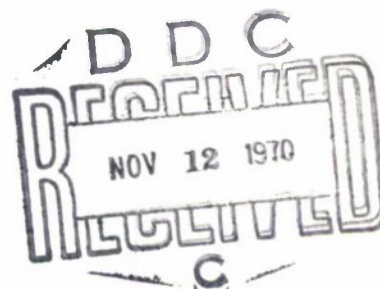


MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

A USER'S GUIDE
TO LISTAR

LINCOLN MANUAL 94

OCTOBER 1970



The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored in part by the Department of the Air Force under Contract F19628-70-C-0230 and in part by the Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institutes of Health, Department of Health, Education, and Welfare, under Contract NLM-69-7.

This document has been approved for public release and sale; its distribution is unlimited.

LEXINGTON

MASSACHUSETTS

20100827156

| | | |
|---------------------------------|---------------|-------------------------------------|
| ACCESSION for | | |
| CFSTI | WHITE SECTION | <input checked="" type="checkbox"/> |
| DOC | DIFF SECTION | <input type="checkbox"/> |
| UNANNOUNCED | | <input type="checkbox"/> |
| JUSTIFICATION | | |
| BY | | |
| DISTRIBUTION/AVAILABILITY CODES | | |
| DISC. | AVAIL. and/or | SERIAL |
| 1 | | |

LISTAR

Lincoln Information Storage and Associative Retrieval System (LISTAR) is an on-line interactive storage and retrieval system which permits a user to define, search, modify, and cross associate data files. LISTAR runs under the IBM CP/CMS Time Sharing System operating on the Lincoln Laboratory IBM 360/67 computer. LISTAR users communicate to the system by way of a keyboard terminal (IBM 2741, IBM 1050, IBM 2260, Advance Remote Display System (ARDS) or teletype).

This User's Guide describes procedures for LISTAR operations. For instructions on CP/CMS procedures, consult CP/CMS User's Guide, Cambridge Scientific Center, IBM Data Processing Division, or call 617-862-5500, Extension 7374.

Prepared by: A. Armenti and S.W. Galley
System Implementation by: A. Armenti, S.W. Galley, R.P. Goldberg, D. Hall, W.H. Joyner, J.F. Nolan, A. Sholl.

LISTAR OVERVIEW

LISTAR is a general purpose on-line interactive system which permits a user to define, search, modify, and cross associate data files to suit his special interests and needs. LISTAR runs under the IBM CP/CMS Time Sharing System.*

File Set Creation

Information describing the internal structure of files created by a user is contained in a directory called the Master File. A collection of files which have a common directory is called a "file set." The limit on the number of files one may create within a file set is determined by the size of auxiliary storage, the size of each file, and the number of file sets. A Master File is automatically generated for the user when he creates a new file set.

File Organization

A file in LISTAR is a set of entries. Entries are made up of data fields. The number of data fields per entry, the number of entries per file, and the number of files per file set are all variable. When the user defines a file, he specifies the name, type, and, where necessary, the maximum length of each data field making up an entry. All entries of a file have the same data fields. Data fields may be added to, or deleted from, any file any time after it has

* "CP/CMS User's Guide," IBM Data Processing Division, Cambridge Scientific Center, October 1967.

been defined. The names of file sets, files and data fields are assigned by the user and may be any alphanumeric designation. A data field may have any one of five designated types: character (alphanumeric), integer, floating point, hexadecimal (binary) or 'user'. The field type specifies the form of the data value for internal representation and for input and output. A data field having a 'user' type accepts data values in a specially coded form determined by the user. A 'user' type field presupposes a user prepared conversion program which may be called by LISTAR for inputting and outputting data values.

A field may be denoted as fixed, variable, repeated fixed, or repeated variable. The user must specify a maximum length for fixed fields and repeated fixed fields. Only character (alphanumeric) and hexadecimal (binary) fields may be typed as variable or repeated variable. The repeated fixed and repeated variable types permit the user to store (retrieve) more than one value in the designated field.

Command Language

The user at a terminal communicates to LISTAR by way of relatively simple English commands. The LISTAR command set is open ended and indefinitely expandable through the addition of command subroutines. The set of commands available to the user is referred to as the LISTAR MENU. The command name and qualifiers required by the command appear in this guide in upper case letters; parameter names and a brief explanation of the command appear in lower case. Where the user may choose among several alternative modes of the same command, the alternatives are placed in parentheses. For these cases, the user must select one of the alternatives. Optional parameters are set off by the paired symbols "<" and ">". Parameters are delimited by spaces. A sequence of parameters of indefinite length is represented by a string containing the name of the first and last parameter separated by three dots (...).

Using LISTAR

The time sharing system operates on two levels named CP and CMS. After log in, the time sharing system is operating at the CP level. The command 'i cms' causes the time sharing system to move to the CMS level. LISTAR operates at the CMS level. To move from the CMS level to the CP level, the user must type a special symbol called the 'attention' symbol. The attention symbol for the teletype is a combination of the control character and the letter 'f' (control-f). Both characters must be pressed at the same time. The attention symbol for the ARDS is either control-f or the special key labelled 'BREAK'. The attention symbol for the IBM 2741 terminal is the special key labelled 'ATTN'. To resume normal operation (i.e., return to CMS) after a move to CP, the user must type the letter 'b' and carriage return. The carriage return for the ARDS is control-s.

To execute LISTAR commands, the user first types 'LISTAR'. The LISTAR system will then be loaded and the user may issue any of the commands listed in the MENU. A LISTAR command consists of a series of "items" on one or more lines, followed by a carriage return. The final carriage return signals the end of a command line. An "item" is a series of characters set off by blanks. An item that is more than one word long must be enclosed in single quote delimiters. Single quotes within a multi word item must be typed as double quotes (") to distinguish them from item

delimiters. For example, the following are all single items:
disease, Parkinson's, 'Parkinson's disease'.

The first set of items in a command line is the command name. The command name is one or more words that specify the kind of operation that LISTAR is to carry out. The other items in a command line supply additional information required by the command. The end of a command line is signalled to the system by a carriage return, unless the last character on the line is a hyphen, in which case, the terminal user continues the command in the usual way, on the following line.

A description of any command can be obtained on-line during a terminal session without reference to a separate manual. The command INFO, followed by the command name, causes LISTAR to type back a prototype of the command together with a written explanation.

LISTAR presents a prompting arrow (==>) to the user to indicate that it is ready to receive a command line. Commands may be issued in any order. LISTAR completes one command before signalling for another. Commands which call for lengthy execution, e.g., printing many entries, can be terminated in mid execution. To interrupt execution, the user strikes the 'attention' key then the character 'e' and carriage return. LISTAR will respond by typing a prompting arrow, indicating it is ready to receive another command.

Useful Features of Lincoln Laboratory's Time Sharing System*

1. An input line is terminated by a carriage return. The carriage return on the teletype or the Advance Remote Display Station (ARDS) is executed by pressing the control key and the letter 's'.
2. Typographical errors during input may be corrected by typing special symbols. The symbol '@' is called a character-delete symbol and is used to delete the preceding character. If n character delete symbols are typed, the preceding n characters (and the character delete symbols) are deleted from the input line. The symbol '|' is called a line delete symbol and is used on the teletype and the ARDS to delete all the preceding characters in the input line including itself. The character 'z' is the line delete symbol on the IBM 2741.
3. A number of commands to be executed in sequence may be typed as a single input line. Each of these commands must be separated by a special symbol called a break symbol. The break symbol for all terminals is '#'.
4. To suppress typing or display without stopping program execution, the user must press attention twice, type the letters 'kt' (kill typing) and then execute a carriage return. To resume typing or display, the user must press attention twice, type the letters 'rt' (resume typing) and then execute a carriage return.
5. Messages can be sent to other users when the system is operating at CP level. For example, to send the message 'hello' to the user named USER1, type the following:
user1 hello

*CP/CMS is undergoing continual change to improve system performance. The user should consult the User's Guide for changes to conventions described here.

LISTAR MENU

LISTAR MENU

?

'?' causes LISTAR to print or display all LISTAR commands at the terminal. '?' should be executed whenever the user is in doubt about the name or spelling of a command or does not know what commands are included in the menu.

INFO command-name(s)

'INFO' causes a description of the command(s) whose name(s) are given to be typed.

ADD <count> <ENTRY> <TO> file-name

'ADD ENTRY' permits the user to add new entries to a LISTAR file from the terminal. LISTAR displays the name of each data field of the file in turn, and the user types in the field value. When all fields have been presented, the system inserts the new entry into the file in its proper place, according to the chief-field value. The words 'ENTRY' and 'TO' may be included at the user's convenience. If 'count' is omitted, only one entry will be added to the file.

ADD FIELDS field-name-1 field-type-1 field-length-1 ... - field-name-n field-type-n field-length-n <TO> file-name

'ADD FIELDS' causes the given data fields to be defined as additional fields for the specified file. The rules for field names, types, and lengths are the same as for DEFINE FILE. A chief field cannot be added to a file; it must be supplied at the time the file is defined.

APPEND marker-name field-name new-value-1 ... new-value-n

'APPEND' adds the new values at the end of the field of the entry at which the given marker-name is located. This command may be used only with repeated fields.

BULK BACKSPACE (TAPn) format-name
(cms-filename cms-filetype) <count>

'BULK BACKSPACE' moves a tape (TAPn) or disk file (cms-filename cms-filetype) backward the number of entries specified by count. The record characteristics, i.e., block and record size, are taken from the bulk format specified in the command. If 'count' is omitted, BULK BACKSPACE will move the tape or disk file back one entry. If tape is to be backspaced, the logical tape number must be given as TAPn where n is an integer from 1 to 4.

BULK FORMAT format-name record-length records-per-entry -
field-start-1 <TO> field-end-1 field-name-1 ... -
field-start-n <TO> field-end-n field-name-n

'BULK FORMAT' supplies control information for a subsequent bulk transfer of data between a file on bulk storage (tape or disk) and a LISTAR file. The format name can be any alphanumeric string of characters. Only the first four characters will be taken as a format name. The record length of the record on auxiliary storage must be specified in bytes. The number of records per entry must be an integer greater than or equal to one. The maximum record size is 240 bytes. For each field on the bulk storage file, the user must specify the beginning and ending location in the entry. The field names in the format must correspond identically to the field names of the LISTAR file without regard to order.

Example: Assume an input tape having 10 fields per record and 1 record per entry. Each record is 200 bytes long. A LISTAR file has been defined having 10 fields corresponding to the fields of the tape entries. The first field of the entries on tape is a date and its values appear in byte locations 1 to 6. Another field is a book title and its values appear in byte locations 160 to 200. Then 1,6 would be given as field-start-1,field-end-1 with 'date' as field-name-1 and 160,200 would be given as field-start-10, field-end-10 with 'book title' as field-name-10. Similar specifications would be given for each of the other 8 fields.

The field locations on tape must be specified in ascending order. The numbering of field locations extends over the entire entry, regardless of the number of physical records making up the entry. Field locations must not cross physical record boundaries. All fields need not be included in the format, only those for which input values are desired. A user may have up to four bulk formats defined simultaneously during a session.


```

BULK INPUT (TAPn                ) listar-file-name
            (cms-filename cms-filetype )      format-name
            (TERMINAL              )          <count>

```

'BULK INPUT' causes entries to be read from the specified bulk storage device into the LISTAR file according to the named format. If the input medium is tape, the logical tape number must be specified as TAPn where n is an integer from 1 to 4. If the medium is the user's disk, the cms-filename and cms-filetype of the input file must be specified. The command BULK REWIND must be issued before BULK INPUT to begin reading at the head of a disk file. The format-name designates a previously defined bulk format. 'Count' must be an integer specifying the number of entries to be read from the input file. If 'count' is omitted and input is from TERMINAL, BULK INPUT reads one entry into the file. If 'count' is omitted and input is from disk or tape, BULK INPUT reads all input entries into the file, i.e., it continues reading until it reaches end-of-file on the disk or tape.

BULK LIST

'BULK LIST' lists by name all the bulk formats currently defined.

BULK PRINT format-name

'BULK PRINT' displays a description of the specified bulk format.

```

BULK REWIND (TAPn                )
            (cms-filename cms-filetype )

```

'BULK REWIND' rewinds a tape (TAPn) or disk file (cms-filename cms-filetype). If a tape is to be rewound, the logical tape number must be specified as TAPn where n is an integer from 1 to 4.

```

BULK SKIP (TAPn                ) format-name <count>
          (cms-filename cms-filetype )

```

'BULK SKIP' move a tape (TAPn) or disk file (cms-filename cms-filetype) forward the number of entries specified by the count. The record characteristics (block and record size) are taken from the bulk format named in the command. If 'count' is omitted, BULK SKIP will move the tape or disk file forward one entry. If tape records are to be skipped, the logical tape number must be specified as TAPn where n is an integer from 1 to 4.

CHANGE marker-name field-name <id> old-string new-string

'CHANGE' scans the value of the given field of the entry at which marker-name is located. If found, it is replaced by the new string. The field type must be either 'C' or 'X'. If the field is repeated, enough of the field value to identify the value to be changed must be given as the id.

CMS cms-command-name argument-1 ... argument-n

'CMS' executes the designated CMS command. Only commands specifically allowable by CMS may be executed. Users not familiar with CMS should refer to the CMS user's guide for special applications of this command. The CMS commands listed below are particularly useful. Each command should be preceded by 'CMS' when issued.

BREAK character

The break symbol is used to signify a carriage return. A series of LISTAR commands typed on one line with a break symbol separating command lines will be executed as a sequence of individual commands in the order typed. 'CMS BREAK' sets the break symbol to the specified keyboard character. If this command is not issued, CMS will assume the number character (#) to be the break symbol.

CLOSIO PRINTER

'CLOSIO PRINTER' prints the output that has been directed to the high speed printer. This command is used in conjunction with SELECT DATOUT PTR. This SELECT command directs all output to a special disk area for printing. Printing will not occur until the CLOSIO command is issued or the user logs off.

ERASE fileset-name fileset-type

'ERASE' erases the CMS file on disk storage. The CMS file may be a LISTAR fileset. The user should be cautious about executing this command to avoid losing important files. Very large files containing valuable information should be stored on magnetic tape for security.

INPUT cms-filename cms-filetype

'INPUT' creates a disk file with the specified name and type. This command is particularly useful for creating a disk file of LISTAR commands which can be executed as a TYPIN file by SELECT TYPIN DISK. The user creates the file by typing successive lines, each up to 80 characters long. The user terminates INPUT by typing a blank line, i.e., an extra carriage return.

LISTF cms-filename cms-filetype

'LISTF' displays the name, type, mode, size, and date/time information of the specified CMS file at the terminal. The date/time records the time of the most recent change to the file.

OFFLINE PRINT cms-filename cms-filetype <n1> <n2>

'OFFLINE PRINT' prints a CMS disk file on the high-speed printer beginning at line number n1 and ending with line number n2. If n1 and n2 are omitted, the entire file is printed. If n2 is omitted, the lines from n1 to the end of the file are printed.

PRINTF cms-filename cms-filetype <n1> <n2> <n3>

'PRINTF' (Print File) is like OFFLINE PRINT except that the disk file is printed on the typewriter terminal. The additional argument, n3, is the number of leading characters of each line to be printed; if omitted, the first 80 characters of each line are printed.

STAT

'STAT' (Status) prints status information regarding the user's disk files on the typewriter terminal. The status information includes the total number of records assigned to the user's account, the number of records used, and the number available for use.

COPY old-marker-name new-marker-name-1 ... new-marker-name-n

'COPY' places additional new markers with the given names at the entry where the first named marker is located. If a new marker is already active, it is moved to the old marker location. This permits the user to keep track of a file position of special interest while moving other markers through the file.

DEFINE FILE file-name
field-name-1 field-type-1 field-length-1 ... -
field-name-n field-type-n field-length-n

DEFINE FILE creates a file description for the given file. No data are entered into the file. The file name and field-names cannot be longer than 16 characters. A field length for fixed length fields must be designated as an integer within a range determined by the field type. The field types, their abbreviations for the DEFINE FILE command, and the corresponding length ranges are the following:

| TYPE | ABBR | ACCEPTABLE LENGTHS |
|-------------|------|--------------------------|
| character | c | 1-255 characters |
| integer | i | 1-10 decimal digits |
| real | r | 1-16 significant digits |
| hexadecimal | h | 1-510 hexadecimal digits |
| user | u | 1-255 internal bytes |

Repeated Field:

To indicate that a field is a repeated field, i.e., one that will accept multiple field values, the user types r as a suffix to the type code. For example, 'cr' is the code for a repeated character field.

Variable Field:

To indicate that a field has a variable length, the user types v for field length. Only a field of type 'c' or 'h' may have a length designation 'v'. The maximum size of a variable field value is 4000 characters or 8000 digits.

Chief Field:

At least one field must be designated the chief field, i.e., the field of primary ordering. To indicate that a field is the chief field, the user prefixes the type code with an asterisk (*). For example, '*c' is the code for a chief field with a character type. Fields typed repeated (r), variable (v), and user (u) cannot be chief fields.

DELETE <ENTRY> (chief-field-value FROM file-name)
 (marker-name)

'DELETE' deletes an entry from a LISTAR file. If a chief-field value is given, the first entry found with its chief field equal to the given value is deleted. If a marker name is given, the entry at which the marker is located is deleted. All markers located at a deleted entry are moved to its successor.

DELETE FIELDS field-name-1 ... field-name-n <FROM> file-name

'DELETE FIELDS' causes the given fields to be removed from every entry of the file. The fields are also removed from any existing formats and bulk formats.

DESCRIBE FORMAT format-name

'DESCRIBE FORMAT' prints the field names assigned to the given format on the typewriter terminal. For a TAB type format column numbers are also printed.

EDIT

'EDIT' puts LISTAR into a special mode that allows the user to modify the previous command, execute it again, save it on the disk, or all of these. In the edit-mode the user issues special edit-commands in response to the prompting arrow, E==>. The available edit-commands are:

PRINT

The command (or REPEAT-sequence) is typed out for inspection.

DELETE item-1 ... item-n

The first occurrence of each item given is deleted from the command.

INSERT preceding-item item-1 ... item-n

The given items are added to the command immediately following the first occurrence of the item first given.

(Note: since there is no "change" edit-command, changing X to Y is done by INSERT X Y followed by DELETE X.)

NPRINT <number>

Like PRINT, except that each item is numbered and appears on a separate line. If a REPEAT-sequence is being edited, succeeding commands are separated by a dashed line. If a number is given as an argument, only items with numbers equal to or greater than the number are typed out.

NDELETE item-number-1 ... item-number-n*

Like DELETE, except that the items to be deleted are designated by their numbers.

NINSERT preceding-item-number item-1 ... item-n*

Like INSERT, except that the "preceding item" only is designated by its number.

NSEPARATE item-number-1 ... item-number-n*

For each number in turn, the corresponding item is "separated" from the one following it, so that the two are parts of separate commands.

```
SAVE (-----)
      (?)
      (cms-filename cms-filetype <line-number>)
```

'SAVE' stores the last command line or command sequence in a CMS file on disk. The disk file can later be selected as a TYPIN file (see SELECT TYPIN DISK). The first time SAVE is issued, the user must name a new or existing disk file to be used for saving the command lines. The saved lines will be appended to the disk file beginning with 'line-number' or at the end of the file if no line-number is given. Subsequent SAVE commands will assume the most recently named file if SAVE is issued without arguments (-----). If 'SAVE ?' is issued, the current file and line number are displayed and no lines are saved.

QUIT

The system leaves the edit-mode without executing the command.

GO

The system executes the command as it stands and then returns to normal LISTAR.

?

The names of the available edit-commands are typed out.

* NPRINT must be issued before any other "N-commands." Afterwards, the item numbers assigned by NPRINT remain unchanged until another NPRINT is done.

ERASE file-name

'ERASE' completely erases the given LISTAR file.

FIND file-name format-name <WHEN> -
field-name-1 condition-1 value-1 AND -
... AND field-name-n condition-n value-n

'FIND' performs a serial boolean search of the specified file and displays entries that satisfy the conditions according to the designated format. The request is restricted to a conjunctive form, i.e., the terms of the search can only be joined by 'and'. Each term of the conjunction must include a field-name, a condition and reference value.

FORMAT format-name file-name -
(NORM <field-name-1 ... field-name-n >)
(TAB <field-name-1 <column-1> ... fld-name-n <col-n>>)

'FORMAT' is used to specify formats for displaying contents of entries. Each format is given a name and can be applied only to entries in the file named in the command. Data fields are displayed in the order given in the command. The user may have several formats defined at any one time. A format may be deleted and another defined in its place at any time. To delete a format, the user types 'FORMAT' and the format name only. There are two kinds of formats as follows:

- (1) NORM -- Each field value is displayed on a separate line, preceded by the field name and a colon. The name of a repeated field appears with the first value only; subsequent values are indented and set off by a colon. If no field names are given in the command, all fields that make up the entry are displayed.
- (2) TAB -- The field values are printed on one line, at the column-positions given in the format specification. Successive values of a repeated field are printed one under another. The column numbers should be given in ascending order. If a column number is omitted, the system assigns a suitable number automatically; if no fields are given, all fields will be printed, with column numbers assigned automatically. A heading consisting of the field names may be printed to label the columns of field values resulting from successive entry printings.

If only the 'format-name' is given in the command, that format is deleted.

INDEX file-name

'INDEX' creates an index for the given file as a permanent part of the file set. The index is constructed on the chief field. Thereafter MOVE TO and DELETE ENTRY FROM operate more quickly on that file.

LA file-name string

'LA' (Locate All) prints all chief-field values of the given file in which the string of characters given by 'string' occurs.

LAF file-name field-name string

'LAF' (Locate All Field) prints all field values of the given file and field in which the string of characters given by 'string' occurs.

```
LIST ( FILES          )  
      ( FIELDS file-name )  
      ( FORMATS        )  
      ( MARKERS        )
```

'LIST FILES' displays the names of all the files in the current file set.

'LIST FIELDS' displays the names, types, and lengths of all fields in the given file.

'LIST FORMATS' displays the names and types of all formats specified during current operations on this file set.

'LIST MARKERS' displays the names of all markers currently active and the file each is in.

LOAD filesset-name < filesset-type <filesset-mode> >

'LOAD' loads the given file set from disk storage into computer memory. If the file set does not exist, it will be created. If 'filesset-type' is omitted, it will be taken to be 'LISTAR'. If 'filesset-mode' is omitted, all available disks (including shared ones) will be searched to find the file set; otherwise only the given disk will be searched. Any formats previously specified or markers currently active will be deleted. The user should note that a file set on a shared disk can be searched but not modified.

MOVE marker-name <count> <field-name <zone> condition value>

'MOVE' moves the given marker down a file until either the number of entries given by 'count' has been examined or the field-condition is met. The entry at which the given marker is located is examined first. If 'count' is omitted when a field-condition is given, it will be taken to be very large. A field name may be any field that occurs in the file description (see LIST FIELDS). The zone (optional) is an integer telling how many characters in the data field should be examined. The allowable conditions are the following:

- EQ -- EQUAL TO
- NE -- NOT EQUAL TO
- LT -- LESS THAN
- LE -- LESS THAN OR EQUAL TO
- GT -- GREATER THAN
- GE -- GREATER THAN OR EQUAL TO

(Note that, for character fields, 'LESS THAN' means 'precedes alphabetically', etc.)

The 'value' can be any legal value for the type of 'field'. If the field-condition is omitted, the marker will be moved unconditionally the number of entries given by 'count'. If 'count' is also omitted, the marker will be moved to the next entry in the file.

MOVE marker-name UP <count> <field-name <zone> condition value>

'MOVE ... UP' is identical to MOVE except that it moves the marker up a file rather than down.

MOVE marker-name TO chief-field-value

'MOVE ... TO' moves the marker directly to the entry having the given value for its chief field. Only the number of characters needed to identify the entry uniquely need be given.

PMOVE marker-name <count> <field-name <zone> condition value >

'PMOVE' (Print Move) is the same as MOVE except that the chief-field value is printed after the move.

PMOVE marker-name UP <count> <field-name <zone> condition value>

'PMOVE ... UP' is identical to PMOVE except that it moves the marker up the file rather than down.

PRINT (marker-name) format-name < begin-column <end-column> >
(HEADING)

'PRINT' causes selected data-field values (in the entry at which the given marker is located) to be printed according to the given format. If 'marker-name' is HEADING and the given format is a TAB type, then selected data-field names will be printed. If 'begin-column' is given, only the output in that column and following columns is printed; if omitted, it is assumed to be 1. If 'end-column' is also given, only the output in that column and preceding columns is printed; if omitted it is assumed to be very large.

PUT marker-name FILE file-name

'PUT ... FILE' puts the given marker at the first entry of the named file. If the marker is not active, it will be made active.

REMOVE marker-name field-name <id>

'REMOVE' resets the value of the given field of the entry at which marker-name is located to the initialization value. If the field is repeated, enough of the field value to identify the value to be removed must be given by the id; the value is removed completely unless it is the only one, in which case it is reset to the initialization value.

REPEAT number-of-times

.
.
.
END

'REPEAT ... END' allows a sequence of commands to be executed repeatedly. When the user issues this command, the system allows him to type in the sequence, one command per line. The sequence must be terminated with the 'END' command; then the sequence is executed the number of times given. Any command can be included in the sequence, including other 'REPEAT ... END' pairs. If 'number-of-times' is 0, the following sequence will be skipped over without execution; if it is omitted, it is assumed to be very large.

REPLACE marker-name field-name <id> new-value

'REPLACE' replaces the value of the given field of the entry at which marker-name is located by the new value. If the field is repeated, enough of the field value to identify the value to be replaced must be given as the id.

SEARCH file-name format-name

'SEARCH' searches the entire file and prints, using the given format, all entries that satisfy certain conditions. After the user enters the command line as above, he is presented with a numbered prompting symbol '1 : '. On this line the first condition should be typed in the form:

field-name test reference-value

where the following are legal tests: =, >, <, EQ, NE, LT, GT, GE, LE, HAS. 'HAS' is valid only for a character field and is satisfied if the string of characters given as the 'reference-value' occurs anywhere in the value of the given field. After the user types the first condition, a second prompting symbol '2 : ' will be presented, and another condition may be given. At any time after the first condition is given, a response to a prompting symbol consisting of a single carriage return will signal the end of inputting of conditions and will lead to the following prompting arrow: 'S==>'. At this time the conditions may be combined in any manner using the logical operators '&' (AND), 'v' (OR), and '~' (NOT). There must be no embedded blanks in the logical expression. The conditions are referred to by the number or letter in the prompting arrow, i.e., '1', '2', etc. Thus a sample combination would be 1&(2v3)&4. Note that '&' has precedence over 'v' and thus 1&2v3 means (1&2)v3 and not 1&(2v3).

```

SELECT TYPIN (TYPE
              (TYPN
              (DISK filename filetype <begin-line <end-line>>))
            )

```

```

SELECT DATOUT (?
              (<TYPE <line-length>> <PTR> -
              <DISK filename filetype <line-no> <line-len>>))

```

'SELECT' chooses one or more units to be used for input or output.

'SELECT TYPIN' specifies from which source LISTAR will obtain its commands. The user can designate any of the following as input sources:

- (1) TYPE -- the typewriter terminal. This source is taken if no selection is stipulated after TYPIN.
- (2) TYPN -- the typewriter terminal. Letters are read in both upper and lower case. They are not automatically converted to upper case.
- (3) DISK -- the CMS file with the given name and type, starting at the line whose number is given first (the first line if this number is omitted) and stopping after the line whose number is given second (the last line if this number is omitted).

'SELECT DATOUT' specifies the unit to which LISTAR will direct output resulting from the commands 'PRINT', 'SEARCH', 'LA', and 'LAP'. Note that more than one unit may be specified and the output will appear identically on each one. The output choices are the following:

- (1) ? -- used to query LISTAR. The current output unit will be identified.
- (2) TYPE -- the typewriter terminal. A line length may be specified. If an output line is longer than this length, it will appear on succeeding lines. The longest line available for TYPE is 119.
- (3) DISK -- the CMS file. The output will be stored at line locations beginning with line-number. A line length may be specified. The line length for DISK may not exceed 800.
- (4) PTR -- the offline printer. Output is directed to a special buffer area (not to disk). When the command 'CMS CLOSIO PRINTER' is issued, the buffer is printed on the high-speed printer. The line length is 132.

If 'SELECT' is not issued or no arguments are given with it, the typewriter terminal is taken as the input/output device and the line length is set to 119.

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

| | | | |
|---|--|--|-------------------------|
| 1. ORIGINATING ACTIVITY (Corporate author) Lincoln Laboratory, M.I.T. | | 2a. REPORT SECURITY CLASSIFICATION Unclassified | |
| | | 2b. GROUP None | |
| 3. REPORT TITLE A User's Guide to LISTAR | | | |
| 4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Lincoln Manual | | | |
| 5. AUTHOR(S) (Last name, first name, initial) Armenti, Amedeo W. and Galley, Stuart W. | | | |
| 6. REPORT DATE October 1970 | | 7a. TOTAL NO. OF PAGES 36 | 7b. NO. OF REFS None |
| 8a. CONTRACT OR GRANT NO. F19628-70-C-0230 | | 9a. ORIGINATOR'S REPORT NUMBER(S) Lincoln Manual 94 | |
| b. PROJECT NO. NLM-69-7 | | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) ESD-TR-70-317 | |
| c. | | | |
| d. | | | |
| 10. AVAILABILITY/LIMITATION NOTICES This document has been approved for public release and sale; its distribution is unlimited. | | | |
| 11. SUPPLEMENTARY NOTES None | | 12. SPONSORING MILITARY ACTIVITY Air Force Systems Command, USAF Department of Health, Education, and Welfare | |
| 13. ABSTRACT This User's Guide describes procedures for LISTAR operations. Lincoln Information Storage and Associative Retrieval system (LISTAR) is an on-line interactive storage and retrieval system which permits a user to define, search, modify, and cross associate data files. LISTAR runs under the IBM CP/CMS Time Sharing System operating on the Lincoln Laboratory IBM 360/67 computer. LISTAR users communicate to the system by way of a keyboard terminal (IBM 2741, IBM 1050, IBM 2260, Advance Remote Display System (ARDS) or teletype). | | | |
| 14. KEY WORDS LISTAR time-sharing computer systems data retrieval IBM-360/67 computer language | | | |