

# Refer — A Bibliography System

*Bill Tuthill*

Computing Services  
University of California  
Berkeley, CA 94720

## *ABSTRACT*

**Refer** is a bibliography system that supports data entry, indexing, retrieval, sorting, runoff, convenient citations, and footnote or endnote numbering. This document assumes you know how to use some Unix editor, and that you are familiar with the **nroff/troff** text formatters.

The **refer** program is a preprocessor for **nroff/troff**, like **eqn** and **tbl**, except that it is used for literature citations, rather than for equations and tables. Given incomplete but sufficiently precise citations, **refer** finds references in a bibliographic database. The complete references are formatted as footnotes, numbered, and placed either at the bottom of the page, or at the end of a chapter.

A number of ancillary programs make **refer** easier to use. The **addbib** program is for creating and extending the bibliographic database; **sortbib** sorts the bibliography by author and date, or other selected criteria; and **roffbib** runs off the entire database, formatting it not as footnotes, but as a bibliography or annotated bibliography.

Once a full bibliography has been created, access time can be improved by making an index to the references with **indxbib**. Then, the **lookbib** program can be used to quickly retrieve individual citations or groups of citations. Creating this inverted index will speed up **refer**, and **lookbib** will allow you to verify that a citation is sufficiently precise to deliver just one reference.

# Refer — A Bibliography System

*Bill Tuthill*

Computing Services  
University of California  
Berkeley, CA 94720

## Introduction

Taken together, the **refer** programs constitute a database system for use with variable-length information. To distinguish various types of bibliographic material, the system uses labels composed of upper case letters, preceded by a percent sign and followed by a space. For example, one document might be given this entry:

```
%A Joel Kies
%T Document Formatting on Unix Using the -ms Macros
%I Computing Services
%C Berkeley
%D 1980
```

Each line is called a field, and lines grouped together are called a record; records are separated from each other by a blank line. Bibliographic information follows the labels, containing data to be used by the **refer** system. The order of fields is not important, except that authors should be entered in the same order as they are listed on the document. Fields can be as long as necessary, and may even be continued on the following line(s).

The labels are meaningful to **nroff/troff** macros, and, with a few exceptions, the **refer** program itself does not pay attention to them. This implies that you can change the label codes, if you also change the macros used by **nroff/troff**. The macro package takes care of details like proper ordering, underlining the book title or journal name, and quoting the article's title. Here are the labels used by **refer**, with an indication of what they represent:

```
%H Header commentary, printed before reference
%A Author's name
%Q Corporate or foreign author (unreversed)
%T Title of article or book
%S Series title
%J Journal containing article
%B Book containing article
%R Report, paper, or thesis (for unpublished material)
%V Volume
%N Number within volume
%E Editor of book containing article
%P Page number(s)
%I Issuer (publisher)
%C City where published
%D Date of publication
%O Other commentary, printed at end of reference
%K Keywords used to locate reference
%L Label used by -k option of refer
%X Abstract (used by roffbib, not by refer)
```

Only relevant fields should be supplied. Except for %A, each field should be given only once; in the case

of multiple authors, the senior author should come first. The %Q is for organizational authors, or authors with Japanese or Arabic names, in which cases the order of names should be preserved. Books should be labeled with the %T, not with the %B, which is reserved for books containing articles. The %J and %B fields should never appear together, although if they do, the %J will override the %B. If there is no author, just an editor, it is best to type the editor in the %A field, as in this example:

```
%A  Bertrand Bronson, ed.
```

The %E field is used for the editor of a book (%B) containing an article, which has its own author. For unpublished material such as theses, use the %R field; the title in the %T field will be quoted, but the contents of the %R field will not be underlined. Unlike other fields, %H, %O, and %X should contain their own punctuation. Here is a modest example:

```
%A  Mike E. Lesk
%T  Some Applications of Inverted Indexes on the Unix System
%B  Unix Programmer's Manual
%I  Bell Laboratories
%C  Murray Hill, NJ
%D  1978
%V  2a
%K  refer mkey inv hunt
%X  Difficult to read paper that dwells on indexing strategies,
    giving little practical advice about using \fBrefer\fp.
```

Note that the author's name is given in normal order, without inverting the surname; inversion is done automatically, except when %Q is used instead of %A. We use %X rather than %O for the commentary because we do not want the comment printed all the time. The %O and %H fields are printed by both **refer** and **roffbib**; the %X field is printed only by **roffbib**, as a detached annotation paragraph.

### Data Entry with Addbib

The **addbib** program is for creating and extending bibliographic databases. You must give it the filename of your bibliography:

```
% addbib database
```

Every time you enter **addbib**, it asks if you want instructions. To get them, type **y**; to skip them, type RETURN. **Addbib** prompts for various fields, reads from the keyboard, and writes records containing the **refer** codes to the database. After finishing a field entry, you should end it by typing RETURN. If a field is too long to fit on a line, type a backslash (\) at the end of the line, and you will be able to continue on the following line. Note: the backslash works in this capacity only inside **addbib**.

A field will not be written to the database if nothing is entered into it. Typing a minus sign as the first character of any field will cause **addbib** to back up one field at a time. Backing up is the best way to add multiple authors, and it really helps if you forget to add something important. Fields not contained in the prompting skeleton may be entered by typing a backslash as the last character before RETURN. The following line will be sent verbatim to the database and **addbib** will resume with the next field. This is identical to the procedure for dealing with long fields, but with new fields, don't forget the % key-letter.

Finally, you will be asked for an abstract (or annotation), which will be preserved as the %X field. Type in as many lines as you need, and end with a control-D (hold down the CTRL button, then press the "d" key). This prompting for an abstract can be suppressed with the **-a** command line option.

After one bibliographic record has been completed, **addbib** will ask if you want to continue. If you do, type RETURN; to quit, type **q** or **n** (quit or no). It is also possible to use one of the system editors to correct mistakes made while entering data. After the "Continue?" prompt, type any of the following: **edit**, **ex**, **vi**, or **ed** — you will be placed inside the corresponding editor, and returned to **addbib** afterwards, from where you can either quit or add more data.

If the prompts normally supplied by **addbib** are not enough, are in the wrong order, or are too numerous, you can redefine the skeleton by constructing a promptfile. Create some file, to be named after the `-p` command line option. Place the prompts you want on the left side, followed by a single TAB (control-I), then the **refer** code that is to appear in the bibliographic database. **Addbib** will send the left side to the screen, and the right side, along with data entered, to the database.

### Printing the Bibliography

**Sortbib** is for sorting the bibliography by author (%A) and date (%D), or by data in other fields. It is quite useful for producing bibliographies and annotated bibliographies, which are seldom entered in strict alphabetical order. It takes as arguments the names of up to 16 bibliography files, and sends the sorted records to standard output (the terminal screen), which may be redirected through a pipe or into a file.

The `-sKEYS` flag to **sortbib** will sort by fields whose key-letters are in the *KEYS* string, rather than merely by author and date. Key-letters in *KEYS* may be followed by a '+' to indicate that all such fields are to be used. The default is to sort by senior author and date (printing the senior author last name first), but `-sA+D` will sort by all authors and then date, and `-sATD` will sort on senior author, then title, and then date.

**Roffbib** is for running off the (probably sorted) bibliography. It can handle annotated bibliographies — annotations are entered in the %X (abstract) field. **Roffbib** is a shell script that calls **refer** `-B` and **nroff** `-mbib`. It uses the macro definitions that reside in `/usr/lib/tmac/tmac.bib`, which you can redefine if you know **nroff** and **troff**. Note that **refer** will print the %H and %O commentaries, but will ignore abstracts in the %X field; **roffbib** will print both fields, unless annotations are suppressed with the `-x` option.

The following command sequence will lineprint the entire bibliography, organized alphabetically by author and date:

```
% sortbib database | roffbib | lpr
```

This is a good way to proofread the bibliography, or to produce a stand-alone bibliography at the end of a paper. Incidentally, **roffbib** accepts all flags used with **nroff**. For example:

```
% sortbib database | roffbib -Tdtc -s1
```

will make accent marks work on a DTC daisy-wheel printer, and stop at the bottom of every page for changing paper. The `-n` and `-o` flags may also be quite useful, to start page numbering at a selected point, or to produce only specific pages.

**Roffbib** understands four command-line number registers, which are something like the two-letter number registers in `-ms`. The `-rN1` argument will number references beginning at one (1); use another number to start somewhere besides one. The `-rV2` flag will double-space the entire bibliography, while `-rV1` will double-space the references, but single-space the annotation paragraphs. Finally, specifying `-rL6i` changes the line length from 6.5 inches to 6 inches, and saying `-rO1i` sets the page offset to one inch, instead of zero. (That's a capital O after `-r`, not a zero.)

### Citing Papers with Refer

The **refer** program normally copies input to output, except when it encounters an item of the form:

```
.[  
  partial citation  
.]
```

The partial citation may be just an author's name and a date, or perhaps a title and a keyword, or maybe just a document number. **Refer** looks up the citation in the bibliographic database, and transforms it into a full, properly formatted reference. If the partial citation does not correctly identify a single work (either finding nothing, or more than one reference), a diagnostic message is given. If nothing is found, it

will say “No such paper.” If more than one reference is found, it will say “Too many hits.” Other diagnostic messages can be quite cryptic; if you are in doubt, use **checknr** to verify that all your `.[`s have matching `.]`s.

When everything goes well, the reference will be brought in from the database, numbered, and placed at the bottom of the page. This citation,<sup>1</sup> for example, was produced by:

```
This citation,
.[
lesk inverted indexes
.]
for example, was produced by
```

The `.[` and `.]` markers, in essence, replace the `.FS` and `.FE` of the `-ms` macros, and also provide a numbering mechanism. Footnote numbers will be bracketed on the the lineprinter, but superscripted on daisy-wheel terminals and in **troff**. In the reference itself, articles will be quoted, and books and journals will be underlined in **nroff**, and italicized in **troff**.

Sometimes you need to cite a specific page number along with more general bibliographic material. You may have, for instance, a single document that you refer to several times, each time giving a different page citation. This is how you could get “p. 10” in the reference:

```
.[
kies document formatting
%P 10
.]
```

The first line, a partial citation, will find the reference in your bibliography. The second line will insert the page number into the final citation. Ranges of pages may be specified as “%P 56-78”.

When the time comes to run off a paper, you will need to have two files: the bibliographic database, and the paper to format. Use a command line something like one of these:

```
% refer -p database paper | nroff -ms
% refer -p database paper | tbl | nroff -ms
% refer -p database paper | tbl | neqn | nroff -ms
```

If other preprocessors are used, **refer** should precede **tbl**, which must in turn precede **eqn** or **neqn**. The `-p` option specifies a “private” database, which most bibliographies are.

### Refer's Command-line Options

Many people like to place references at the end of a chapter, rather than at the bottom of the page. The `-e` option will accumulate references until a macro sequence of the form

```
.[
$LIST$
.]
```

is encountered (or until the end of file). **Refer** will then write out all references collected up to that point, collapsing identical references. Warning: there is a limit (currently 200) on the number of references that can be accumulated at one time.

It is also possible to sort references that appear at the end of text. The `-sKEYS` flag will sort references by fields whose key-letters are in the *KEYS* string, and permute reference numbers in the text accordingly. It is unnecessary to use `-e` with it, since `-s` implies `-e`. Key-letters in *KEYS* may be followed by a ‘+’ to indicate that all such fields are to be used. The default is to sort by senior author and date, but `-sA+D` will sort on all authors and then date, and `-sA+T` will sort by authors and then title.

---

<sup>1</sup> Mike E. Lesk, “Some Applications of Inverted Indexes on the Unix System,” *Unix Programmer's Manual*, 2a, Bell Laboratories, Murray Hill, NJ (1978).

**Refer** can also make citations in what is known as the Social or Natural Sciences format. Instead of numbering references, the `-l` (letter ell) flag makes labels from the senior author's last name and the year of publication. For example, a reference to the paper on Inverted Indexes cited above might appear as [Lesk1978a]. It is possible to control the number of characters in the last name, and the number of digits in the date. For instance, the command line argument `-l6,2` might produce a reference such as [Kernig78c].

Some bibliography standards shun both footnote numbers and labels composed of author and date, requiring some keyword to identify the reference. The `-k` flag indicates that, instead of numbering references, key labels specified on the `%L` line should be used to mark references.

The `-n` flag means to not search the default reference file, located in `/usr/dict/papers/Rv7man`. Using this flag may make **refer** marginally faster. The `-an` flag will reverse the first  $n$  author names, printing Jones, J. A. instead of J. A. Jones. Often `-a1` is enough; this will reverse the names of only the senior author. In some versions of **refer** there is also the `-f` flag to set the footnote number to some pre-determined value; for example, `-f23` would start numbering with footnote 23.

### Making an Index

Once your database is large and relatively stable, it is a good idea to make an index to it, so that references can be found quickly and efficiently. The **indxbib** program makes an inverted index to the bibliographic database (this program is called **pubindex** in the Bell Labs manual). An inverted index could be compared to the thumb cuts of a dictionary — instead of going all the way through your bibliography, programs can move to the exact location where a citation is found.

**Indxbib** itself takes a while to run, and you will need sufficient disk space to store the indexes. But once it has been run, access time will improve dramatically. Furthermore, large databases of several million characters can be indexed with no problem. The program is exceedingly simple to use:

```
% indxbib database
```

Be aware that changing your database will require that you run **indxbib** over again. If you don't, you may fail to find a reference that really is in the database.

Once you have built an inverted index, you can use **lookbib** to find references in the database. **Lookbib** cannot be used until you have run **indxbib**. When editing a paper, **lookbib** is very useful to make sure that a citation can be found as specified. It takes one argument, the name of the bibliography, and then reads partial citations from the terminal, returning references that match, or nothing if none match. Its prompt is the greater-than sign.

```
% lookbib database
> lesk inverted indexes
%A  Mike E. Lesk
%T  Some Applications of Inverted Indexes on the Unix System
%J  Unix Programmer's Manual
%I  Bell Laboratories
%C  Murray Hill, NJ
%D  1978
%V  2a
%X  Difficult to read paper that dwells on indexing strategies,
    giving little practical advice about using \fBrefer\fp.
>
```

If more than one reference comes back, you will have to give a more precise citation for **refer**. Experiment until you find something that works; remember that it is harmless to overspecify. To get out of the **lookbib** program, type a control-D alone on a line; **lookbib** then exits with an "EOT" message.

**Lookbib** can also be used to extract groups of related citations. For example, to find all the papers by Brian Kernighan found in the system database, and send the output to a file, type:

```
% lookbib /usr/dict/papers/Ind > kern.refs
> kernighan
> EOT
% cat kern.refs
```

Your file, “kern.refs”, will be full of references. A similar procedure can be used to pull out all papers of some date, all papers from a given journal, all papers containing a certain group of keywords, etc.

### Refer Bugs and Some Solutions

The **refer** program will mess up if there are blanks at the end of lines, especially the %A author line. **Addbib** carefully removes trailing blanks, but they may creep in again during editing. Use an editor command — `g/ *$/s///` — to remove trailing blanks from your bibliography.

Having bibliographic fields passed through as string definitions implies that interpolated strings (such as accent marks) must have two backslashes, so they can pass through copy mode intact. For instance, the word “téléphone” would have to be represented:

```
te\\*`le\\*`phone
```

in order to come out correctly. In the %X field, by contrast, you will have to use single backslashes instead. This is because the %X field is not passed through as a string, but as the body of a paragraph macro.

Another problem arises from authors with foreign names. When a name like “Valéry Giscard d’Estaing” is turned around by the `-a` option of **refer**, it will appear as “d’Estaing, Valéry Giscard,” rather than as “Giscard d’Estaing, Valéry.” To prevent this, enter names as follows:

```
%A Vale\\*`ry Giscard\0d’Estaing
%A Alexander Csoma\0de\0Ko\\*ro\\*::s
```

(The second is the name of a famous Hungarian linguist.) The backslash-zero is an **nroff/troff** request meaning to insert a digit-width space. It will protect against faulty name reversal, and also against mis-sorting.

Footnote numbers are placed at the end of the line before the `.[` macro. This line should be a line of text, not a macro. As an example, if the line before the `.[` is a `.R` macro, then the `.R` will eat the footnote number. (The `.R` is an `-ms` request meaning change to Roman font.) In cases where the font needs changing, it is necessary to do the following:

```
\flet al.\fR
.[
awk aho kernighan weinberger
.]
```

Now the reference will be to Aho *et al.*<sup>2</sup> The `\fi` changes to italics, and the `\fR` changes back to Roman font. Both these requests are **nroff/troff** requests, not part of `-ms`. If and when a footnote number is added after this sequence, it will indeed appear in the output.

### Internal Details of Refer

You have already read everything you need to know in order to use the **refer** bibliography system. The remaining sections are provided only for extra information, and in case you need to change the way **refer** works.

The output of **refer** is a stream of string definitions, one for each field in a reference. To create string names, percent signs are simply changed to an open bracket, and an `[F` string is added, containing the footnote number. The %X, %Y and %Z fields are ignored; however, the **annobib** program changes

---

<sup>2</sup> Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger, “Awk — A Pattern Scanning and Processing Language,” *Unix Programmer’s Manual*, 2a, Bell Laboratories, Murray Hill, NJ (1978).

the %X to an .AP (annotation paragraph) macro. The citation used above yields this intermediate output:

```
.ds [F 1
.]-
.ds [A Mike E. Lesk
.ds [T Some Applications of Inverted Indexes on the Unix System
.ds [J Unix Programmer's Manual
.ds [I Bell Laboratories
.ds [C Murray Hill, NJ
.ds [D 1978
.ds [V 2a
.nr [T 0
.nr [A 0
.nr [O 0
.][ 1 journal-article
```

These string definitions are sent to **nroff**, which can use the `-ms` macros defined in `/usr/lib/mx/tmac.xref` to take care of formatting things properly. The initializing macro `.]-` precedes the string definitions, and the labeled macro `.][` follows. These are changed from the input `.[` and `.]` so that running a file twice through **refer** is harmless.

The `.][` macro, used to print the reference, is given a type-number argument, which is a numeric label indicating the type of reference involved. Here is a list of the various kinds of references:

Field	Value	Kind of Reference
%J	1	Journal Article
%B	3	Article in Book
%R %G	4	Report, Government Report
%I	2	Book
%M	5	Bell Labs Memorandum (undefined)
none	0	Other

The order listed above is indicative of the precedence of the various fields. In other words, a reference that has both the %J and %B fields will be classified as a journal article. If none of the fields listed is present, then the reference will be classified as “other.”

The footnote number is flagged in the text with the following sequence, where *number* is the footnote number:

```
\*([.number\*(.)
```

The `\*([.` and `\*(.)` stand for bracketing or superscripting. In **nroff** with low-resolution devices such as the `lpr` and a `crt`, footnote numbers will be bracketed. In **troff**, or on daisy-wheel printers, footnote numbers will be superscripted. Punctuation normally comes before the reference number; this can be changed by using the `-P` (postpunctuation) option of **refer**.

In some cases, it is necessary to override certain fields in a reference. For instance, each time a work is cited, you may want to specify different page numbers, and you may want to change certain fields. This citation will find the Lesk reference, but will add specific page numbers to the output, even though no page numbers appeared in the original reference.



```
.[  
lesk inverted indexes  
%P 7-13  
%I Computing Services  
%O UNIX 12.2.2.  
.]
```

The %I line will also override any previous publisher information, and the %O line will append some commentary. The **refer** program simply adds the new %P, %I, and %O strings to the output, and later strings definitions cancel earlier ones.

It is also possible to insert an entire citation that does not appear in the bibliographic database. This reference, for example, could be added as follows:

```
.[  
%A Brian Kernighan  
%T A Troff Tutorial  
%I Bell Laboratories  
%D 1978  
.]
```

This will cause **refer** to interpret the fields exactly as given, without searching the bibliographic database. This practice is not recommended, however, because it's better to add new references to the database, so they can be used again later.

If you want to change the way footnote numbers are printed, signals can be given on the .[ and .] lines. For example, to say "See reference (2)," the citation should appear as:

```
See reference  
.[(  
partial citation  
.)],
```

Note that blanks are significant on these signal lines. If a permanent change in the footnote format is desired, it's best to redefine the [. and .] strings.

### Changing the Refer Macros

This section is provided for those who wish to rewrite or modify the **refer** macros. This is necessary in order to make output correspond to specific journal requirements, or departmental standards. First there is an explanation of how new macros can be substituted for the old ones. Then several alterations are given as examples. Finally, there is an annotated copy of the **refer** macros used by **roffbib**.

The **refer** macros for **nroff/troff** supplied by the `-ms` macro package reside in `/usr/lib/mx/tmac.xref`; they are reference macros, for producing footnotes or endnotes. The **refer** macros used by **roffbib**, on the other hand, reside in `/usr/lib/tmac/tmac.bib`; they are for producing a stand-alone bibliography.

To change the macros used by **roffbib**, you will need to get your own version of this shell script into the directory where you are working. These two commands will get you a copy of **roffbib** and the macros it uses: †

```
% cp /usr/lib/tmac/tmac.bib bibmac
```

You can proceed to change `bibmac` as much as you like. Then when you use **roffbib**, you should specify your own version of the macros, which will be substituted for the normal ones

```
% roffbib -m bibmac filename
```

where *filename* is the name of your bibliography file. Make sure there's a space between `-m` and **bibmac**.

If you want to modify the **refer** macros for use with **nroff** and the `-ms` macros, you will need to get a copy of “`tmac.xref`”:

```
% cp /usr/lib/ms/s.ref refmac
```

These macros are much like “`bibmac`”, except they have `.FS` and `.FE` requests, to be used in conjunction with the `-ms` macros, rather than independently defined `.XP` and `.AP` requests. Now you can put this line at the top of the paper to be formatted:

```
.so refmac
```

Your new **refer** macros will override the definitions previously read in by the `-ms` package. This method works only if “`refmac`” is in the working directory.

Suppose you didn’t like the way dates are printed, and wanted them to be parenthesized, with no comma before. There are five identical lines you will have to change. The first line below is the old way, while the second is the new way:

```
.if !"\*(D"" , \*(D\c
.if !"\*(D"" \& (\*(D)\c
```

In the first line, there is a comma and a space, but no parentheses. The “`\c`” at the end of each line indicates to **nroff** that it should continue, leaving no extra space in the output. The “`\&`” in the second line is the do-nothing character; when followed by a space, a space is sent to the output.

If you need to format a reference in the style favored by the Modern Language Association or Chicago University Press, in the form (city: publisher, date), then you will have to change the middle of the book macro [2] as follows:

```
\& (\c
.if !"\*(C"" \*(C:
\*(I\c
.if !"\*(D"" , \*(D\c
)\c
```

This would print (Berkeley: Computing Services, 1982) if all three strings were present. The first line prints a space and a parenthesis; the second prints the city (and a colon) if present; the third always prints the publisher (books must have a publisher, or else they’re classified as other); the fourth line prints a comma and the date if present; and the fifth line closes the parentheses. You would need to make similar changes to the other macros as well.

## Acknowledgements

Mike Lesk of Bell Laboratories wrote the original **refer** software, including the indexing programs. Al Stangenberger of the Forestry Department wrote the first version of **addbib**, then called **bibin**. Greg Shenaut of the Linguistics Department wrote the original versions of **sortbib** and **roffbib**. All these contributions are greatly appreciated.

## Table of Contents

Introduction . . . . .	1
Data Entry with Addbib . . . . .	2
Printing the Bibliography . . . . .	3
Citing Papers with Refer . . . . .	3
Refer's Command-line Options . . . . .	4
Making an Index . . . . .	5
Refer Bugs and Some Solutions . . . . .	6
Internal Details of Refer . . . . .	6
Changing the Refer Macros . . . . .	8
Acknowledgements . . . . .	9
Commented Refer Macros . . . . .	10