

# *Utroff Refer documentation*

*Enhancement of Heirloom troff refer tools which insert references*

*(Beta version : user interface may change)*

Refer readme . . . . .	2	Hunt manual . . . . .	16
Refer manual . . . . .	3	Changes of refer . . . . .	18
Refer input and output format	6	Original BSD License . . . . .	20
Sortbib manual . . . . .	10	Common Development and	
Inv manual . . . . .	12	DiStribution License Version 1.0	21
Mkey manual . . . . .	14	Tables . . . . .	27

---

# *Refer readme*

## ***Presentation***

*Utroff refer* is a modification of the *heirloom refer* tools which manages references for troff. The modification of this tools has two aims :

- Follow the iso-690 bibliography standart.
- Add better support of names in bibliography files (support multiple issuer, use real small capitals).
- Add some manual pages.

## ***Build and install***

To build and install, edit the file `config.mk` from the root directory, and run :

```
| make install
```

In some circumstances, you might need utroff tools themselves to build the manual pages. In that case, build and install binaries before manual pages like this :

```
| make installbin  
| make installman
```

You can uninstall files and clean the working directory :

```
| make uninstall  
| make clean
```

For a more complete description of the build process, look at the README in the root directory.

## ***License***

*Refer* and the other binaries are distributed under the CDDL 1.0 license<sup>1</sup>. Manual pages of *refer* and *sortbib* are distributed under the original BSD license<sup>2</sup>. The manual pages of *hunt*, *inv*, *mkey* and *referformat* have been made from *Some applications of inverted indexes on the UNIX system* by M. E. LESK, which was part of the *Unix User Supplementary Document*<sup>3</sup> distributed under the original BSD license<sup>4</sup>, as said on the *Berkely Software Documentation*<sup>5</sup> webpage.

## ***Help and bugs***

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

---

<sup>1</sup> See p. 21.

<sup>4</sup> See p. 20.

<sup>2</sup> See p. 20.

<sup>5</sup> <<http://netbsd.org/docs/bsd/lite2/>>

<sup>3</sup> <<http://netbsd.org/docs/bsd/lite2/usd.html>>

---

# *Refer manual*

## **Name**

**refer** – (utroff) insert iso-690 compliant references in document

## **Synopsis**

@BINDIR@/refer [-abcdeinPS] [-fn] [-kx] [-lm,n] [-p bib] [-skeys] [-Bl.m] [file ...]

## **Description**

*Refer* is an enhancement of the *Heirloom refer* preprocessor for *nroff* and *troff* that finds and formats references for footnotes or endnotes. It is also the base for a series of programs designed to index, search, sort, and print stand-alone bibliographies, or other data entered in the appropriate form.

This version adds to *Heirloom refer* the ability to sort following the *iso-690(7)* bibliography standard (-i option), a better capitalize option (-d), support editor names like authors names, and insertion of the universal "&" instead of the english "and" to join names.

Given an incomplete citation with sufficiently precise keywords, *refer* will search a bibliographic database for references containing these keywords anywhere in the title, author, journal, etc. The input file (or standard input) is copied to standard output, except for lines between .[ and .] delimiters, which are assumed to contain keywords, and are replaced by information from the bibliographic database. The user may also search different databases, override particular fields, or add new fields. The reference data, from whatever source, are assigned to a set of *troff* strings. Macro packages such as *ms* print the finished reference text from these strings. By default references are flagged by footnote numbers.

The following options are available :

- a *n* Reverse the first *n* author or editor names (Jones, J. A. instead of J. A. Jones). If *n* is omitted all author and editor names are reversed.
- b Bare mode : do not put any flags in text (neither numbers nor labels).
- c *keys* Capitalize (with CAPS SMALL CAPS) the whole field whose key-letters are in the first-name part of fields whose key-letters are in *keys*. The firstname part is the last word of the fields or the previous one if it ends with a comma (",").
- d *keys* Insert strings around the lastname, firstname and junior part of fields whose key-letters are in *keys*, for later formatting. The firstname part is the last word of the field or the previous one if it ends with a comma (","). Strings are \\*(+F and \\*(-F around firstname, \\*(+L and \\*(-L around lastname, \\*(+J and \\*(-J around junior.
- e Instead of leaving the references where encountered, accumulate them until a sequence of the form

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

is encountered, and then write out all references collected so far. Collapse references to same source.

- f *n* Set the footnote number to *n* instead of the default of 1 (one). With labels rather than
-

- 
- numbers, this flag is a no-op.
  - i Sort following the iso-690 standard order of fields : If **J** is defined, the order of fields is "**QATESVBJ**"; Else, if **B** is defined, the order of fields is "**QATESVB**"; Else, the order of fields is "**QAESVT**". The *u-ref* macro need this option to work properly.
  - k *x* Instead of numbering references, use labels as specified in a reference data line beginning %*x*; by default *x* is **L**.
  - l *m* , *n* Instead of numbering references, use labels made from the senior author's last name and the year of publication. Only the first *m* letters of the last name and the last *n* digits of the date are used. If either *m* or *n* is omitted the entire name or date respectively is used.
  - n Do not search the default file `/$REFLIB$/papers/Ind`. If there is a **REFER** environment variable, the specified file will be searched instead of the default file ; in this case the **-n** flag has no effect.
  - p *bib* Take the next argument *bib* as a file of references to be searched. The default file is searched last.
  - s *keys* Sort references by fields whose key-letters are in the *keys* string ; permute reference numbers in text accordingly. Implies **-e**. The key-letters in *keys* may be followed by a number to indicate how many such fields are used, with + taken as a very large number. The default is **AD** which sorts on the senior author and then date ; to sort, for example, on all authors and then title, use **-sA+T**.
  - B *l.m* Bibliography mode. Take a file composed of records separated by blank lines, and turn them into *troff* input. Label *l* will be turned into the macro *.m* with *l* defaulting to %**X** and *.m* defaulting to **.AP** (annotation paragraph).
  - P Place punctuation marks ., : ; ? ! after the reference signal, rather than before. (Periods and commas used to be done with strings.)
  - S Produce references in the Natural or Social Science format.

To use your own references, put them in the format described below. They can be searched more rapidly by running *indxbib*(1B) on them before using *refer*; failure to index results in a linear search. When *refer* is used with the *eqn*, *neqn* or *tbl* preprocessors *refer* should be first, to minimize the volume of data passed through pipes.

The *refer* preprocessor and associated programs expect input from a file of references composed of records separated by blank lines. A record is a set of lines (fields), each containing one kind of information. Fields start on a line beginning with a "%", followed by a key-letter, then a blank, and finally the contents of the field, and continue until the next line starting with "%". The output ordering and formatting of fields is controlled by the macros specified for *nroff/troff* (for footnotes and endnotes) or *roffbib* (for stand-alone bibliographies). For a list of the most common key-letters and their corresponding fields, see *addbib*(1B). An example of a *refer* entry is given below.

### Example

%A	M. E. Lesk
%T	Some Applications of Inverted Indexes on the UNIX System
%B	UNIX Programmer's Manual
%V	2b
%I	Bell Laboratories
%C	Murray Hill, NJ
%D	1978

### Files

@REFDIR@/papers directory of default publication lists @REFDIR@ directory of companion programs

### See also

---

---

referformat(7)<sup>1</sup>, sortbib(1)<sup>2</sup>, mkey(1)<sup>3</sup>, inv(1)<sup>4</sup>, hunt(1)<sup>5</sup>, and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

### **Notes**

Blank spaces at the end of lines in bibliography fields will cause the records to sort and reverse incorrectly. Sorting large numbers of references causes a core dump.

### **License**

*Refer* is distributed under the CDDL-1.0<sup>6</sup> license. This manual page is distributed under the original BSD<sup>7</sup> license.

### **Help and bugs**

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

### **Authors**

Written by Mike LESK, modified by Pierre-Jean FICHET.

---

<sup>1</sup> See p. 6.

<sup>2</sup> See p. 10.

<sup>3</sup> See p. 14.

<sup>4</sup> See p. 12.

<sup>5</sup> See p. 16.

<sup>6</sup> See p. 21.

<sup>7</sup> See p. 20.

## *Refer input and output format*

### *Refer file*

A reference file is a set of bibliographic references usable with *refer*. It can be indexed using the softwares *inv* and *mkey* for fast searching.

The strings defined by *refer* are taken directly from the files of references, which are in the following format. The references should be separated by blank lines. Each reference is a sequence of lines beginning with % and followed by a key-letter. The remainder of that line, and successive lines until the next line beginning with %, contain the information specified by the key-letter. In general, *refer* does not interpret the information, but merely presents it to the macro package for final formatting. A user with a separate macro package, for example, can add new key-letters or use the existing ones for other purposes without bothering *refer*.

The meaning of the key-letters given below, in particular, is that assigned by the *-ms* macros. Not all information, obviously, is used with each citation. For example, if a document is both an internal memorandum and a journal article, the macros ignore the memorandum version and cite only the journal article. Some kinds of information are not used at all in printing the reference; if a user does not like finding references by specifying title or author keywords, and prefers to add specific keywords to the citation, a field is available which is searched but not printed (K).

The key letters currently recognized by *refer* and *-ms*, with the kind of information implied, are :

<b>A</b>	Author's name
<b>B</b>	Title of book containing item
<b>C</b>	City of publication
<b>D</b>	Date
<b>E</b>	Editor of book containing item
<b>G</b>	Government (NTIS) ordering number
<b>I</b>	Issuer (publisher)
<b>J</b>	Journal name
<b>K</b>	Keys (for searching)
<b>L</b>	Label
<b>M</b>	Memorandum label
<b>N</b>	Issue number
<b>O</b>	Other information
<b>P</b>	Page(s) of article
<b>R</b>	Technical report reference
<b>T</b>	Title
<b>V</b>	Volume number
<b>X</b>	or
<b>Y</b>	or
<b>Z</b>	Information not used by refer

### *Refer output*

The output of *refer* is a stream of string definitions, one for each of the fields of each reference, as shown below.

| .]-

---

```
.ds [A authors' names ...
.ds [T title ...
.ds [J journal ...
...
.][ type-number
```

The *refer* program, in general, does not concern itself with the significance of the strings. The different fields are treated identically by *refer*, except that the X, Y and Z fields are ignored (see the -i option of *mkey*) in indexing and searching. All *refer* does is select the appropriate citation, based on the keys. The macro package must arrange the strings so as to produce an appropriately formatted citation. In this process, it uses the convention that the 'T' field is the title, the 'J' field the journal, and so forth. The *refer* program does arrange the citation to simplify the macro package's job, however. The special macro .]- precedes the string definitions and the special macro .][ follows. These are changed from the input .[ and .] so that running the same file through *refer* again is harmless.

The .]- macro can be used by the macro package to initialize. The .][ macro, which should be used to print the reference, is given an argument type-number to indicate the kind of reference, as follows :

- 1 Journal article
- 2 Book
- 3 Article within book
- 4 Technical report
- 5 Bell Labs technical memorandum
- 0 Other

The type is determined by the presence or absence of particular fields in the citation (a journal article must have a 'J' field, a book must have an 'T' field, and so forth). To a small extent, this violates the above rule that *refer* does not concern itself with the contents of the citation; however, the classification of the citation in *troff* macros would require a relatively expensive and obscure program. Any macro writer may, of course, preserve consistency by ignoring the argument to the .][ macro. The reference is flagged in the text with the sequence

```
| \*[.number\*(.]
```

where number is the footnote number. The strings [. and .] should be used by the macro package to format the reference flag in the text. These strings can be replaced for a particular footnote. The footnote number (or other signal) is available to the reference macro .][ as the string register [F. To simplify dealing with a text reference that occurs at the end of a sentence, *refer* treats a reference which follows a period in a special way. The period is removed, and the reference is preceded by a call for the string <. and followed by a call for the string >. For example, if a reference follows 'end.' it will appear as

```
| end\*(<\*[.number\*(.)\*(>.
```

where number is the footnote number. The macro package should turn either the string >. or <. into a period and delete the other one. This permits the output to have either the form 'end[31]'. or 'end.31' as the macro package wishes. Note that in one case the period precedes the number and in the other it follows the number.

In some cases users wish to suspend the searching, and merely use the reference macro formatting. That is, the user doesn't want to provide a search key between .[ and .] brackets, but merely the reference lines for the appropriate document. Alternatively, the user can wish to add a few fields to those in the reference as in the standard file, or override some fields. Altering or replacing fields, or supplying whole references, is easily done by inserting lines beginning with %; any such line is taken as direct input to the reference processor rather than keys to be searched. Thus

---

```
.[
key1 key2 key3 ...
%Q New format item
%R Override report name
.]
```

makes the indicates changes to the result of searching for the keys. All of the search keys must be given before the first % line. If no search keys are provided, an entire citation can be provided in-line in the text. For example, if the eqn paper citation were to be inserted in this way, rather than by searching for it in the data base, the input would read

```
.[
%A B. W. Kernighan
%A L. L. Cherry
%T A System for Typesetting Mathematics
%J Comm. ACM
%V 18
%N 3
%P 151-157
%D March 1975
.]
```

This would produce a citation of the same appearance as that resulting from the file search. As shown, fields are normally turned into troff strings. Sometimes users would rather have them defined as macros, so that other troff commands can be placed into the data. When this is necessary, simply double the control character % in the data. Thus the input

```
.[
%V 23
%%M
Bell Laboratories,
Murray Hill, N.J. 07974
.]
```

is processed by refer into

```
.ds [V 23
.de [M
Bell Laboratories,
Murray Hill, N.J. 07974
..
```

The information after %%M is defined as a macro to be invoked by .[M while the information after %V is turned into a string to be invoked by \\*(V. At present -ms expects all information as strings.

## License

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD<sup>1</sup> license.

## See also

refer(1)<sup>2</sup>, sortbib(1)<sup>3</sup>, mkey(1)<sup>4</sup>, inv(1)<sup>5</sup>, hunt(1)<sup>6</sup>, and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

<sup>1</sup> See p. 20.

<sup>4</sup> See p. 14.

<sup>2</sup> See p. 3.

<sup>5</sup> See p. 12.

<sup>3</sup> See p. 10.

<sup>6</sup> See p. 16.



***Help and bugs***

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

***Authors***

M. E. LESK. Modified by Pierre-Jean FICHET.

---

---

# *Sortbib manual*

## **Name**

**sortbib** – Sort refer bibliographic database following iso-690 standart

## **Synopsis**

*sortbib* [-i|-sKEYS] database ...

## **Description**

*Sortbib* sorts files of records containing *refer* key-letters by user-specified keys. Records may be separated by blank lines, or by .[ and .] delimiters, but the two styles may not be mixed together. This program reads through each *database* and pulls out key fields, which are sorted separately. The sorted key fields contain the file pointer, byte offset, and length of corresponding records. These records are delivered using disk seeks and reads, so *sortbib* may not be used in a pipeline to read standard input.

By default, *sortbib* alphabetize the first %A and %D fields, which contain the senior author and date.

The -i option is used to sort database following an algorithm which aims to respect the *iso-690* bibliography standart :

- If J is defined, the order of fields is : QA+TE+SVBJ.
- Else, if B is defined, the order of fields is : QA+TE+SVB.
- Else, the order of fields is : QA+E+SVT.

The -s option is used to specify new KEYS. For instance, -sATD will sort by author, title, and date, while -sA+D will sort by all authors, and date. Sort keys past the fourth are not meaningful. No more than 16 databases may be sorted together at one time. Records longer than 4096 characters will be truncated.

*Sortbib* sorts on last word on the %A and %E line, which are assumed to be the author's and editor's last name. A word in the final position, such as "jr." or "ed.", will be ignored if the name beforehand ends with a comma. Authors with two-word last names or unusual constructions can be sorted correctly by using the *nroff* convention "\0" in place of a blank. A %Q field is considered to be the same as %A, except sorting begins with the first, not the last, word. *Sortbib* sorts on the last word of the %D line, usually the year. If a sort-significant field is absent from a record, *sortbib* places that record before other records containing that field.

## **License**

*Sortbib* is distributed under the CDDL1.0 license<sup>1</sup>. This manual page is distributed under a four clause BSD<sup>2</sup> License.

## **Files**

---

<sup>1</sup> See p. 21.

<sup>2</sup> See p. 20.

@BINDIR@/sortbib

### ***See also***

refer(1)<sup>3</sup>, u-ref(7).

### ***Help and bugs***

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

### ***Authors***

Written by Greg SHENAUT & Bill TUTHILL. Modified by Pierre-Jean FICHET.

---

<sup>3</sup> See p. 3.

---

## *Inv manual*

### **Name**

*inv* – Make an inverted index from output of mkey.

### **Synopsis**

*inv* [-danpv] [-hn] [-i [u] name] outfile]

### **Description**

The *inv* program computes the hash codes and writes the inverted files. It reads the output of *mkey* and writes the set of files described earlier in this section. It expects one argument, which is used as the base name for the three (or four) files to be written. Assuming an argument of *Index* (the default) the entry file is named *Index.ia*, the posting file *Index.ib*, the tag file *Index.ic*, and the key file (if present) *index.id*.

The *inv* program recognize the following options :

- a      Append the new keys to a previous set of inverted files, making new files if there is no old set using the same base name.
- d      Write the optional key file. This is needed when you can not check for false drops by looking for the keys in the original inputs, i.e. when the key derivation procedure is complicated and the output keys are not words from the input files.
- hn     The hash table size is *n* (default 997); *n* should be prime. Making *n* bigger saves search time and spends disk space.
- i [u] name  
      Take input from file *name*, instead of the standard input; if **u** is present name is unlinked when the sort is started. Using this option permits the sort scratch space to overlap the disk space used for input keys.
- n      Make a completely new set of inverted files, ignoring previous files.
- p      Pipe into the sort program, rather than writing a temporary input file. This saves disk space and spends processor time.
- v      Verbose mode; print a summary of the number of keys which finished indexing.

About half the time used in *inv* is in the contained sort. Assuming the sort is roughly linear, however, a guess at the total timing for *inv* is 250 keys per second. The space used is usually of more importance : the entry file uses four bytes per possible hash (note the **-h** option), and the tag file around 15-20 bytes per item indexed. Roughly, the posting file contains one item for each key instance and one item for each possible hash code; the items are two bytes long if the tag file is less than 65536 bytes long, and the items are four bytes wide if the tag file is greater than 65536 bytes long. To minimize storage, the hash tables should be over-full; for most of the files indexed in this way, there is no other real choice, since the entry file must fit in memory.

### **Files**

@BINDIR@/invExecutable.

Assuming an argument of *Index* (the default) :

*Index.ia* Entry file.

*Index.ib* Posting file.

---

*Index.ic* Tag file.

*Index.id* Key file.

### ***License***

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD<sup>1</sup> license. The *inv* software is distributed under the CDDL-1.0<sup>2</sup> license.

### ***See also***

refer(1)<sup>3</sup>, referformat(7)<sup>4</sup>, mkey(1)<sup>5</sup>, hunt(1)<sup>6</sup>, and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

### ***Help and bugs***

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

M. E. LESK. Modified by Pierre-Jean FICHET

---

<sup>1</sup> See p. 20.

<sup>2</sup> See p. 21.

<sup>3</sup> See p. 3.

<sup>4</sup> See p. 6.

<sup>5</sup> See p. 14.

<sup>6</sup> See p. 16.

---

# Mkey manual

## Name

**Mkey** – Build an index of keys from files

## Synopsis

**mkey** [-sw] [-c *name*] [-f *name*] [-i *chars*] [-kn] [-ln] [-nm] *file* [*file...*]

## Description

**Mkey** reads its input from the file names given as arguments, and, if there is no arguments, it reads from the standart input. It assumes that blank lines in the input delimit separate items for each of which a different line of **keys** should be generated. The lines of keys are written on the standard output. **Keys** are any alphanumeric string in the input not among the most frequent words in English and not entirely numeric (except that all-numeric strings are acceptable if they are between 1900 and 1999). In the output, keys are translated to lower case, and truncated to six characters in length ; any associated punctuation is removed.

The following flags arguments are recognized by **mkey**.

- c *name* Name of file of common words. Default is @LIBDIR@/eign.
- f *name* Read a list of files from *name* and take each as an input argument.
- i *chars* Ignore each line wich begin with '%' followed by any character in *chars*.
- kn Use at most *n* keys per input item. Unless this option is used, the output of **mkey** is comparable in size to its input.
- ln Ignore items shorter than *n* letter longs. Default is 3.
- nm Ignore as a key any word in the first *m* words of the list of common English words. The default is 100.
- s Remove the labels (*file :start,length*) from the output ; just give the keys. Used when searching rather than indexing.
- w Each whole file is a separate item ; blank lines are irrelevant.

## Files

@BINDIR@/mkeyExecutable.

@LIBDIR@/eignList of frequent english words.

## License

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD<sup>1</sup> license. The software *mkey* is distributed under the CDDL-1.0<sup>2</sup> license.

## See also

refer(1)<sup>3</sup>, referformat(7)<sup>4</sup>, sortbib(1)<sup>5</sup>, inv(1)<sup>6</sup>, hunt(1)<sup>7</sup>, and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

---

<sup>1</sup> See p. 20.

<sup>2</sup> See p. 21.

### ***Help and bugs***

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

### ***Authors***

This manual page is made from the document *Some application of Inverted Indexes in the UNIX System* by M. E. LESK. Modified by Pierre-Jean FICHET

---

<sup>3</sup> See p. 3.

<sup>4</sup> See p. 6.

<sup>5</sup> See p. 10.

<sup>6</sup> See p. 12.

<sup>7</sup> See p. 16.

---

# *Hunt manual*

## **Name**

*hunt* – Retrieve items from an *mkey* index.

## **SYNOPSIS**

*hunt* [-agp] [-Cn] [-F[ynd]] [-i *string*] [-l n] [-o *string*] [-T[ynd]] [-t *string*] < *file*

## **Description**

The *hunt* program retrieves items from an index. It combines, as mentioned above, the two parts of that process : search and delivery. The reason why it is efficient to combine delivery and search is partly to avoid starting unnecessary processes, and partly because the delivery operation must be a part of the search operation in any case. Because of the hashing, the search part takes place in two stages : first items are retrieved which have the right hash codes associated with them, and then the actual items are inspected to determine false drops, i.e. to determine if anything with the right hash codes doesn't really have the right keys. Since the original item is retrieved to check on false drops, it is efficient to present it immediately, rather than only giving the tag as output and later retrieving the item again. If there were a separate key file, this argument would not apply, but separate key files are not common.

Input to *hunt* is taken from the standard input, one query per line. Each query should be in *mkey* -s output format ; all lower case, no punctuation. The *hunt* program takes one argument which specifies the base name of the index files to be searched. Only one set of index files can be searched at a time, although many text files may be indexed as a group, of course. If one of the text files has been changed since the index, that file is searched with *fgrep* ; this may occasionally slow down the searching, and care should be taken to avoid having many out of date files.

The following option arguments are recognized by *hunt* :

- a        Give all output ; ignore checking for false drops.
- Cn       Coordination level *n* ; retrieve items with not more than *n* terms of the input missing ; default -C0 , implying that each search term must be in the output items.
- F[ynd]   '-Fy' gives the text of all the items found ; '-Fn' suppresses them. '-Fd' where *d* is an integer gives the text of the first *d* items. The default is -Fy.
- g        Do not use *fgrep* to search files changed since the index was made ; print an error comment instead.
- i *string* Take *string* as input, instead of reading the standard input.
- l n       The maximum length of internal lists of candidate items is *n* ; default is 1000.
- o *string*   Put text output ('Fy') in *string* ; of use only when invoked from another program.
- p        Print hash code frequencies ; mostly for use in optimizing hash table sizes.
- T[ynd]   '-Ty' gives the tags of the items found ; '-Tn' suppresses them. '-Td' where *d* is an integer gives the first *d* tags. The default is -Tn.
- t *string* Put tag output ('Ty') in *string* ; of use only when invoked from another program.

The timing of *hunt* is complex. Normally the hash table is overfull, so that there will be many false drops on any single term ; but a multi-term query will have few false drops on all terms. Thus if a query is underspecified (one search term) many potential items will be examined and

---



discarded as false drops, wasting time. If the query is overspecified (a dozen search terms) many keys will be examined only to verify that the single item under consideration has that key posted.

As would be expected, the optimal search is achieved when the query just specifies the answer; however, overspecification is quite cheap. In general, overspecification can be recommended; it protects the user against additions to the data base which turn previously uniquely-answered queries into ambiguous queries.

### **Files**

@BINDIR/huntExecutable.

### **License**

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD<sup>1</sup> license. The software *hunt* is distributed under the CDDL-1.0<sup>2</sup> license.

### **See also**

refer(1)<sup>3</sup>, referformat(7)<sup>4</sup>, sortbib(1)<sup>5</sup>, hunt(1)<sup>6</sup>, mkey(1)<sup>7</sup>, inv(1)<sup>8</sup>, and *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk.

### **Help and bugs**

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

### **Authors**

M. E. LESK, modified by Pierre-Jean FICHET.

---

<sup>1</sup> See p. 20.

<sup>2</sup> See p. 21.

<sup>3</sup> See p. 3.

<sup>4</sup> See p. 6.

<sup>5</sup> See p. 10.

<sup>6</sup> See p. 16.

<sup>7</sup> See p. 14.

<sup>8</sup> See p. 12.

---

## *Changes of refer*

**2014/03/12 14 :25 :26 VERSION-0.16**

- version.c : troff files : fix link to license.

**2013/10/30 20 :18 :17 VERSION-0.15**

- troff files : fix link to license.

**2013/10/30 16 :27 :50 VERSION-0.14**

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

**2013/10/30 14 :58 :23 VERSION-0.13**

- makefile : FIX : Use new build system.

**2013/04/07 17 :37 :35 VERSION-0.12**

- makefile : s/ROOT/DESTDIR/g (bug report by Seb).

**2013/03/31 07 :51 :24 VERSION-0.11**

- makefile, refer : various typo bug fix.
- refer : close properly firstname small capital.

**2013/03/22 11 :27 :55 VERSION-0.10**

- Add makefile.

**2013/03/22 11 :26 :33 VERSION-0.9**

- refer : strings to define small capitals don't need to be escaped with "\".

**2013/03/22 11 :25 :19 VERSION-0.8**

- refer : do not add "," after lastname if there's no firstname when reversing.

**2013/03/22 11 :24 :16 VERSION-0.7**

- refer : use string "&" to join names instead of "and" and "AND".

**2013/03/22 11 :23 :06 VERSION-0.6**

- refer : add support for editor names in the same way as author names (multiple editor names, small capitals...).

**2013/03/22 11 :21 :46 VERSION-0.5**

- refer : don't truncate utf8 chars when copying strings in comments of file.

**2013/03/22 11 :19 :03 VERSION-0.4**

- refer : add -d option to use strings to define small capitals instead of hardcoded fake small capitals. Strings are : .

**2013/03/22 11 :17 :34 VERSION-0.3**

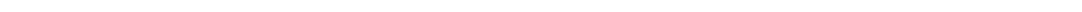
- refer, sortbib : add -i option to sort following iso-690 standart.

**2013/03/22 11 :16 :26 VERSION-0.2**

- ADD : portions copyright and new RCS Id.
-

***2013/03/22 11 :15 :15 VERSION-0.1***

- Initial revision, from Heirloom Refer.



## *Original BSD License*

Copyright (c) 1980, 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met :

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement : This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

# *Common Development and Distribution License Version 1.0*

## **1. Definitions**

### **1.1.**

"Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

### **1.2.**

"Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

### **1.3.**

"Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

### **1.4.**

"Executable" means the Covered Software in any form other than Source Code.

### **1.5.**

"Initial Developer" means the individual or entity that first makes Original Software available under this License.

### **1.6.**

"Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

### **1.7.**

"License" means this document.

### **1.8.**

"Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

### **1.9.**

"Modifications" means the Source Code and Executable form of any of the following :

- A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications ;
- B. Any new file that contains any part of the Original Software or previous Modifications ;  
or
- C. Any new file that is contributed or otherwise made available under the terms of this License.

### **1.10.**

"Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

---

**1.11.**

"Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

**1.12.**

"Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

**1.13.**

"You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

**2. License Grants.****2.1. The Initial Developer Grant.**

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license :

- (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work ; and
- (b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
- (c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
- (d) Notwithstanding Section 2.1(b) above, no patent license is granted : (1) for code that You delete from the Original Software, or (2) for infringements caused by : (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

**2.2. Contributor Grant.**

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license :

- (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work ; and
- (b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of : (1) Modifications made by that Contributor (or portions thereof) ; and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
- (c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
- (d) Notwithstanding Section 2.2(b) above, no patent license is granted : (1) for any code that Contributor has deleted from the Contributor Version ; (2) for infringements caused by :

(i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

### ***3. Distribution Obligations.***

#### ***3.1. Availability of Source Code.***

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

#### ***3.2. Modifications.***

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

#### ***3.3. Required Notices.***

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

#### ***3.4. Application of Additional Terms.***

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

#### ***3.5. Distribution of Executable Versions.***

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

#### ***3.6. Larger Works.***

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

## ***4. Versions of the License.***

---

#### **4.1. New Versions.**

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

#### **4.2. Effect of New Versions.**

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

#### **4.3. Modified Versions.**

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You : (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

#### **5. Disclaimer of Warranty.**

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

#### **6. Termination.**

##### **6.1.**

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

##### **6.2.**

If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

---



**6.3.**

In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

**7. *Limitation of Liability.***

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

**8. *U.S. Government End Users.***

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

**9. *Miscellaneous.***

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

**10. *Responsibility for Claims.***

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

---

***Notice Pursuant to Section 9 of the Common Development and Distribution License (CDDL)***

For Covered Software in this distribution, this License shall be governed by the laws of the State of California (excluding conflict-of-law provisions).

Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

---

---

# Tables

## Table of content

Utroff Refer documentation . . . . .	1
Refer readme . . . . .	2
Presentation . . . . .	2
Build and install . . . . .	2
License . . . . .	2
Help and bugs . . . . .	2
Refer manual . . . . .	3
Name . . . . .	3
Synopsis . . . . .	3
Description . . . . .	3
Example . . . . .	4
Files . . . . .	4
See also . . . . .	4
Notes . . . . .	5
License . . . . .	5
Help and bugs . . . . .	5
Authors . . . . .	5
Refer input and output format . . . . .	6
Refer file . . . . .	6
Refer output . . . . .	6
License . . . . .	8
See also . . . . .	8
Help and bugs . . . . .	9
Authors . . . . .	9
Sortbib manual . . . . .	10
Name . . . . .	10
Synopsis . . . . .	10
Description . . . . .	10
License . . . . .	10
Files . . . . .	10
See also . . . . .	11
Help and bugs . . . . .	11
Authors . . . . .	11
Inv manual . . . . .	12
Name . . . . .	12
Synopsis . . . . .	12
Description . . . . .	12
Files . . . . .	12
License . . . . .	13
See also . . . . .	13
Help and bugs . . . . .	13
Mkey manual . . . . .	14
Name . . . . .	14
Synopsis . . . . .	14
Description . . . . .	14
Files . . . . .	14

---

---

License . . . . .	14
See also . . . . .	14
Help and bugs . . . . .	15
Authors . . . . .	15
Hunt manual . . . . .	16
Name . . . . .	16
SYNOPSIS . . . . .	16
Description . . . . .	16
Files . . . . .	17
License . . . . .	17
See also . . . . .	17
Help and bugs . . . . .	17
Authors . . . . .	17
Changes of refer . . . . .	18
2014/03/12 14 :25 :26 VERSION-0.16 . . . . .	18
2013/10/30 20 :18 :17 VERSION-0.15 . . . . .	18
2013/10/30 16 :27 :50 VERSION-0.14 . . . . .	18
2013/10/30 14 :58 :23 VERSION-0.13 . . . . .	18
2013/04/07 17 :37 :35 VERSION-0.12 . . . . .	18
2013/03/31 07 :51 :24 VERSION-0.11 . . . . .	18
2013/03/22 11 :27 :55 VERSION-0.10 . . . . .	18
2013/03/22 11 :26 :33 VERSION-0.9 . . . . .	18
2013/03/22 11 :25 :19 VERSION-0.8 . . . . .	18
2013/03/22 11 :24 :16 VERSION-0.7 . . . . .	18
2013/03/22 11 :23 :06 VERSION-0.6 . . . . .	18
2013/03/22 11 :21 :46 VERSION-0.5 . . . . .	18
2013/03/22 11 :19 :03 VERSION-0.4 . . . . .	18
2013/03/22 11 :17 :34 VERSION-0.3 . . . . .	18
2013/03/22 11 :16 :26 VERSION-0.2 . . . . .	18
2013/03/22 11 :15 :15 VERSION-0.1 . . . . .	19
Original BSD License . . . . .	20
Common Development and DiStribution License Version 1.0 . . . . .	21
1. Definitions . . . . .	21
1.1. . . . .	21
1.2. . . . .	21
1.3. . . . .	21
1.4. . . . .	21
1.5. . . . .	21
1.6. . . . .	21
1.7. . . . .	21
1.8. . . . .	21
1.9. . . . .	21
1.10. . . . .	21
1.11. . . . .	22
1.12. . . . .	22
1.13. . . . .	22
2. License Grants. . . . .	22
2.1. The Initial Developer Grant. . . . .	22
2.2. Contributor Grant. . . . .	22
3. DiStribution Obligations. . . . .	23
3.1. Availability of Source Code. . . . .	23
3.2. Modifications. . . . .	23
3.3. Required Notices. . . . .	23
3.4. Application of Additional Terms. . . . .	23
3.5. DiStribution of Executable Versions. . . . .	23
3.6. Larger Works. . . . .	23

---

---

4. Versions of the License. . . . .	23
4.1. New Versions. . . . .	24
4.2. Effect of New Versions. . . . .	24
4.3. Modified Versions. . . . .	24
5. Disclaimer of Warranty. . . . .	24
6. Termination. . . . .	24
6.1. . . . .	24
6.2. . . . .	24
6.3. . . . .	25
7. Limitation of Liability. . . . .	25
8. U.S. Government End Users. . . . .	25
9. Miscellaneous. . . . .	25
10. Responsibility for Claims. . . . .	25
Notice Pursuant to Section 9 of the Common Development and Distribution License (CDDL) . . . . .	26
Tables . . . . .	27
Table of content . . . . .	27

---