlsr\&u_um\&he
```

```
D Karlsruhe  IFI, Universitat KARLSRUHE
D SIGPLAN SIG\&PLAN
================================== bibinc.shortnames
```

In this mode, users of bib/ditroff will have to be very careful that their macro names do not conflict with anything in normal text.

The command:

```
%bibinc tex tib <bibinc.names
```

produces:

```
================================== tibinc.fullnames
D a_um \"{a}
D o_sl \o
D o_um \"{o}
D u_um \"{u}
D TCOLADA $\mbox{TCOL}_\mbox{Ada}$
D dash --
D Rn  $\mbox{R}^\mbox{n}$
D AMP \&
D fuer f|u_um|r
D IFI   Institut |fuer| Informatik
D Universitat Universit|a_um|t
D KARLSRUHE Karlsr|u_um|he
D Karlsruhe  |IFI|, |Universitat| |KARLSRUHE|\
%C |KARLSRUHE|, West Germany
D SIGPLAN SIGPLAN Notices
================================== tibinc.fullnames
```

and

```
================================== tibinc.shortnames
D a_um \"{a}
D o_sl \o
D o_um \"{o}
D u_um \"{u}
D TCOLADA $\mbox{TCOL}_\mbox{Ada}$
D dash --
D Rn  $\mbox{R}^\mbox{n}$
D AMP \&
D fuer f|u_um|r
D IFI   Institut |fuer| Informatik
D Universitat Universit|a_um|t
D KARLSRUHE Karlsr|u_um|he
D Karlsruhe  |IFI|, |Universitat| |KARLSRUHE|
D SIGPLAN SIGPLAN
================================== tibinc.shortnames
```

Notice how bibinc handles the bib/troff use of the special character sequence \& in the above definitions of the SIGPLAN macro. I.e., bibinc removes all user inserted \&'s when tib/tex is specified. NOTICE that when tib/tex is specified, any other \<char> is passed through exactly as is, unless <char> is \, in which case bibinc reduces it to a single \. Otherwise, all other \<char> are passed through as is.

An Example Makefile

Included with the distribution of *bib* is an example makefile that provides some hints on how to print out one's bibliographic database. The file is **BMACLIB**/Makefile.e.g., with a final period. For example, by typing

make -f Makefile.e.g. KEYWORD=cache print

you will get a listing of all entries in your database that have the keyword *cache* in them. Typing

make -f Makefile.e.g. bib

a complete listing of the database is produced. And finally,

make -f Makefile.e.g. KEYWORD=Ineed2read journals

will list all entries which have the keyword *Ineed2read*, presumably put in the %K field of references the user wants to read. Moreover, the listing is then sorted by the %J (journal) field, which is a convenient ordering for organizing your trips to the library.

Converting a database to a |database|

If at any point in the future you intend to begin using the TeX or LaTeX text processors, you will face the problem of converting your existing database files either to BibTeX format or to Tib format. Use of the -Tibx option to *bib* can simplify the conversion to *tib*. There is a public domain program called *tr2tex* which may also help convert existing documents; however, it is not included with the distribution of *bib*. Try contacting Kamal Al-Yahya at kamal@hanauma.stanford.edu. If that doesn't work try kamal%hanauma@score.stanford.edu.

To convert a database, run your usual sequence of commands to create your document using *bib*, except add the option -Tibx to *bib*'s command line. After the creation of your document, will also find the file *bib.m4.in which contains a set of m4 macro definitions. For each database file INDEX you wish to convert, type*

% bib2tib INDEX

and that's it! Well, almost. At this point your database files have had almost all macro calls converted to the vertical bar style. The ones that have not been caught are those that have ampersands in them: bib allows ampersands in macro names, but the m4 macro processor does not. Therefore, you will need to chase those down by hand.

Also, this has done nothing to remove any *troff* specific commands that may be embedded in you source files: they, too, will need to be changed *but only if you are converting to TeX*. If you are staying with *troff* for a while, or even for a long time, there is nothing more you need do.

Acknowledgements

bib was inspired by *refer*, written by M. Lesk. Enhancements to support *tib* style macros were discussed in depth with Michael van De Vanter and Ethan Munson. However, they did not do the implementation: blame that on Dain Samples.

1. A. V. Aho and J. D. Ullman, *Principles of Compiler Design*, Addison-Wesley, 1977.
2. J. C. Alexander, *Tib: A TeX bibliographic preprocessor*, (draft), 1987.
3. B. W. Kernighan and L. L. Cherry, A System for Typesetting Mathematics, *CACM 18*, 3 (MAR 1978), 151-156.
4. D. Knuth, *The TeXbook*, ADDISON, 1984.
5. L. Lamport, *LaTeX: A Document Preparation System*, ADDISON, 1986.
6. M. E. Lesk, Some Applications of Inverted Indexes on the UNIX System, Bell Laboratories Computing Science Technical Report 69, JUN 1978.
7. M. E. Lesk, Tbl - A Program to Format Tables, *Unix Programmer's Manual, Vol 2A*, .

APPENDIX 1

Standard Names

The following list gives some of the standard names recognized in most citation styles. Various different forms for the output are used by the different styles. In the longer reference style, the conference proceedings will also refer to the date (%D), city (%C), and when the proceedings are published as a journal, the journal name (%J), volume (%V) and number (%N). Not all names are listed here. See BMACLIB/bibmacs/bibinc.fullnames for a complete list.

Miscellaneous Names

a_um	ä
o_um	ö
u_um	ü
A_um	Ä
O_um	Ö
U_um	Ü
o_sl	ø
AMP	&
dash	-
ARCH	Architecture
COMP	Computer
COMPs	Computers
COMPg	Computing
CONF	Conference
CORP	Corporation
CS	Computer Science
CSD	Computer Science Department
DCS	Department of Computer Science
DEPT	Department
DISS	Dissertation
ENG	Engineering
EE	Electrical Engineering
fuer	für
INTL	International
JOUR	Journal
LANG	Language
LANGs	Languages
NATL	National
PHD	PhD Dissertation
PROC	Proceedings
PROG	Program
PRing	Programming
PROCSIGPLAN	Proceedings of the ACM-SIGPLAN
SCI	Science
SCIs	Sciences
SOFT	Software
SYMP	Symposium
SYS	System
SYSs	Systems
TRANS	Transactions
TR	Technical Report
UNIV	University
UNIVCA	University of California

Journal Names

ACMCS	ACM Computing Surveys
ACTA	Acta Informatica
AT&TTJ	AT&T Technical Journal

BIT	BIT
BSTJ	Bell System Technical Journal
CACM	Communications of the ACM
CAN	Computer Architecture News
COMPJOUR	The Computer Journal
COMPLANG	Journal of Computer Languages
COMPSUR	ACM Computing Surveys
HCI	Human-Computer Interaction
HUMFACT	Human Factors
I&C	Information and Control
IBMJRD	IBM Journal of Research and Development
IBMSJ	IBM Systems Journal
IEEEC	IEEE Computer
IEEEIT	IEEE Transactions on Information Theory
IEEEM	IEEE Micro
IEEESW	IEEE Softare
IEEETC	IEEE Transactions on Computers
IEEETIT	IEEE Transactions on Information Theory
IEEETSE	IEEE Transactions on Softare Engineering
IEEETSMC	IEEE Transactions on Systems, Man, and Information Sciences
IJCAI	International Joint Conference on Artificial Intelligence
IJCIS	International Journal of Computer and Information Sciences
IJCM	International Journal of Computer Mathematics
IJMMS	International Journal of Man-Machine Studies
IPL	Information Processing Letters
JACM	Journal of the ACM
JALG	Journal of Algorithms
JCSS	Journal of Computer and System Sciences
JPhil	Journal of Philosophy
JS&S	Journal of Systems and Softare
MATHST	Mathematics Systems Theory
NMATH	Numerical Mathematics
PerfEv	Performance Evaluation
SIAMJAM	SIAM Journal on Applied Mathematics
SIAMJC	SIAM Journal on Computing
SIAMJNA	SIAM Journal on Numerical Analysis
SIGACT	SIGACT News
SIGCHI	ACM SIGCHI Bulletin
SIGPLAN	SIGPLAN Notices
SIGSOFT	Softare Engineering Notes
SP&E	Softare−Practice & Experience
TIPSJ	Transactions of the Information Processing Society of Japan (Japanese)
TOCS	ACM Transactions on Computer Systems
TODS	ACM Transactions on Database Systems
TOMS	ACM Transactions on Mathematical Softare
TOOIS	ACM Transactions on Office Information Systems
TOPLAS	ACM Transactions on Programming Languages and Systems
TSE	IEEE Transactions on Softare Engineering

Selected Conferences

ADA80	PROC of the ACM-SIGPLAN Symposium on the Ada Programming Language, SIGPLAN
ASPLOS82	PROC of the SYMP on Architectural Support for Programming Languages and Operating Systems, SIGPLAN
CCC79	PROC of the SIGPLAN 1979 SYMP on Compiler Construction, SIGPLAN
CCC82	PROC of the SIGPLAN 1982 SYMP on Compiler Construction, SIGPLAN
CCC84	PROC of the SIGPLAN 1984 SYMP on Compiler Construction, SIGPLAN
CONF	Conference
FJCC	Fall Joint Computer Conference
FOCS	Annual SYMP on Foundations of Computer Science

HICSS	Hawaii International CONF on System Science
ICSE	International CONF on Software Engineering
JER3	PROC Third Jerusalem CONF on Information Technology
JICAI	Joint International CONF on Artificial Intelligence
PLISS83	PROC SIGPLAN 1983 SYMP on Programming Language Issues in Software Systems, SIGPLAN
POPL	ACM SYMP on Principles of Programming Languages
POPL5	Conference Record of the Fifth POPL
POPL6	Conference Record of the Sixth POPL
POPL7	Conference Record of the Seventh POPL
POPL8	Conference Record of the Eighth POPL
POPL9	Conference Record of the Ninth POPL
POPL10	Conference Record of the Tenth POPL
POPL11	Conference Record of the Eleventh POPL
PROC	Proceedings
SOSP	SYMP on Operating System Principles
STOC	Annual ACM SYMP on Theory of Computing
SYMP	Symposium
WJCC	PROC Western Joint Computer CONF

Place Names

BTLHO	Bell Laboratories
BTLMH	Bell Laboratories
CMU	Carnegie-Mellon University
CMUCS	Computer Science Department, Carnegie-Mellon University
DG	Data General
MITAI	MIT Artificial Intelligence Laboratory
MITLCS	MIT Laboratory for Computer Science
SUCS	Computer Science Department, Stanford University
SUCSL	Computer Systems Lab., Stanford Electronics Lab., Dept. of Electrical Engineering and Computer Science
SUEE	Department of Electrical Engineering, Stanford University
TUM	Technische Universität München
UCB	University of California, Berkeley
UCBCS	Computer Science Division, EECS, UCB
UCBERL	ERL, EECS, UCB

Months of the year

JAN	January
FEB	February
MAR	March
APR	April
MAY	May
JUN	June
JUL	July
AUG	August
SEP	September
OCT	October
NOV	November
DEC	December

Publishers

ACADEMIC	Academic Press
ACPRESS	Academic Press
ADDISON	Addison Wesley
ANSI	American National Standards Institute
CSPRESS	Computer Science Press
DIGITAL	Digital Press

ELSEVIER	American Elsevier
FREEMAN	W. H. Freeman and Company
GPO	U. S. Government Printing Office
HOLT	Holt, Rinehart, and Winston
IEEEP	IEEE Press
MCGRAW	McGraw-Hill
MGHILL	McGraw-Hill
MITP	MIT Press
NHOLL	North-Holland
NYC	New York, NY
PRENTICE	Prentice Hall
PRHALL	Prentice Hall
SPRINGER	Springer Verlag
SRA	Science Research Associates
WILEY	John Wiley & Sons
WINTH	Winthrop Publishers