The UNIX Text Processing System
Kaare Christian
John Wiley & Son, 1987.

CHAPTER 10

References and the refer Preprocessor

Although references form but a small part of business correspondence, they are an important part of most scientific and technical writing. There are several difficulties with typesetting references that are addressed by the body of UNIX programs loosely known as `refer`. One problem is tedium—references are often used in multiple papers and it is tedious to rekey a reference each time it is used. Another difficulty is the formatting that is required by certain journals for references. Yet another problem is finding the correct citation; perhaps you know the name of the author but can't find an accurate title, date, etc.

The refer system is a much more comprehensive tool than the `tbl` and `eqn` preprocessors discussed in Chapters 8 and 9. The `refer` program is a `troff` preprocessor, and there are additional programs for creating a data base of references, sorting the data base, printing the data base, and for independent queries of the data base. Throughout this chapter the term refer *software* will cover the entire group of programs; I will try to say `refer` *program* when I am talking about the `troff` preprocessor named `refer`.

The refer software is less widely available than the `tbl` or `eqn` preprocessors. The refer software first appeared in Version 7 UNIX, and it is also a standard part of Berkeley UNIX. However, the refer software is not a standard part of UNIX System V, and it is not a standard part of the optional Documenter's Workbench software for System V. Even worse, the `refer` program is supported only by `-ms` and `-me` macros, which are not part of System V.

The original description of `refer`'s technology appeared in *Some Applications of Inverted Indexes on the UNIX System*, by Mike Lesk. For users more interested in applications than software technology, there is *Refer—A Bibliography System*, by Bill Tuthill.

10.1 THE REFER SOFTWARE SYSTEM

Even if you infrequently format a table or equation, you should learn about `tbl` and `eqn`, because the alternative is almost unthinkable. However, the refer software is not necessary for occasional references, because small numbers of references can easily be managed manually.

The advantage of the refer software is productivity. You can cite a multiauthor paper that has an excessively long-winded title with just a few keystrokes. Students can sound as learned as their elders simply by borrowing and citing a seasoned data base. An extensive reference data base is even useful for on-line literature searches.

The `refer` program is a `troff` preprocessor. It recognizes an informally phrased reference citation in the input text, examines the reference data bases to find the exact citation, and places the text of the citation into the text in a style amenable to processing by the `-ms` macro package. Each user can have multiple reference data bases, and there is also a default systemwide reference data base. The reference data bases are maintained separately from documents containing references by the `addbib` and `indxbib` programs. Reference data bases can be searched by the `lookbib` program, and they can be printed using `sortbib` and `roffbib`.

10.2 ADDBIB

A reference data base is simply a text file containing a set of references. The format of the text in a reference data base will be described below. There are many ways to organize reference data bases. If you write papers in several disciplines, you may want to maintain separate reference data bases for each subject. In a departmental setting, it may be preferable for each member to maintain his or her own data base, or the members can pool their references and maintain a departmentwide reference data base. `refer` can search several data bases during one pass through a document, so you can partition your references into separate data bases as necessary.

In a reference data base, each record (citation) is separated from other citations by a blank line. Within a citation, each element (e.g., author, title) of the citation is on a separate line, and each such line starts with a `%` followed by the citation key letter followed by a blank. Each element in a citation can span multiple lines if necessary. Here is a citation for Charles Darwin's last book, a treatise on worms.

```
%A C. Darwin
%T The Formation of Vegetable Mould, Through
the Action of Worms
%I John Murray
%C London
%D 1881
%K mold
```

In this citation the author field, `%A`, has the data *C. Darwin*. If there had been a coauthor, then there would have been several `%A` fields, the first being the name of the principal author. Notice that Darwin's long title is placed on two lines. The `%I` field is the publisher, and the `%C` and `%D`, are the city and date of publication. Other fields often used are `%J` for the name of a journal containing a paper, `%B` for the name of a book containing a paper, `%V` and `%N` for the volume and number within a volume, `%E` for the name of a book's editor, and `%P` for the pages of interest. In some styles of papers, the footnotes sometimes contain preamble commentary (printed before the reference) or concluding commentary, specified using `%H`

Table 10.1. `refer`'s Field Identifiers.

`%H`	Header Commentary (printed before reference)	`%O`	Other Commentary (printed after reference)
`%A`	Author's Name	`%Q`	Corporate or Foreign Author
`%T`	Title	`%S`	Series Title
`%J`	Journal Name	`%B`	Book Name
`%R`	Report, Paper, or Thesis (unpublished material)	`%E`	Editor (of book containing article)
`%V`	Volume Number	`%N`	Number within Volume
`%P`	Page Number	`%D`	Date of Publication
`%I`	Publisher	`%C`	City of Publisher
`%K`	Additional Keywords	`%X`	Abstract
`%L`	Label (for alternate refer style)		

The `%T` keyword is used for the title of an article or a book. If it is the title of an article, the `%J` or the `%B` keyword should also be used to identify the source. Don't use both `%J` and `%B` in one citation, and don't use `%B` except to name a book in which the given article appears. Authors identified with the `%A` keyword may have the last word of the name printed first (when the `-a` option of `refer` is invoked); use the `%Q` keyword for authors whose names can't be reversed sensibly.

and %O, respectively. An abstract of the paper may appear following the %X indicator, and relevant keywords can be specified following the %K. The keyword *mold* was specified in this case, because people would more likely refer to this book using the American spelling than the British spelling (*mould*), which Darwin used in his original title. Unless you write your own macros to support `refer`, you needn't specify keywords that appear elsewhere in the citation. Table 10.1 contains a full list of the `refer` system key letters.

One method of maintaining reference data base files is using `vi` or some other text editor. The data base is a text file, so all you need to do is to stick to the given format. If you choose to maintain these files by hand, there are two things you must do very carefully: you must always leave a blank line between citations, and you must never allow any white space (blanks or tabs) at the end of a line.

`addbib` is a menu program that helps you to enter references into a reference data base. `addbib` is not as flexible as a text editor, but it makes sure that the format of the reference file is maintained. After new entries have been placed into a reference data base file using `addbib`, you can fix minor mistakes using a text editor.

`addbib` prints a series of prompts for the most important citation entries. At each prompt you can enter the requested data, or just hit <CR> to skip to the next field. Entering a hyphen will back up one field, which is useful for repeating the author field. You can continue a long field onto several lines by typing a backslash as the last character on a line. Here is a typical dialogue with `addbib`, showing how the citation listed above was entered.

```
$ addbib darwin
Instructions? n

   Author:     C. Darwin
    Title:     The Formation of Vegetable Mould, Through\
>the Action of Worms
  Journal:
   Volume:
    Pages:
Publisher:     John Murray
     City:     London
     Date:     1881
    Other:
 Keywords:     mold
 Abstract: (ctrl-d to end)
^D
Continue? n
$ _
```

Other possible answers to the "Continue?" prompt are `y` or *carriage return* to continue entering references, and `ed` or `vi` to call up the `ed` or `vi` text editors to patch up the reference file.

If you are creating a bibliography data base for `troff` use, you will probably want to place accent marks in the names of foreign authors and publications. When the `refer` program processes the data base, each field, except for the `X` (abstract) field, is stored in a `troff` string. This means that you should escape each `troff` embedded command with an extra leading backslash. The `X` field is treated as a paragraph, so you don't need to treat embedded commands specially inside the `X` field.

10.3 SORTBIB AND ROFFBIB

For the purpose of citing references in papers, the order of references in your reference data base file doesn't matter. But if you want a hard-copy listing of the entire file, it is important for the references to be ordered appropriately. The `sortbib` program sorts a reference data base, producing the sorted version on the standard output. `roffbib` will print a reference data base, either one sorted by `sortbib` or one in its natural order. Both `sortbib` and `roffbib` are Berkeley additions to the refer software system. They are found on Berkeley UNIX systems but won't necessarily be found on a Version 7 UNIX system.

By default, `sortbib` sorts a reference data base primarily by author's last name and secondarily by date. Although `sortbib`'s output may be collected in a file, it is often sent to `roffbib` for printing. `roffbib` has a host of printing options, perhaps the two most important of which are *-Tterm* to tell `nroff` which terminal you are using, and `-Q` to send the output to the typesetter or laser printer.

Typical usage is to pipe the output of `sortbib` to `roffbib`:

```
$ sortbib darwin | roffbib -Q
$ _
```

`sortbib`'s input must be a file or files; it cannot read data from the standard input.

Here is my 'darwin' reference data base, sorted by the default `sortbib` strategy:

Darwin, C., *The Structure and Distribution of Coral Reefs,* Smith, Elder, London, 1842.

Darwin, C., *On the Origin of Species,* John Murray, London, 1859.

Darwin, C., *The Various Contrivances by Which Orchids Are Fertilized by Insects,* John Murray, London, 1862.

Darwin, C., *Variation of Animals and Plants under Domestication,* John Murray, London, 1868.

Darwin, C., *Different Forms of Flowers on Plants of the Same Species,* John Murray, London, 1877.

Darwin, C., *The Formation of Vegetable Mould, Through the Action of Worms,* John Murray, London, 1881.

Notice that the author part of the citations shown above has been reversed by `roffbib`.

The `-s`*keys* option of `sortbib` can tell `sortbib` to use a different set of primary and secondary sort keys, with up to four keys total. The letters following the `-s` are the sort fields, in order. For example, the option `-sDTA` would make `sortbib` sort primarily by date and then secondarily by title and author. The option `-sID` would sort primarily by publisher and secondarily by date. Here is the 'darwin' data base sorted by title using the `-sT sortbib` option:

Darwin, C., *Different Forms of Flowers on Plants of the Same Species,* John Murray, London, 1877.

Darwin, C., *The Formation of Vegetable Mould, Through the Action of Worms,* John Murray, London, 1881.

Darwin, C., *On the Origin of Species,* John Murray, London, 1859.

Darwin, C., *The Structure and Distribution of Coral Reefs,* Smith, Elder, London, 1842.

Darwin, C., *Variation of Animals and Plants under Domestication,* John Murray, London, 1868.

Darwin, C., *The Various Contrivances by Which Orchids Are Fertilized by Insects,* John Murray, London, 1862.

Notice that `sortbib` does its best to ignore articles (such as *the*) at the beginning of a title while sorting.

10.4 INDXBIB AND LOOKBIB

Before a reference data base can be used efficiently, it must be *indexed*. Indexing speeds access to the data base, and it also makes it possible to search multiple data bases by specifying a single data base. The `indxbib` program can index one or more data bases into a single set of index files. This allows you to maintain your references in several separate data bases, but then search them as if they were one large data base. Index files created by `indxbib` have the same base name as the data base with suffixes `.ia`, etc.

```
$ ls -l darwin*
-rw-r--r--  1 kc       681 Feb 13 22:57 darwin
$ indxbib darwin
$ ls -l darwin*
-rw-r--r--  1 kc       681 Feb 13 22:57 darwin
-rw-r--r--  1 kc      2056 Feb 16 17:37 darwin.ia
-rw-r--r--  1 kc       352 Feb 16 17:37 darwin.ib
-rw-r--r--  1 kc        87 Feb 16 17:37 darwin.ic
$ _
```

Each time a reference data base is updated using `addbib` (or a text editor), it should be reindexed. If you forget to reindex a data base after an update, then the `refer` program or `lookbib` may fail to find references that are in the data base.

`lookbib` is an interactive program for querying a reference data base. For example, you can ask `lookbib` to find all papers by a given author in a given year, or all papers mentioning a given keyword, or all papers from a given publisher. You can cite a group of papers by simply mentioning the words that should be matched. For example, you can get all papers published by John Murray by mentioning the word Murray. `addbib` isn't picky about turning up too many references; for example, the keyword *Murray* might turn up a paper by the author Bill Murray in addition to listing books published by the publisher John Murray.

Most reference citations consist of a handful of words to narrow the focus. The citation will usually list the author and the subject or the author and the date. While using `lookbib`, it's acceptable (and often desirable) to turn up multiple references for a given query. However, one of the most common uses of `lookbib` is to determine a query (a set of keywords) that is sufficiently focused to turn up a specific reference, because those keywords can be used in a manuscript to specify a reference for inclusion by the `refer` program. Figure 10.1 shows a sample `lookbib` dialogue.

10.5 THE REFER PREPROCESSOR

Although large reference data bases are useful for on-line literature searches, for most people the motivation for creating a reference data base is including citations in their papers. That's what the `refer` program does. The `refer` program is a `troff` preprocessor. It takes a text file containing text and reference citations, and it fills in the reference citations by looking them up in the data base. Once the reference citation has been filled in by `refer`, the file is passed to `troff`, where it becomes the responsibility of the `troff` macro package to format the references correctly. `refer` merely includes the necessary information in the document; it is the macro package that actually controls the format of the reference.

```
$ lookbib darwin
Instructions? n
> plants
%A C. Darwin
%T Variation of Animals and Plants under Domestication
%I John Murray
%C London
%D 1868
%K domestication

%A C. Darwin
%T Different Forms of Flowers on Plants of the Same Species
%I John Murray
%C London
%D 1877
%K flowers

> plants flowers
%A C. Darwin
%T Different Forms of Flowers on Plants of the Same Species
%I John Murray
%C London
%D 1877
%K flowers

> ^D
$ _
```

Figure 10.1. A `lookbib` dialogue. Notice that `lookbib`'s prompt is a >. The *plants* query pulls out two references from the data base. Adding the word *flowers* to the query focuses the query to a single book by Darwin.

`refer` is used similarly to any other `troff` preprocessor. When `refer` is used with any of the other preprocessors, it should be first in the pipeline. The most important argument for `refer` is `-p`, which is followed by the name of the reference data base file. Other file name arguments are presumed to be the document file(s).

```
$ refer -p darwin galap1.t | tbl | troff -ms
$ _
```

The command shown above uses the 'darwin' data base to resolve the references in the 'galap1.t' document. In this example the document file is presumed to contain tabular data, so the output of `refer` is sent to the `tbl` preprocessor and then to the `troff` formatter for printing, using the manuscript macros.

In a document a reference is cited by placing keywords inside the `.[, .]` braces. For example, the following citation will pull the full citation of Darwin's most famous work:

```
.[
darwin origin species
.]
```

Of course, this citation is overly precise (overly cautious) for our trivially small 'darwin' data base. However, overspecification is usually a good idea for large data bases, because it protects you from future entries that are similar to an existing one.

refer expects that the keywords will produce a unique citation; it is an error for the keywords to lead to zero references or to more than one reference. By default, the reference will be formatted as a footnote on the page where it appears in the document. This style is shown in Figure 10.2.

Several reference labeling systems are available. The default strategy is to number the citations, starting at 1. The starting point can be controlled by the **-f***n* command line argument, where *n* is the starting number. In some manuscript styles, it is best to include the labels in the data base, usually in the %L field. The **-k** command line argument will use the contents of the %L field as a label. (An arbitrary field can be used as the label using the **-k***x* command line argument, where *x* is the key letter of the label field.)

Another labeling style uses the senior author's last name and date of publication as the label. **refer** will produce this alternate style when the **-l***m,n* command line argument is present. (l is lowercase ell.) The *m* specifies the number of characters to take from the author, and the *n* specifies the number of characters to take of the date. If *m* (or *n*) is omitted, the full name (or date) is used. For example, the command line argument **-l4.2** would produce the label **Darw59a** for *The Origin of Species*.

Another area of reference control offered by **refer** is the ordering of the author's names. The default is to print the author names in their natural order—first name (or initials), then last name. However, the **-a***n* command line argument of **refer** can be used to reverse the names of the first *n* authors. If *n* is omitted, then all author names will be reversed.

refer can also produce *endnotes*. You must specify the **-e** command line argument, and then you must place the special **refer** citation

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

in your document at the place where the endnotes should be placed. When you are using the endnote style, the **-s***keys* command line argument can be used to control the ordering of the endnotes (using the same syntax as for the **sortbib** program). **refer**'s ability to produce endnotes is shown in Figure 10.3.

INPUT:

```
.LP
Although Darwin is known primarily for his voyage on the
Beagle, his discoveries on the Galapagos Islands, and for
his seminal book \f2The Origin of Species\fP,
.[
darwin origin species
.]
we should also remember the methods that he pioneered in
achieving his greatness.  Evolution in the gradeschool sense
is a theory that explains how modern species have formed. But
in a deeper sense the theory of evolution is a method, a
technique for examining and understanding nature.  Before
Darwin's work scientists didn't know how to form or test
hypotheses about the history of natural processes. Darwin
showed that small, easily observable changes are the stuff
from which epochal changes are made. This argument was made
with greatest force in Darwin's last book, the oft
misunderstood \f2The Formation of Vegetable Mould, Through
the Action of Worms\fP.
.[
darwin mold
.]
```

OUTPUT:

Although Darwin is known primarily for his voyage on the Beagle, his discoveries on the Galapagos Islands, and for his seminal book *The Origin of Species*,[1] we should also remember the methods that he pioneered in achieving his greatness. Evolution in the gradeschool sense is a theory that explains how modern species have formed. But in a deeper sense the theory of evolution is a method, a technique for examining and understanding nature. Before Darwin's work scientists didn't know how to form or test hypotheses about the history of natural processes. Darwin showed that small, easily observable changes are the stuff from which epochal changes are made. This argument was made with greatest force in Darwin's last book, the oft misunderstood *The Formation of Vegetable Mould, Through the Action of Worms*.[2]

1. C. Darwin, *On the Origin of Species,* John Murray, London (1859).
2. C. Darwin, *The Formation of Vegetable Mould, Through the Action of Worms,* John Murray, London (1881).

Figure 10.2. This example shows how two citations in the body of the text are printed as footnotes using the **refer** preprocessor with the **-ms** macros.

INPUT:

```
.LP
Although Darwin is known primarily for his voyage on the
Beagle, his discoveries on the Galapagos Islands, and for
his seminal book \f2The Origin of Species\fP,
.[
darwin origin species
.]
we should also remember the methods that he pioneered in
achieving his greatness.  Evolution in the gradeschool sense
is a theory that explains how modern species have formed. But
in a deeper sense the theory of evolution is a method, a
technique for examining and understanding nature.  Before
Darwin's work scientists didn't know how to form or test
hypotheses about the history of natural processes. Darwin
showed that small, easily observable changes are the stuff
from which epochal changes are made. This argument was made
with greatest force in Darwin's last book, the oft
misunderstood \f2The Formation of Vegetable Mould, Through
the Action of Worms\fP.
.[
darwin mold
.]
.[
$LIST$
.]
```

Figure 10.3(a). This example shows how two citations in the body of the text are printed as endnotes using the **-e** option of the **refer** preprocessor with the **-ms** macros.

Occasionally it is desirable to specify one of the data base fields within a **refer** style reference in a manuscript. For example, in one context you might want to mention a given page number in one reference to a paper and another page number when referring to that paper in some other context. Or you might want to use the %L (label) field differently in different manuscripts, etc. All of these needs are addressed by the ability to redefine a field within a citation simply by placing a bibliographic style line in the reference. For example, you could cite page 108 of Darwin's *Origin of Species* with the following citation.

```
.[
darwin origin species
%P 108
.]
```

OUTPUT:

Although Darwin is known primarily for his voyage on the Beagle, his discoveries on the Galapagos Islands, and for his seminal book *The Origin of Species*,[1] we should also remember the methods that he pioneered in achieving his greatness. Evolution in the gradeschool sense is a theory that explains how modern species have formed. But in a deeper sense the theory of evolution is a method, a technique for examining and understanding nature. Before Darwin's work scientists didn't know how to form or test hypotheses about the history of natural processes. Darwin showed that small, easily observable changes are the stuff from which epochal changes are made. This argument was made with greatest force in Darwin's last book, the oft misunderstood *The Formation of Vegetable Mould, Through the Action of Worms*.[2]

References

1. C. Darwin, *On the Origin of Species,* John Murray, London (1859).
2. C. Darwin, *The Formation of Vegetable Mould, Through the Action of Worms,* John Murray, London (1881).

Figure 10.3(b). Sample output from the input in Figure 10.3(a).