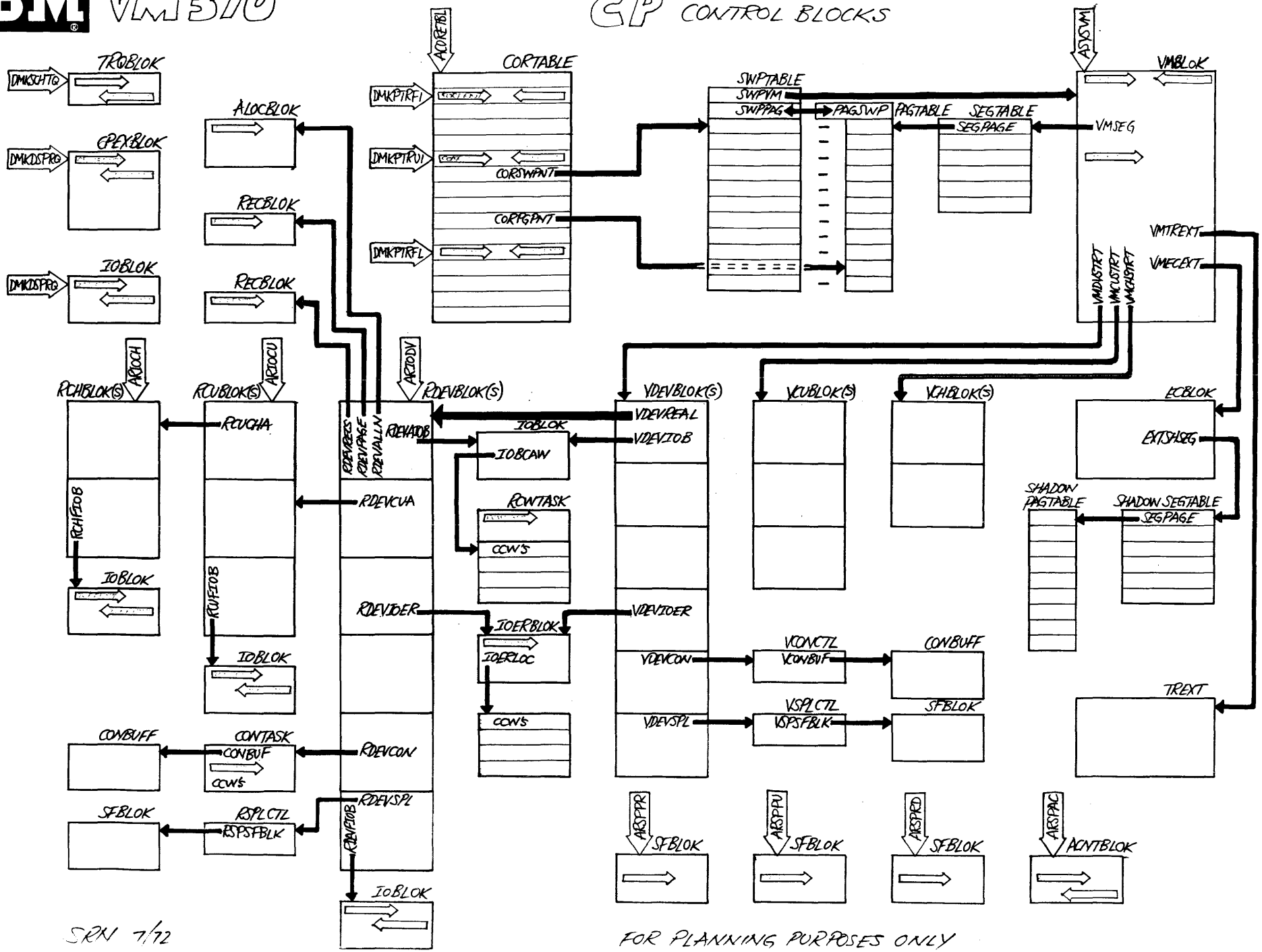
IBM VM370

CP CONTROL BLOCKS



SRN 7/72

FOR PLANNING PURPOSES ONLY

IBM VIRTUAL MACHINE FACILITY
EDIT GUIDE

PRELIMINARY PUBLICATION

Post/Preliminary Draft

IBM Virtual Machine Facility/370 (VM/370)

ZDIT Guide

March 28, 1972

File Number: S370-2-0
Order Number: GCnn-nnnn-0

IBM Virtual Machine Facility (VM/370)
EDIT Guide

This publication explains for CMS users how to use the CMS editor to create or modify files. It contains descriptions of:

- The EDIT command
- The EDIT subcommands
- The EDIT macros

Examples are also included.

IBM Virtual Machine Facility (VM/370), Command Language User's Guide, GCnn-nn is a prerequisite for a thorough understanding and for the effective use of this publication.

For titles and abstracts of associated publications, see the IBM System/360 and System/370 Bibliography, GA22-6822.

VM/370 Release 1

First Edition (November 1972)

This edition applies to Release 1 of IBM Virtual Machine Facility/370 and to all subsequent releases until otherwise indicated in editions or Technical Newsletters. Changes are continually made to the specifications herein: before using this publication in connection with the operation of IBM systems, consult the latest System/360 and System/370 SRL Newsletter, GN20-0360, for the editions that are applicable and current.

Request for copies of IBM publications should be made to your IBM Representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, CP-67/CMS Development Group, 545 Technology Square, Cambridge, Massachusetts, 02139. Comments become the property of IBM.

ERPREFAC 00007

PREFACE

ERPREFAC 00009 This publication is intended for system programmers,
ERPREFAC 00010 application programmers, and terminal users. It describes
ERPREFAC 00011 how to use the CMS EDIT program to create or modify files at
ERPREFAC 00012 the terminal.

ERPREFAC 00013 It is divided into three sections, contains two appendices
ERPREFAC 00014 and a glossary of terms.

ERPREFAC 00015 The Introduction describes the facilities of the editor and
ERPREFAC 00016 how it operates. Subcommand and macro conventions are also
ERPREFAC 00017 included.

ERPREFAC 00019 The section EDIT Subcommands and Macros discusses each
ERPREFAC 00021 subcommand and macro in detail.

ERPREFAC 00023 Examples, How to Use the Editor will contain examples of
ERPREFAC 00025 creating and modifying files using the editor.

ERPREFAC 00027 Appendix A., Summary of EDIT Subcommands, lists the
ERPREFAC 00028 subcommands with their parameters.

ERPREFAC 00030 Appendix B., Editor Messages, lists the editor messages and
ERPREFAC 00032 contains a brief description for each message.

ERPREFAC 00033 The Glossary contains some of the terms that are used in
ERPREFAC 00034 this publication.

ERPREFAC 00035 Other related publications for the effective use of this
ERPREFAC 00036 manual are:

ERPREFAC 00039 IBM Virtual Machine Facility (VM/370), System Messages,
ERPREFAC 00040 GCnn-nnnn

ERPREFAC 00044 IBM Control Program-67/Cambridge Monitor System (CP-67/CMS)
ERPREFAC 00047 Version 3, CMS Script Users Manual, GH20-0860.

ERPIGS 00001 IBM Data Processing Glossary, GC20-1699

ERFIGS 00003

FIGURES

INTRODUCTION

The CMS Editor is a facility that allows you to:

- Create, from a terminal, sequential files containing either fixed or variable length records
- With certain filetypes, use the standard default values for record length, record format, tab settings, serialization, linemode, and lowercase to uppercase translation
- Add, delete, and/or change any part of a file
- Move one or more lines in a file from one place to another
- Extract all or part of a file and make it a new file or imbed it into another file
- Search and change a file through context-direction, line displacement, and line number
- Receive prompting with line numbers if desired
- Display at a terminal all or part of a file
- Use the CMS EXEC interpreter to utilize user written macros

The editor operates in either one of two modes, INPUT and EDIT. You control the mode in which the editor operates by the instructions that you enter at the terminal.

Associated with each file is a line pointer. The line pointer points to the line or record of the file that you are processing. When in INPUT mode, the editor can assign a number to the line of data that you are entering. The record number is incremented for each line of data. Thus, when in EDIT mode, you can refer to a line of data by its number; you can also refer to all or part of a record by indicating the specific contents of the record to be processed. The latter method of processing can be especially useful when it is necessary to locate/change a specific word or phrase within the file because it relieves you of the task of keeping track of record numbers and displacements.

OPERATION OF THE EDITOR

Editing is performed upon a copy of the file that is in main storage. This approach provides for rapid movement both forward and backward through the file. It does, however, limit the files to be edited to those that can be wholly contained within the available virtual storage. It is possible to perform editing operations upon larger files by splitting the file into smaller files that can be handled by the editor.

LINE POINTER

Associated with each file is a line pointer. The line pointer identifies the line in the file that is considered to be the current line. The current line is defined as the line that is being created or edited in the file.

Various EDIT subcommands and macros are provided for moving the current line pointer. You can move the line pointer to:

- A specific record (indicated by its record number)
- A record specified by its relative displacement (in number of records) forward or backward from the current position of the pointer
- A record containing a specified string of characters or having a specified label.

The latter ability to search for strings allows you to be concerned with only the textual context, relieving you of keeping track of record numbers and counts of inserted and deleted records.

When you change the mode from INPUT to EDIT, the pointer is positioned to the last line entered from the terminal, from EDIT to INPUT, the pointer is positioned such that the lines being entered follow the last line edited.

After you issue the EDIT command and enter EDIT mode, the pointer is positioned at a null line that the editor places in front of the first line of the file. If, during the processing of the file, a TOP or an UP subcommand is issued, the pointer is positioned at this null line. The null line is not written onto direct access storage, nor is it displayed at the terminal. If you issue a TYPE subcommand when the pointer is positioned at the null line, a null line with a carriage return is typed.

If you enter a null line while in INPUT mode, the mode changes to EDIT. If you enter a null line while in EDIT mode, the confirming message EDIT: is printed. To change from EDIT mode to INPUT mode, issue the INPUT subcommand without any parameter.

INPUT MODE

The INPUT mode is indicated by the message INPUT:, a carriage return, and the unlocking of the keyboard. You can then type successive lines of input to the file. To insert a blank line in a file, type at least one space and then press the carriage return key. A null line (that is, a carriage return with no prior blanks or characters) entered when in INPUT mode does not add a blank line to the file, it changes the mode to EDIT

EDIT MODE

The EDIT mode is indicated by the message EDIT:, a carriage return, and the unlocking of the keyboard. You can then make changes to a file by issuing EDIT subcommands and macros. All changes to the file become effective immediately in main storage, thus, allowing subsequent modifications to be made to a file. The changed file is written onto direct access storage with the FILE and SAVE commands.

If you issue a QUIT subcommand, then the changes are ignored and the original file is kept as it existed before any changes were made. If the file was newly created, it is not written onto disk.

EDIT Responses

Verification, invoked by the VERIFY subcommand, is the normal response mode of the editor. It causes each line that you have changed or found to be displayed at your terminal. If you turn verify off, normal editor messages are printed. However, to display a line at a terminal, you must issue the TYPE command.

OPERATIONAL CHARACTERISTICS

MAIN STORAGE EXCEEDED

If the available main storage becomes full, the editor issues the message AVAILABLE STORAGE IS NOW FULL. It is then possible to add new lines only if existing ones are deleted first. If you attempt to insert a line (or a file) when there is no room, the editor responds NO ROOM, ensures that EDIT mode is set, deletes any stacked lines and issues the message STACKED LINES CLEARED in order to avoid multiple error messages.

FULL DISK CONDITION

In executing a FILE or SAVE subcommand, the editor writes a temporary work-file, called CMSUT1, alters the name to that required, and erases the old copy of the file, if it exists. During this process, if the output disk becomes full, CMS types the message **z-DISK (xxx) IS FULL **. The editor immediately erases the work-file (which will be incomplete), and issues the message SET NEW FILEMODE, OR ENTER CMS SUBSET AND MAKE MORE ROOM. Thus, the FILE or SAVE subcommand have no overall effect except to produce the messages. A similar response follows an attempt to create more than the maximum number of CMS files.

TOP AND END-OF-FILE CONDITIONS

At the top of every file there is a null line. It is not a part of the file, is present only during editing, and cannot be modified. If printed, it appears as a blank line. There is also an End-of-File (EOF) condition. It occurs if you attempt to go beyond the last line of the file. The message EOF: is printed.

EDIT subcommands, when issued at the top of the file or at EOF behave as follows:

Subcommand	Effect at the top of the file.
ALTER, CHANGE	If the subcommand applies to the top line only, the message NOT FOUND is issued; to more than one line, the number is decremented by one before proceeding to the next line.

DELETE		DELETE is processed as if it were successful, but the top line is not deleted.
GETFILE		The new lines obtained are inserted in the file and the last line that is inserted becomes the current line.
DOWN, NEXT TYPE, STACK		The top line is processed as a blank line.
OVERLAY		If OVERLAY applies only to the top line, it has no effect; to more than one line, the number is decremented before proceeding to the next line.
REPLACE		The new line is inserted and becomes the current line.
TOP, UP		Have no effect.

Subcommand	Effect when the EOF Condition exists
ALTER, CHANGE, FIND, LOCATE	The EOF condition is cleared, and the subcommand is processed as if it were issued a the top of the file.
BOTTOM	The EOF condition is cleared, and the last line of the file becomes the current line.
DELETE, DOWN, NEXT, OVERLAY STACK	If verification is off, the subcommand is ignored; if on, the message EOF: is issued.
GETFILE	The EOF condition is cleared, the new lines obtained are inserted after the last line of the file, and the last line obtained becomes the current line.
TYPE	Produces the message EOF:.
INPUT REPLACE	The EOF condition is cleared, and the new level(s) are inserted after the last line.
UP	UP is processed as if the current line was null line following the last line of the file.

SUBCOMMAND AND MACRO CONVENTIONS

The conventions used in this publication to illustrate EDIT subcommands and macros are as follows:

1. Uppercase letters and punctuation marks (except as described in items 3 through 6) represent information that must be coded exactly as shown.

2. Lowercase letters and terms represent information that you must supply.

3. Information contained within brackets [ ] represents an option that you can include or omit, depending on the requirements of the program.

4. Options contained within braces { } represent alternatives, one of which must be chosen.

5. An ellipsis (a series of three periods) indicates that a variable number of items may be included.

6. Some subcommands may be specified by entering the subcommand or any portion of the subcommand from it's minimum acceptable truncation to the complete subcommand. For example, the minimum truncation for ALTER is AL; thus, AL, ALT, ALTE, or ALTER will invoke the ALTER subcommand. Those subcommands that can be truncated are contained within braces with the subcommand separated from its minimum truncation by a vertical bar (logical OR).

7. Underlined elements represent an assumed option in the event a parameter is omitted.

8. When options appear in a string, they are separated by a vertical bar (logical OR).

For all entries for the filename, filetype and filemode options, you must adhere to the conventions for these options as described in the discussion of the CMS File Management System in VM/370 Command Language User's Guide, GXnn-nnnn.

EDITOR ERROR PROCEDURES

All subcommands and macros are fully checked for validity. A subcommand or macro which is invalid is not processed, the count for the REPEAT subcommand is reset to 1, and the message ? or ?EDIT: ... is issued. If a valid subcommand or

macro fails during execution, a more detailed message is typed.

Also, during a terminal session, the editor communicates status information pertinent to current requests. These messages do not have an associated error code and are not indicative of a condition serious enough to require termination of editor processing. The messages are issued primarily to inform you of possibly unexpected circumstances in subcommand execution so, if necessary, you may be more fully able to contend with the condition. Some messages are simple or an informative nature or only act as a reassurance of continued editor operation.

For your convenience, a complete list of responses and informative messages is listed in Appendix B. For a more complete description of the messages, see VM/370 System Messages, GCnn-nnnn.

NOTES ON WRITING EDIT MACROS

You may write your own EDIT macros. A good knowledge of the CMS EXEC language is required. It is your responsibility to insure that any EDIT macro that you write checks the validity of its parameters and types an error message if necessary.

Parameters of a macro request must be separated from the macro name, and from each other, by at least one blank. Single and double quote marks cannot be entered as parameters, since they have a special meaning to the EXEC interpreter. Parameters passed to an EDIT macro are subject to the same rules as any other EXEC file (that is, the length of a parameter must be less than nine bytes).

If an EDIT macro is issued, and the EDIT macro file does not exist, the editor issues the message ?EDIT... . If it is used incorrectly, the editor types a message, and the macro is ignored. If an EDIT macro request is assigned to X or Y, it is an error to issued that X or Y request with a numerical parameter other than 0 or 1.

Some EDIT macros need to monopolize the CMS stacking facilities during their execution. If stacked lines exist when one of these macros is invoked, the stacked lines are deleted and the editor issues the message STACKED LINES CLEARED BY macro name. For example, this occurs if the CMS linend character has been used to stack additional requests after the macro is issued. Do not interrupt the execution of an EDIT macro via attention key.

```
ERMACRO  00117
ERMACRO  00118
ERMACRO  00119
ERMACRO  00120
ERMACRO  00122

ERMACRO  00123
ERMACRO  00124
ERMACRO  00125
ERMACRO  00126
ERSBCOM1 00001
ERSBCOM1 00002
```

An EDIT macro that uses FIFO stacking should normally ensure that the stack is initially clear. If the operation of an EDIT macro is completed with an error, the editor clears any stack lines and issues the message STACKED LINES CLEARED. Thus, the macro has no effect on the editor or its contents.

Generally, EDIT macros should operate with VERIFY OFF in order to avoid excessive typing by the editor. This can be accomplished without losing your setting by stacking PRESERVE and VERIFY OFF for execution first, and RESTORE for execution last.

ERSBCOM1 00005 EDIT SUBCOMMANDS AND MACROS

ERSBCOM1 00018 ## AGAIN SUBCOMMAND

ERSBCOM1 00019 The AGAIN subcommand allows you to stack (LIFO) the last
ERSBCOM1 00020 EDIT subcommand except AGAIN or a question mark and then
ERSBCOM1 00021 execute stacked subcommands.

Subcommand	Parameter
AGAIN	[subcommand]

subcommand
Specify any EDIT subcommand.

If an invalid EDIT subcommand is specified and it is not executed, the stacked subcommand is deleted. Thus, the invalid subcommand has no overall effect except to type an error message.

AGAIN can also be used to repeat a valid EDIT subcommand or, in some cases, to correct an invalid one. For instance, consider a user, without verification, mistakenly thinks he is INPUT mode:

```
The self-abnegation must be complete.  From
¬EDIT:THE SELF
again i
the point of view of Guatama, the dread of
death, that greed for an endless continuation
...
```

ERSBCOM1 00058 ## ALTER SUBCOMMAND

ERSBCOM1 00060 The ALTER subcommand allows you to search all or any part of
ERSBCOM1 00061 a file for a character or a byte of data to alter that
ERSBCOM1 00062 character with another character or byte of data.

Subcommand	Parameters
{ALTER|AL}	{option1} {option2} [1|n|* [1|G|*]]

option1
Specify either the character or two contiguous hexadecimal digits (0-9 or A-F) that the editor is to search the file for and subsequently alter the file at that location to the second option.

option2
Specify either the character or two contiguous hexadecimal digits (0-9 or A-F) that the editor is to alter the file to.

n or *
Starting at the current line, n indicates the number of lines to be searched. If * is entered, the search is performed until the EOF condition occurs. If this option is omitted then 1 line is searched. If The EOF condition exists when the ALTER subcommand is entered, scanning starts at the top of the file.

G or *
Indicates, during the scan, to alter every location in the lines specified that contain the data indicated by the first option. If omitted only the first occurrence in each line specified is altered. Note: The default, 1, cannot be specified.

With verification, each line is printed at the terminal as it is altered.

BOTTOM SUBCOMMAND

You can use the BOTTOM subcommand to make the last line of the file the current line without encountering the EOF condition. With verification, the line is printed at the terminal. The BOTTOM subcommand does not have any parameters.

```
┌──────────────┐
│  Subcommand  │
├──────────────┤
│  {BOTTOM|B}  │
└──────────────┘
```

CASE SUBCOMMAND

With the CASE subcommand you can indicated how the editor is to process or inquire how the editor is processing uppercase and lowercase letters.

Subcommand	Parameters
CASE	[M|U]

M

M indicates that the editor is to accept both uppercase and lowercase letters for the file as they are entered at the terminal.

U

U indicates that the editor is to translate all lowercase letters to uppercase letters before the letters are entered into the file.

The editor assumes U for all filetypes except MEMO and SCRIPT. If a parameter is not specified for CASE, the present setting is printed at the terminal. If a logical line follows the CASE subcommand in the same physical line (separated by the CMS linend character), translation of the logical line is not affected.

CHANGE SUBCOMMAND

When you issue the CHANGE subcommand, the editor searches all or any part of a file for a group of characters and changes that group of characters to another group of characters.

Subcommand	Parameters
{CHANGE|C}	/{string1}/{string2}[/ [1|n|* [1|G|*]]]

/

Specify any unique delimiting character that does not appear in the string.

string1

Specify the group of characters to be changed.

string2

Specify the group of characters to be inserted.

n or *

Starting at the current line n indicates the number of lines to be searched. If * is entered, the search is performed until the EOF condition occurs. If this option is omitted, then one line is searched. If the EOF condition exists when

the CHANGE command is entered, scanning starts at the top of the file.

n or *
Indicates, during the scan, to change every occurrence of string1. If omitted, only the first occurrence in each line specified is changed. Note: The default, 1, cannot be specified.

With verification, each line is printed at the terminal.

CMS SUBCOMMAND

The CMS subcommand causes the editor to enter the CMS Subset mode and allows you to execute those CMS commands that do not require the use of the main storage being used by the editor.

Subcommand
CMS

The CMS commands that you can execute are: RENAME, DISK, ERASE, EXEC, QUERY, LISTFILE, PRINT, PUNCH, READCARD, TYPE, ACCESS, STATE, and SET. Any attempt to execute an invalid CMS command or one which requires main storage provokes the response ?CMS SUBSET: ... . The commands LOAD, INCLUDE (RESET), START, and RUN should not be executed. To resume editing, enter the CMS command RETURN.

DELETE SUBCOMMAND

The DELETE subcommand allows you to delete all or any part of a file. The new current line is the one following the deleted lines.

Subcommand	Parameters
{DELETE|DEL}	[1|n|*]

n or *
Starting at the current line, n indicates the number of lines to be deleted. If * is entered, the remainder of the file is deleted as * indicates EOF. If omitted, one line is

deleted.

DOWN SUBCOMMAND

The DOWN subcommand allows you to advance the pointer down or forward into the file. The line pointed to becomes the new current line.

Subcommand	Parameters
{DOWN|D}	[1|n]

n
Starting at the current line, n indicates the number of lines to advance the pointer. If omitted, the pointer is advanced one line.

With verification the line pointed to is typed on your terminal or if the DOWN subcommand encounters EOF, the message EOF: is typed.

FILE SUBCOMMAND

The FILE subcommand allows you to modify the file identification specification originally supplied in the initial EDIT parameter list and/or write the file onto disk.

Subcommand	Parameters
FILE	[filename [filetype [filemode]]]

filename
Indicates the filename for the file. The filename can also be changed by the FNAME subcommand.

filetype
Indicates the filetype for the file.

filemode
Indicates filemode for the file. The filemode can also be changed by the FMODE subcommand.

Any existing file that has an identical fileid is replaced, and the operation of the editor is terminated. If errors

are encountered in performing the operation, appropriate action messages are typed and control returns to EDIT mode to allow recovery attempts. See Appendix B. Editor Messages.

FIND SUBCOMMAND

The FIND subcommand allows you to search all or a part of a file for a line of data.

Subcommand	Parameter
{FIND|F}	[line]

line
Indicate any valid input line. It may contain blanks and tabs.

The file is searched, from the next line, examining only those characters in each line which correspond in position to the non-blank characters in 'line'. The first line in which a match occurs becomes the new current line. If none is found, the message NOT FOUND is typed and the EOF condition is raised. If 'line' is null or blank, the search is successful on the first line examined. If the EOF condition exists when the FIND subcommand is issued, the search starts from the top of the file.

With verification, the line is displayed at your terminal when it is found.

FMODE SUBCOMMAND

The FMODE subcommand allows you to reset the filemode for subsequent FILE and SAVE subcommands, or display the current filemode setting.

Subcommand	Parameter
{FMODE|FM}	[filemode]

filemode
Indicates the filemode that is to replace the current filemode setting. If only a mode letter is given, the

existing mode number is retained (or if the previous mode was 'ST', the mode number is made '1' and the new mode is typed). If the filemode is not specified, then the defaults are:

For Existing Files: The filemode is not changed.

For New Files: The filemode in EDIT parameter list is displayed. If a filemode does not exist then A1 is assumed.

If a subsequent FILE or SAVE is issued and the editor determines that the filemode prevents writing, a message is typed to tell you to change the filemode.

FNAME SUBCOMMAND

The FNAME subcommand allows you to reset the filename for subsequent unqualified FILE and SAVE subcommands or display the current filename.

Subcommand	Parameter
{FNAME|FN}	[filename]

filename
Indicates the filename that is to replace the current filename. If the filename is not specified, then the filename from the EDIT parameter list is displayed.

GETFILE SUBCOMMAND

The GETFILE subcommand allows you to insert all or a part of a specific file into the file that you are processing.

Subcommand	Parameters
{GETFILE|G}	[{filename} [filetype [*|filemode]] [1|n [*|n]]

filename
Indicates the filename of the file that contains the data to be inserted into the file you are processing.

filetype
Indicates the filetype of the file that contains the data to

be inserted. If filetype is not specified, the filetype in the EDIT parameter list for the file you are processing is assumed.

filemode or *
Indicates the filemode of the file that contains the data to be inserted. If filemode is not specified, then * is assumed and all disks are searched until the file is found.

n
n indicates the first line of the file that is to be inserted into the file being edited. If n is omitted, then the first line of the file is assumed.

n or *
Starting with the line specified by n, n indicates the number of lines to be inserted. If a number is not specified or * is specified, then all or that part of the file between n and EOF is inserted.

With verification, if EOF of the the file containing the data is encountered, the message EOF REACHED is typed. The last line inserted becomes the new current line. If the record length of the file containing the data to be inserted exceeds that of the file being edited, a message is typed, and the GETFILE is not executed; if shorter, the records are padded to the record length of the file being edited.

IMAGE SUBCOMMAND

The IMAGE subcommand allows you to control the building of a 'line image' for input entered in the form of line, turn on canonical ordering, or display the current IMAGE setting.

Subcommand	Parameters
IMAGE	[ON|OFF|CANON]

ON
If ON is specified, text entered in the form of lines, as in FIND, INSERT, OVERLAY, and REPLACE, is expanded into a 'line image' where backspaces are removed and tabs are replaced by the appropriate number of blanks. The process that builds the line image simulates a typewriter having per line 135 columns for printing and the appropriate tab stops. If the input line contains backspaces, the column pointer can move forwards and backwards. The rules are as follows:

- Characters in the line image that are backspaced over are

not affected.

- Characters that are moved over forwards are replaced.

- If an attempt is made to put a non-blank character beyond column 135 or if a non-blank character is left in the line beyond the column of truncation for the file, then truncation occurs.

Text entered in the form of delimited strings, as in CHANGE, LOCATE, and (effectively) ALTER, is not expanded. Thus, tabs and backspaces are treated in the same way as other characters.

Except for SCRIPT files, ON is assumed for all other filetypes.

OFF
If OFF is specified, then tabs and backspaces are treated in the same way as other characters. Thus, they enter the file without being expanded or reordered.

In the same line or input, when a line of text follows a subcommand, the text starts at the character that terminates the subcommand if the terminator is not a blank, or the text starts at the next character if it is a blank.

CANON
Backspaces can be used to produce compound characters as in underlining. Before they are inserted in the file, compound characters are put into a canonical form which is independent of the order in which their components were given. (Backspaces are arranged singly between the characters which overlay each other, and the overlaying characters are arranged according to their EBCDIC values.) Tab characters do not receive special treatment, and enter the file in the same way as ordinary printing characters.

CANON is assumed for SCRIPT files.

INPUT SUBCOMMAND

The INPUT subcommand allows you to create new files or insert new data into an existing file.

Subcommand	Parameter
[INPUT|I]	[line]

ERSBCOM2 00333 Line
ERSBCOM2 00334 Specify the exact input line to be entered into the file.
ERSBCOM2 00335 It can contain blanks and tabs.

ERSBCOM2 00337 If a line is specified, it is put into the file following
ERSBCOM2 00338 the current line and subsequently becomes the current line.
ERSBCOM2 00340 This option is invalid if LINEMODE is LEFT or RIGHT.

ERSBCOM2 00341 If line is omitted, the editor enters INPUT mode, receives
ERSBCOM2 00342 lines from the terminal, and puts them into the file
ERSBCOM2 00343 following the current line. As each line is entered, it
ERSBCOM2 00344 becomes the new current line. EDIT mode is restored by
ERSBCOM2 00345 entering a null line. (A blank line is inserted by typing
ERSBCOM2 00346 at least one blank.)

ERSBCOM2 00347 If LINEMODE is LEFT or RIGHT, the editor prompts you with
ERSBCOM2 00348 line numbers. The line numbers are usually succesive
ERSBCOM2 00349 multiples of the prompting increment. However, if the
ERSBCOM2 00350 number so generated for prompting is larger than the next
ERSBCOM2 00351 line number in the file, the editor selects a number between
ERSBCOM2 00353 the current line number and the next line. If a prompting
ERSBCOM2 00354 cannot be generated, the current line and the next line
ERSBCOM2 00355 differ only be one, then EDIT mode is restored.

ERSBCOM2 00358 ## LINEMODE SUBCOMMAND

ERSBCOM2 00360 The LINEMODE subcommand allows you to choose if the editor
ERSBCOM2 00361 is to create a file in which every line is numbered in
ERSBCOM2 00361 ascending order and where the line number is placed in the
ERSBCOM2 00363 record or to display the current LINEMODE setting.

ERSBCOM2 00365
ERSBCOM2 00366 | Subcommand | Parameters |
ERSBCOM2 00367 |---|---|
ERSBCOM2 00368 | {LINEMODE\|LINE} | [[LEFT\|L]\|[RIGHT\|R]\|OFF] |
ERSBCOM2 00369

ERSBCOM2 00373 LEFT
ERSBCOM2 00374 If LEFT is specified, columns 1-15 are set aside for line
ERSBCOM2 00375 numbers. If the file was previously created, the editor
ERSBCOM2 00376 assumes that all records are numbered in ascending order in
ERSBCOM2 00377 columns 1-15 and that all numbers are right justified with
ERSBCOM2 00378 leading blanks. The near zone and the first column in which
ERSBCOM2 00379 text is entered are set to 7 so the ALTER, CHANGE, FIND,
ERSBCOM2 00379 LOCATE, and OVERLAY subcommands do not process the line
ERSBCOM2 00380 numbers.

ERSBCOM2 00382 RIGHT
ERSBCOM2 00383 If RIGHT is specified, columns 76-80 are set aside for line
ERSBCOM2 00384 numbers. If the file was previously created, the editor
ERSBCOM2 00385 assumes that all records are numbered in ascending order in

columns 76-80 and that all numbers are right justified with leading zeros. RIGHT is only valid for files with fixed length 80-byte records.

Lines entered at the terminal are reformatted so that the line number followed by a blank appears to the left of the line being typed. On entering LINEMODE RIGHT, the verification column is reset to 72 to prevent the line number in columns 76-80 from printing a second time. In INPUT mode you are prompted with a 5-digit number followed by a blank (as when lines are typed). This number is placed into the file in columns 76-80. Its appearance on the left is for your convenience and does not indicate its true position.

If a SAVE or FILE command is issued, reserialization in columns 76-80 is suppressed to allow the line numbers that are saved on disk to have the same values they had during the editing session. If LINEMODE is set OFF before issuing a SAVE, reserialization takes place according to the SERIAL options currently in effect. This allows you to renumber all your lines in the middle of an editing session.

OFF
No line numbering is assumed. There is no prompting in INPUT mode. The nnnnnn command with an option is disabled.

Note: If an option is not specified with LINEMODE, the current setting is displayed.

LOCATE SUBCOMMAND

The LOCATE subcommand allows you to scan the file, from the next line, for the first occurrence of a group of characters.

Subcommand	Parameter
{LOCATE|L}	[/string[/]]

/
Specify any unique delimiting character that does not appear in the string. / may be any non-blank character. The closing delimiter is optional.

string
Specify any group of characters to be searched for in the file.

If it is found, the line containing it becomes the new

current line; otherwise the message NOT FOUND is typed, and the EOF condition is raised.

If the opening delimiter is /, the word LOCATE or an abbreviation for it need not be entered either. If the string is null, the search is successful on the first line encountered. If the EOF condition exists when LOCATE is issued, scanning starts from the top of the file. With verification, the line located is typed on the user's terminal.

LONG SUBCOMMAND

The LONG subcommand instructs the editor to respond ?EDIT when an invalid EDIT subcommand or macro is entered. This is the default mode for the editor.

Subcommand
LONG

NEXT SUBCOMMAND

The NEXT subcommand allows you to point to the next line, or down a specified number of lines. The line pointed to becomes the new current line.

Subcommand	Parameter
{NEXT|N}	[1|n]

n
Indicates the number of lines to advance the pointer. If n is omitted then one line is assumed.

NEXT operates identically to DOWN command. With verification, the line pointed to is displayed on your terminal. Also, if NEXT causes the the number of lines in the file to be exceeded, EOF: is typed.

OVERLAY SUBCOMMAND

The OVERLAY subcommand allows you to overlay the current line with the non-blank characters in the line specified.

Subcommand	Parameter
{OVERLAY|O}	[line]

line
Specify an input line that replaces parts of the current line. An underscore in the line specification forces a blank into corresponding position of the current line.

It is convenient to enter this request by typing an 'O' followed by a backspace followed by the overlaying material. This set up the correct alignment on the terminal.

With verification, the line is displayed on the terminal after it has been overlaid.

PRESERVE SUBCOMMAND

The PRESERVE allows you to save the current settings of CASE, IMAGE, LINEMODE, LONG, MODE, NAME, RECFORM, SERIAL, SHORT, TABS, TRUNC, VERIFY and ZONE.

Subcommand
{PRESERVE|PRE}

PROMPT SUBCOMMAND

The PROMPT subcommand allows you to change the prompting increment that line numbers are incremented by during line number prompting. This is only relevant in INPUT mode with LINEMODE RIGHT or LINEMODE LEFT.

Subcommand	Parameter
PROMPT	[1|n]

n

Specify the prompting increment. If n is omitted, then 1 is assumed.

QUIT SUBCOMMAND

The QUIT subcommand allows you to terminate the editor and leave the previous copy of the file, if any, intact on the disk. If serious errors have been made during the session and you do not wish to preserve the contents of the file currently being edited, then issue the QUIT subcommand. If a SAVE subcommand was issued prior to QUIT and the file was unmodified, the file on disk contains the changes that occurred during the session.

Subcommand
QUIT

RECFORM SUBCOMMAND

The RECFORM subcommand allows you to set the record format for the file or display the current RECFORM setting.

Subcommand	Parameters
{RECFORM|REC}	{F|V}

F
F indicates fixed length records. F is assumed for all new files except LISTING, MEMO, and SCRIPT.

V
V indicates variable length records. V is assumed for all new LISTING, MEMO, and SCRIPT files. Usually, a V format file occupies the smaller disk space because trailing blanks are deleted from each line before it is written onto disk.

If V or F is not entered, then the current RECFORM setting is displayed.

REPEAT SUBCOMMAND

The REPEAT subcommand allows you to execute the immediately following OVERLAY subcommand or its equivalent for a specified number of lines, or to the end of the file.

Subcommand	Parameter
REPEAT	[1|n|*]

n or *

Starting at the current line, n indicates the number of times to repeat the OVERLAY subcommand or its equivalent that immediately follows. * indicates that the overlay is to be repeated until EOF. If n or * is not specified, then only one line is overlaid.

The last line overlaid becomes the new current line. If the next subcommand is not a valid OVERLAY request or an X or Y subcommand assigned to invoke OVERLAY, the count for REPEAT is reset to one.

REPLACE SUBCOMMAND

The REPLACE subcommand allows you to replace the current line with a specified line. If a line is not specified, then the editor enters INPUT mode.

Subcommand	Parameter
{REPLACE|R}	[line]

line

Specify an input line that is to replace the current line. If a line is specified, then the editor puts it into the file in place of the current line. If a line is not specified, then the editor enters INPUT mode, receives lines from the terminal, and puts them into the file inplace of the current line. As each line is entered, it becomes the new current line. EDIT mode is restored by entering a null line. If LINEMODE is LEFT or RIGHT, then REPLACE is not valid.

RESTORE SUBCOMMAND

The RESTORE subcommand allows you to restore the settings of CASE, IMAGE, LINEMODE, LONG, PROBE, PNAME, RECFORM, SERIAL, SHORT, TABS, TRUNC, VERIFY and WORD to the values they held prior to the last issued PRESERVE subcommand or to their initial values if a PRESERVE subcommand has not been issued.

Subcommand
{RESTORE|RES}

SAVE SUBCOMMAND

The SAVE subcommand allows you to save the file being edited on disk.

Subcommand	Parameters
SAVE	[filename [filetype [filemode]]]

filename
Indicates the filename of the file to be saved.

filetype
Indicates the filetype of the file to be saved.

filemode
Indicates the filemode of the file to be saved.

The SAVE subcommand uses the specified filename, filetype, and filemode, or the current values in the EDIT parameter list as defaults for any parameter not specified. Any file on disk that has the identical filename and filetype and filemode specification is replaced.

SERIAL SUBCOMMAND

The SERIAL subcommand allows you to control the serialization of records in columns 73-80.

Subcommand	Parameters
SERIAL	[seq|ON|ALL|OFF [10|n]]

```
ERSBCOM3 00305
ERSBCOM3 00310
ERSBCOM3 00311
ERSBCOM3 00312
ERSBCOM3 00313
ERSBCOM3 00315
ERSBCOM3 00317
ERSBCOM3 00319
ERSBCOM3 00321
ERSBCOM3 00323
ERSBCOM3 00324
ERSBCOM3 00325
ERSBCOM3 00327
ERSBCOM3 00328
ERSBCOM3 00329
ERSBCOM3 00331
ERSBCOM3 00332
ERSBCOM3 00333
ERSBCOM3 00334
ERSBCOM3 00335
ERSBCOM3 00336
ERSBCOM3 00337
ERSBCOM3 00338
ERSBCOM3 00339
ERSBCOM3 00340
ERSBCOM3 00341
ERSBCOM3 00342
ERSBCOM3 00343
ERSBCOM3 00344
ERSBCOM3 00345
ERSBCOM3 00346
ERSBCOM3 00347
ERSBCOM3 00348
ERSBCOM3 00349
ERSBCOM3 00350
ERSBCOM3 00352
ERSBCOM3 00355
ERSBCOM3 00356
ERSBCOM3 00358
ERSBCOM3 00359
```

seq
Specify the three-character identification to be used in columns 73-75.

ON
Indicates the first three characters of the filename are to be used in columns 73-75.

ALL
Indicates columns 73-80 are to be used for serialization.

OFF
Indicates serialization or an identifier is not to be placed in columns 73-80.

n
Specifies the increment for the line number in columns 76-80 (or 73-80). This number also becomes the first line number. If n is not specified, then 10 is assumed.

Initially, serialization is determined by the filetype specified in the EDIT command. The SERIAL subcommand is valid only if the file uses fixed length records with a record length of 80. It takes effect after a FILE or SAVE subcommand is issued.

Normal serialization consisists of a 3-character sequence-name in columns 73-75 followed by a 5-digit number in columns 76-80. This is obtained by issuing SERIAL with a sequence-name or by using the ON option (which means that the sequence name is to be taken from the first three characters of the filename of the file being edited.

An alternative form of serialization uses an eight-digit number in columns 73-80. This is obtained by issuing SERIAL ALL . Serialization can be turned off by issuing SERIAL OFF.

When serialization is indicated, the editor sets TRUNC to the minimum of 72; the existing setting is changed, the message TRUNC SET TO 72 is typed. The end zone is also set to 72; if the end zone is changed, the message END ZONE SET TO 72 is typed. The setting for VERIFY is not changed.

SHORT SUBCOMMAND

The SHORT subcommand allows you to change the message response mode. Invalid EDIT subcommands invoke the response .

```
┌─────────────┐
│ Subcommand  │
├─────────────┤
│ SHORT       │
└─────────────┘
```

STACK SUBCOMMAND

The STACK subcommand allows you to stack lines or EDIT subcommands in the terminal input buffer for subsequent processing.

```
┌──────────────────────────────────────┐
│ Subcommand │ Parameters              │
├────────────┼─────────────────────────┤
│ STACK      │ [1|n|subcommand]        │
└──────────────────────────────────────┘
```

n
Indicate the number of lines to be stacked. If a number or a subcommand is not specified, then one line is assumed. Stack n lines (FIFO), starting with the current line on the terminal input buffer. The last line stacked becomes the new current line. The length of the lines is taken from the column set by TRUNC. If STACK is issued with an argument of 0 (zero), a null line is stacked. A maximum of 500 lines can be stacked.

Specify an EDIT subcommand to be stacked. If a subcommand is specified, it is stacked FIFO. STACK enables subcommands to be issued from the file, and also makes it possible to move or copy lines.

TABSET COMMAND

The TABSET subcommand allows you to set the logical tabs for the file.

```
┌──────────────────────────────────────┐
│ Subcommand     │ Parameters          │
├────────────────┼─────────────────────┤
│ {TABSET|TABS}  │ {n  n ...n }        │
│                │   1  2     x        │
└──────────────────────────────────────┘
```

n
Indicates column positions for logical tab settings. A

maximum of 25 is allowed. The first tab entry indicates the column that the column pointer identifies after the first depression of the tab key after a carriage return. The default settings are:

Filetypes	Tabs
ASSEMBLE, UPDATE	1, 10, 16, 31, 36, 41, 46, 69, 72, 81
FORTRAN, COBOL	1, 7, 10, 15, 20, 25, 30, 81
BASIC	7, 10, 15, 20, 25, 30, 81
PLI	1, 4, 7, 10, 13, 16, 19, 22, 25, 31, 37, 43, 49, 55, 79, 81
Others	1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 61, 71, 81, 91, 101, 111, 121, 131

Tab settings have no effect if canonical ordering is in effect, or if IMAGE is OFF.

TOP SUBCOMMAND

The TOP subcommand allows you to point to the top of the file. The null top line becomes the current line.

Subcommand
TOP

TRUNC SUBCOMMAND

The TRUNC subcommand allows you to truncate records or display the current setting of TRUNC.

Subcommand	Parameters
TRUNC	[n|*]

n
Indicates the column at which truncation is to occur.

*

Indicates truncation is to be set to the record-length.

Set the column of truncation to n, or to the record length. If no argument is specified, the current truncation column is displayed. The defaults are:

Filetypes	Truncation Column
ASSEMBLE, UPDATE	Minimum of 71 and record length
FORTRAN, COBOL, BASIC	Minimum of 72 and record length
Others	Record length

TYPE SUBCOMMAND

The TYPE subcommand allows you to display at the terminal all or any part of a file.

Subcommand	Parameters
{TYPE|T}	[1|m|* [n|*]]

m or *

m indicates the number of lines to be displayed. * indicates all lines beyween the current line and EOF. If m or * is omitted, then one line is typed.

n

Indicates the column at which typing stops.

***

Indicates that typing is taking place for the full record length.

Display m lines or to EOF, starting with the current line. The last line displayed becomes the new current line. The maximum length of the displayed lines is determined by the column set by VERIFY. To display the entire line, specify * or n to indicate the last column to be typed. If LINEMODE RIGHT has been specified, the line numbers stored in columns 76-80 are typed on the left.

UP SUBCOMMAND

The UP subcommand allows you to position the line pointer to a line with a lower line number than the one that you are processing.

Subcommand	Parameters
{UP|U}	[1|n]

n
Indicates the number of lines the pointer is to be moved back. If a number is not specified, then one line is assumed. The line pointed to becomes the new current line.

With verification, the line pointed to is displayed at the user's terminal.

VERIFY SUBCOMMAND

The VERIFY subcommand allows you to display all or any part of a line at the terminal after it is processed or to display the current VERIFY setting.

Subcommand	Parameter
{VERIFY|V}	[ON|OFF [n|*]]

ON
Allow verification. Lines which are located and changed are automatically typed out, and changes between EDIT and INPUT mode are indicated.

OFF
Prevents verification. Lines which are located and changed are not displayed, and changes between EDIT and INPUT mode are not indicated.

n
Indicates the number of columns to verify in each line.

*
Indicates verification of a line to its record length.

If n is specified, the column to which verification and printing is to go; if * is given, it is set to the limit (i.e. the item length). Types out current verify column if no arguments are given. The defaults for this column are:

ERSBCOM4 00149	Filetypes	Column of Verification
ERSBCOM4 00151 ERSBCOM4 00152	ASSEMBLE, UPDATE, FORTRAN, COBOL,	Either of 72 and record length
ERSBCOM4 00154	Others	Record length

ERSBCOM4 00158 **Note:** If no parameters are entered, the current VERIFY
ERSBCOM4 00159 settings are displayed.

ERSBCOM4 00162 ## X or Y SUBCOMMAND

ERSBCOM4 00164 The X or Y subcommands allow you to assign to X or Y a given
ERSBCOM4 00165 EDIT subcommand, or execute the previously assigned
ERSBCOM4 00166 subcommand a specified number of times.

ERSBCOM4 00168

Subcommand	Parameters
{X|Y}	[1|n|subcommand]

ERSBCOM4 00171
ERSBCOM4 00172

ERSBCOM4 00176 n
ERSBCOM4 00177 Indicates the number of times the previously assigned
ERSBCOM4 00179 subcommand is to be executed.

ERSBCOM4 00181 subcommand
ERSBCOM4 00183 Indicates any subcommand.

ERSBCOM4 00185 If n is specified and is greater than 1, the current line is
ERSBCOM4 00186 not necessarily advanced between executions. Advancement
ERSBCOM4 00187 depends upon the dit request that has been assigned to X or
ERSBCOM4 00188 Y. Execution stops at EOF. If a number or a request is not
ERSBCOM4 00189 specified, the previously assigned subcommand is executed
ERSBCOM4 00189 once.

ERSBCOM4 00191 X and Y are initially set to the null string.

ERSBCOM4 00194 ## ZONE SUBCOMMAND

ERSBCOM4 00195 The ZONE subcommand allows you to tell the editor the
ERSBCOM4 00196 portion of each record, starting position and ending
ERSBCOM4 00197 position, to be scanned when scanning occurs or to display
ERSBCOM4 00198 the current ZONE settings.

ERSBCOM4 00200

Subcommand	Parameters

ERSBCOM4 00201
ERSBCOM4 00202

```
| {ZONE|Z}    | [1|e|* [s|*]] |
└─────────────────────────────┘
```

e
Indicates the first column of the zone of each record to be scanned. The default value is column 1.

s
Indicates the last column of the zone of each record to be scanned.

*
Invokes the defaults.

The ZONE settings are used by the ALTER, CHANGE, and LOCATE subcommands. If no parameter is entered, the current settings are displayed. The defaults for the end zone are:

Filetypes	End Zone
ASSEMBLE, UPDATE	Minimum of 71 and record length
FORTRAN, COBOL, BASIC	Minimum of 72 and record length
Others	Record length

? SUBCOMMAND

The ? subcommand allows you to display the last EDIT subcommand except AGAIN or a question mark. The ? cannot be entered in the last EDIT subcommand to be nulled.

Subcommand
?

After an X or Y subcommand, the last edit request is the request that was executed as a result of issuing the X or Y subcommand.

nnnnn SUBCOMMAND

The nnnnn subcommand allows you to enter lines into the file or to search the file for that line.

Subcommand	Parameter

ERSBCOM4 00264 | nnnnn | [text] |
ERSBCOM4 00265

ERSBCOM4 00269 nnnnn
ERSBCOM4 00271 indicates a line number between 0 and 99999.

ERSBCOM4 00273 text
ERSBCOM4 00275 Enter a line of text to be put into the file.

ERSBCOM4 00277 A line number not followed by any text is interpreted to be
ERSBCOM4 00278 a request to locate the line with that number. If such a
ERSBCOM4 00279 line is found it is displayed (unless VERIFY is OFF). If
ERSBCOM4 00280 the line is not found, the line pointer is reset to the
ERSBCOM4 00280 previous line and LINE NOT FOUND is typed. If LINEMODE is
ERSBCOM4 00282 LEFT the line number is assumed to be in columns 1-5 right
ERSBCOM4 00283 justified with leading blanks. If LINEMODE is RIGHT or OFF
ERSBCOM4 00284 a line number is assumed to be in columns 76-80, right
ERSBCOM4 00285 justified with leading zeros.

ERSBCOM4 00287 A line number followed by text is inserted into the file or
ERSBCOM4 00288 replaces a line in the file.

ERSBCOM4 00290 With LINEMODE LEFT the line number is padded with blanks on
ERSBCOM4 00291 the left (if necessary) and inserted in columns 1-5. The
ERSBCOM4 00292 text is placed on the same line at the first tab position
ERSBCOM4 00293 (normally column 7). The entire reformatted line is placed
ERSBCOM4 00295 in the file in the location appropriate to its line number.
ERSBCOM4 00296 If a line with that number already exists, it is replaced.
ERSBCOM4 00297 Line numbers with leading zeros are not accepted.

ERSBCOM4 00299 With LINEMODE RIGHT the line number is padded with zeros on
ERSBCOM4 00300 the left (if necessary) and inserted in columns 76-80. The
ERSBCOM4 00301 text is placed on the same line at the first tab position
ERSBCOM4 00302 (normally column 1). The entire reformatted line is placed
ERSBCOM4 00304 in the file at the location appropriate to its line number.
ERSBCOM4 00305 If a line with that number already exists, it is replaced.

ERSBCOM4 00308 With LINEMODE OFF the text option is invalid. The search
ERSBCOM4 00308 option operates as in LINEMODE RIGHT on files which have
ERSBCOM4 00309 line numbers in columns 76-80.

ERMACROS 00015 EDIT MACROS

ERMACROS 00016 An edit macro defines a sequence of EDIT subcommands which
ERMACROS 00017 can be executed by issuing a single macro which uses CMS
ERMACROS 00019 EXEC files, invoked from the editor, to stack requests.

ERMACROS 00021 The following conventions have been established:

ERMACROS 00024 1. EXEC files which are concerned with edit macros have a
ERMACROS 00025 filename which starts with a ¬. These files are

referred to as 'edit macro' files.

2. An EDIT macro request consists of the name of an EDIT macro file (omitting the initial 7), followed possibly by parameters.

EDIT macros are EXEC files which allow you to create complex EDIT subcommands.

Macro	Parameters
&DUP	[[n]

Duplicate the current line n times. The last copy of the line becomes the new current line. If n is omitted, the line is duplicated once.

Macro	Parameters
&MOVE	n {[[UP|DOWN] m]|[TO label]}

Move n lines (starting with the current line) up or down m lines; or move n lines (starting with the current line) and insert them after 'label', or at EOF if 'label' is not found. The new copy of the last line moved becomes the new current line. The label should start in column 1, and should not contain any lower-case letters. Abbreviations may be used for 'UP', 'DOWN' and 'TO'.

There is a limit of 500 lines which can be moved by one request.

APPENDIX A. SUMMARY OF EDIT COMMANDS

Subcommand	Parameters
AGAIN	{subcommand}
{ALTER|AL}	{option1} {option2} [1|n|* [1G|*]]
{BOTTOM|B}	
CASE	[M|U]
{CHANGE|C}	/{string1}/{string2}[/ [1|n|* [1|G|*]]]
CMS	
{DELETE|DEL}	[1|n|*]
{DOWN|D}	[1|n]
FILE	[filename [filetype [filemode]]]
{FIND|F}	[line]
{FMODE|FM}	[filemode]
{FNAME|FN}	[filename]
{GETFILE|G}	{filename} [filetype [*|filemode]] [1|n [*|n]]
IMAGE	[ON|OFF|CANCN]
{INPUT|I}	[line]
{LINEMODE|LINE}	[[LEFT|L[RIGHT|R]|OFF]
{LOCATE|L}	[/string[/]]
LONG	
{NEXT|N}	[1|n]
{OVERLAY|O}	{line}

```
ERAPPA  00062
ERAPPA  00063
ERAPPA  00064
ERAPPA  00065
ERAPPA  00066
ERAPPA  00067
ERAPPA  00068
ERAPPA  00069
ERAPPA  00070
ERAPPA  00071
ERAPPA  00072
ERAPPA  00073
ERAPPA  00074
ERAPPA  00075
ERAPPA  00076
ERAPPA  00077
ERAPPA  00078
ERAPPA  00079
ERAPPA  00080
ERAPPA  00081
ERAPPA  00082
ERAPPA  00083
ERAPPA  00084
ERAPPA  00085
ERAPPA  00086
ERAPPA  00087
ERAPPA  00088
ERAPPA  00089
ERAPPA  00090
ERAPPA  00091
ERAPPA  00092
ERAPPA  00093
ERAPPA  00094
ERAPPA  00095
ERAPPA  00096
ERAPPA  00097
ERAPPA  00098
ERAPPA  00099
ERAPPA  00100
ERAPPA  00101
ERAPPA  00103
ERAPPA  00104
ERAPPA  00105
ERAPPA  00106
```

Subcommand	Parameters
{PRESERVE|PRE}	
QUIT	
{RECFORM|REC}	[F|V]
REPEAT	[1|n|*]
{REPLACE|R}	[line]
{RESTORE|RES}	
SAVE	[filename [filetype [filemode]]]
SERIAL	{seq|ON|ALL|OFF [10|n]}
SHORT	
STACK	[1|n|subcommand]
{TABSET|TABS}	{n1 n2 ...nx}
TOP	
TRUNC	[n|*]
{TYPE|T}	[1|m|* [n|*]]
{UP|U}	[1|n]
{VERIFY|V}	[ON|OFF [n|*]]
{X|Y}	[1|n|subcommand]
{ZONE|Z}	[1|p|* [q|*]]
?	
nnnnn	[text]

ERAPPB 00008 APPENDIX B. EDITOR MESSAGES

ERAPPB 00012 AVAILABLE STORAGE IS NOT FULL
ERAPPB 00013 The size of the file cannot be increased. Any attempt to
ERAPPB 00014 add lines produces the message NO ROOM. Other commands are
ERAPPB 00015 unaffected.

ERAPPB 00017 CMS SUBSET
ERAPPB 00019 The CMS subcommand has been issued.

ERAPPB 00021 EDIT
ERAPPB 00022 Indicates entry to EDIT mode. During initialization, if the
ERAPPB 00023 fileid specified in the EDIT parameter list is encountered
ERAPPB 00024 on secondary storage, this is the first response, otherwise
ERAPPB 00025 the message NEW FILE follows. Also, during the session,
ERAPPB 00026 this message indicates a return from INPUT mode, or
ERAPPB 00027 indicates a null line while in EDIT mode, subject to the
ERAPPB 00028 setting of the VERIFY switch.

ERAPPB 00030 END ZONE SET TO 72
ERAPPB 00031 The SERIAL subcommand was issued and the second zone
ERAPPB 00032 specification was set within the serialization field. The
ERAPPB 00033 end zone is reset to column 72.

ERAPPB 00035 EOF:
ERAPPB 00036 The current line pointer is positioned after the bottom line
ERAPPB 00037 of the file or if the file is empty, after the null line at
ERAPPB 00039 the top of the file. Subject to VERIFY switch.

ERAPPB 00041 EOF REACHED
ERAPPB 00042 The number of lines beyond the starting line specified in a
ERAPPB 00043 GETFILE request exceeded EOF in the indicated file. The
ERAPPB 00044 lines from starting line to EOF were inserted in the file.
ERAPPB 00045 With verification, the last line inserted is displayed at
ERAPPB 00046 the terminal.

ERAPPB 00048 FILE IS EMPTY
ERAPPB 00049 An attempt to FILE or SAVE a null file was detected. If the
ERAPPB 00050 subcommand was FILE, the editor exits; if it was SAVE,
ERAPPB 00051 there is a return to the EDIT mode. In neither case, the
ERAPPB 00052 file is not written.

ERAPPB 00054 FILE NOT FOUND
ERAPPB 00055 The fileid specified in a GETFILE was not found on secondary
ERAPPB 00056 storage.

ERAPPB 00058 GETFILE IS INCOMPLETE
ERAPPB 00059 The available storage was exceed while attempting to execute
ERAPPB 00060 a GETFILE. The last line inserted into the file is displayed
ERAPPB 00061 at the terminal.

```
ERAPPB   00064          GIVEN STARTING LINE IS BEYOND EOF
ERAPPB   00065          The starting line specified in a GETFILE points beyond the
ERAPPB   00067          last line of the indicated file.

ERAPPB   00070          ILLEGAL CHARACTER IN LINE NUMBER COLUMNS
ERAPPB   00071          A non-numeric character was found in the columns reserved
ERAPPB   00072          for line numbers. The line pointer identifies the line in
ERAPPB   00073          error.

ERAPPB   00075          INPUT
ERAPPB   00076          Indicates entry to INPUT mode wherein lines entered at the
ERAPPB   00078          terminal become part of the file. Subject to VERIFY switch.

ERAPPB   00081          MAXIMUM LINE NUMBER EXCEEDED
ERAPPB   00083          The maximum line number is 99999.

ERAPPB   00085          NEW FILE
ERAPPB   00086          During editor initialization, if the fileid specified in the
ERAPPB   00088          EDIT parameter list does not exist on secondary storage.

ERAPPB   00090          NO ROOM
ERAPPB   00091          An attempt to input additional lines to a file has been
ERAPPB   00092          detected after the full storage message was typed. Any
ERAPPB   00093          stacked lines are cleared to avoid multiple error messages
ERAPPB   00094          or improper subcommand execution sequences.

ERAPPB   00096          NOT FOUND
ERAPPB   00097          The search parameter specfied in the ALTER, CHANGE, FIND, or
ERAPPB   00098          LOCATE subcommand was not encountered in the delimited range
ERAPPB   00099          or before EOF.

ERAPPB   00102          READ ERROR - GETFILE IS INCOMPLETE
ERAPPB   00103          An error, external to the editor, was encountered during the
ERAPPB   00104          execution of a GETFILE. The last line inserted into the
ERAPPB   00105          file is displayed at the terminal.

ERAPPB   00108          RECORD LENGTH OF FILE TOO LARGE
ERAPPB   00109          The fileid specification of a GETFILE indicates a file with
ERAPPB   00111          a record length greater than the file being edited.

ERAPPB   00113          RENUMBER LINES
ERAPPB   00116          (a) The line number prompter cannot proceed because there
ERAPPB   00117              are no more numbers between the current line number and
ERAPPB   00118              the line number of the next line already in the file
ERAPPB   00119              (that is, they differ by one).

ERAPPB   00122          (b) The next line number, 100000, is too large.

ERAPPB   00126          RESERIALIZATION SUPPRESSED
ERAPPB   00127          Reserialization on a SAVE or FILE subcommand is suppressed
ERAPPB   00128          when LINEMODE is NIGHT so that the numbers used during the
ERAPPB   00129          editing session are retained. To reserialize, repeat the
ERAPPB   00130          SAVE or FILE with LINEMODE OFF.
```

```
ERAPPB  00133          SERIALIZATION IS INCOMPLETE
ERAPPB  00134  During the execution of a SAVE or FILE subcommand that is
ERAPPB  00135  serializing a file, the disk becomes full before the last
ERAPPB  00136  line is written.  The partial file is erased and the user is
ERAPPB  00137  notified of the condition.

ERAPPB  00141          SET NEW FILEMODE, OR ENTER CMS SUBSET AND CLEAR SOME SPACE
ERAPPB  00142  During the execution of a SAVE or FILE, the disk became full
ERAPPB  00143  prior to writing the last line.  The partial file was
ERAPPB  00144  erased.  The user can now institute recovery procedures
ERAPPB  00145  since the editor returns to EDIT mode.

ERAPPB  00148          SET NEW FILENAME AND REENTER THE REQUEST
ERAPPB  00149  During the execution of a SAVE or FILE, an error occurred
ERAPPB  00150  while altering the name of the work-file.  The user can now
ERAPPB  00151  institute recovery procedures since the editor returns to
ERAPPB  00152  EDIT mode.  The work-file remains.  It can be erased and a
ERAPPB  00153  different fileid for subsequent SAVE or FILE subcommands can
ERAPPB  00154  be specified.

ERAPPB  00156          STACKED LINES CLEARED
ERAPPB  00157  Multiple subcommands were detected after the failure to
ERAPPB  00158  increase the file size subsequent to the editor's typing NO
ERAPPB  00159  ROOM.  This message is also typed when an exit from the
ERAPPB  00160  EDIT mode occurs to preserve the CMS command environment
ERAPPB  00161  from stacked EDIT requests.

ERAPPB  00163          TOP:
ERAPPB  00164  The current line pointer is positioned at the null line at
ERAPPB  00165  the top of the file.  This message appears either after the
ERAPPB  00166  TOP command has been issued or after an UP command has
ERAPPB  00167  positioned the line pointer at the null line at the
ERAPPB  00168  beginning of the file.

ERAPPB  00170          TOO MANY LINES TO STACK
ERAPPB  00171  During initialization, the parameter of the STACK subcommand
ERAPPB  00172  implies storage requirement in excess of that reserved for
ERAPPB  00174  the execution of the subcommand.  Limit is 500 lines.

ERAPPB  00176          TRUNC SET TO 72
ERAPPB  00177  The SERIAL subcommand was issued and the truncation column
ERAPPB  00178  was set within the serialization field.  The truncation
ERAPPB  00179  column is reset to column 72.

ERAPPB  00181          TRUNCATED
ERAPPB  00182  The current line has exceeded the truncation column.  If
ERAPPB  00183  VERIFY switch is on, the truncated line is displayed,
ERAPPB  00184  followed by the message INPUT: if in INPUT mode.

ERAPPB  00187          WRONG FILE FORMAT FOR LINEMODE RIGHT
ERAPPB  00188  The LINEMODE RIGHT option is not compatible with V-format
ERAPPB  00190  files or files that have a record length other than 80.
```

ERAPPB	00193	WRONG FILE FORMAT FOR SERIALIZATION
?RAPPB	00194	The SERIAL subcommand was issued for a V-format file or for
?RAPPB	00195	a file that does not have a record length of 80.
ERAPPB	00196	
ERAPPB	00198	ZONE ERROR
?RAPPB	00199	The string1 is too long for the current ZONE specification.
?RAPPB	00201	The file is not changed.
ERAPPB	00203	?
?RAPPB	00204	Same as ?EDIT:... , but the input line is not displayed
?RAPPB	00206	because SHORT is in effect.
ERAPPB	00208	?$
?RAPPB	00209	Same as ?EDIT:... , but is typed with an invalid EDIT macro
?RAPPB	00211	is issued and SHORT is in effect.
ERAPPB	00213	?CMS SUBSET:...
?RAPPB	00214	An unknown command has been entered. The complete input
?RAPPB	00216	line is displayed for user inspection.
ERAPPB	00218	?EDIT:...
?RAPPB	00219	An unrecognizable EDIT subcommand or invalid subcommand
?RAPPB	00220	parameter list has been encountered. The input line is
?RAPPB	00221	displayed for inspection. This is the form if LONG is in
?RGLOSS	00001	effect.

GLOSSARY

For a more complete list of data processing terms, refer to Data Processing Glossary, GC20-1699.

character-deletion character: A character within a line of terminal input specifying that it and the immediately preceding character are to be removed from the line.

command scan: A service routine that checks the syntax of commands and subcommands.

context editing: In systems with time sharing, a method of editing a line data set of file without using line numbers. To refer to a particular line, all or part of the contents of that line are specified.

Conversational Monitor System (CMS): A virtual machine operating system that only operates under the control of VM/370.

current line pointer: (1) In systems with time sharing, a pointer that indicates the line of a line data set or file with which the user is currently working. (2) In VM/370, a pointer that indicates the record number of a CMS file with which the user is currently working.

edit mode: In systems with time sharing, an entry mode under the EDIT command that accepts successive subcommands suitable for modifying an existing line data set or file.

IBM Virtual Machine Facility/370 (VM/370): A time-sharing System Control Program that manages the resources of a System/370 computing system in such a way that multiple remote terminal users have a functional simulation of a computing system (a virtual machine) at their disposal.

input line: In VM/370, information typed by a user between the time the typing element of his terminal comes to rest following a carriage return until another carriage return is typed.

ERGLOSS 00047 line-deletion character: A terminal character that
ERGLOSS 00048 specifies that all characters are to be deleted from a line
ERGLOSS 00049 of terminal input.

ERGLOSS 00052 line number: A number associated with a line which can be
ERGLOSS 00053 used to refer to the line.

ERGLOSS 00054 line number editing: In systems with time sharing, a mode
ERGLOSS 00055 of operation under the EDIT command in which lines or
ERGLOSS 00056 records to be modified are referred to by line or record
ERGLOSS 00057 number.

ERGLOSS 00059 null line: In VS/370, a terminal input line consisting of a
ERGLOSS 00059 carriage return or line-end character issued as the first
ERGLOSS 00060 and only information after the typing element of the
ERGLOSS 00061 terminal has come to rest following a previous carriage
ERGLOSS 00062 return or line-end character. Usually used to terminate
ERGLOSS 00064 input mode and enter EDIT mode.

ERGLOSS 00065 password: In systems with time sharing, a
ERGLOSS 00066 one-to-eight-character symbol that the user may be required
ERGLOSS 00067 to supply at the time he logs on the system. The password is
ERGLOSS 00068 confidential as opposed to user identification. Users can
ERGLOSS 00070 also assign passwords to data sets or files.

ERGLOSS 00071 prompting: A function that helps a terminal user by
ERGLOSS 00072 requesting him to supply operands necessary to continue
ERGLOSS 00073 processing.

ERGLOSS 00074 response time: (1) The time between the submission of an
ERGLOSS 00075 item of work to a computing system and the return of
ERGLOSS 00076 results. (2) In systems with time sharing, the time between
ERGLOSS 00077 the end of a block or line-end character of terminal input
ERGLOSS 00078 and the display of the first character of system response at
ERGLOSS 00080 the terminal.

ERGLOSS 00082 verification mode: In systems with the time sharing, a mode
ERGLOSS 00083 of operation under the EDIT command in which all subcommands
ERGLOSS 00084 are acknowledged and any textual changes are displayed as
ERTABCON 00001 they are made.

ERTABCON 00004

CONTENTS

CLCOVER 00030 GC20-1804-0

CLCOVER 00033 IBM VIRTUAL MACHINE FACILITY/370
CLCOVER 00035 COMMAND LANGUAGE USER'S GUIDE

CLCOVER 00037 VM/370 is a system which manages the resources of a single
CLCOVER 00038 System/370 computer such that multiple computing systems
CLCOVER 00039 appear to exist within it. Each such computing system is
CLCOVER 00040 said to run in a virtual machine. VM/370 consists of two
CLCOVER 00041 major components:

CLCOVER 00045 1. A control program (CP), which controls the real
CLCOVER 00045 computer and allows multiple computing systems to run
CLCOVER 00046 under its control.

CLCOVER 00048 2. A Conversational Monitor System (CMS), which runs in a
CLCOVER 00049 virtual machine under control of the control program
CLCOVER 00050 and allows the user to operate his virtual machine from
CLCOVER 00051 a remote terminal. CMS is a time-sharing system that
CLCOVER 00052 provides a general purpose conversational facility
CLCOVER 00053 suitable for program development and problem solving.

CLCOVER 00056 This publication describes the commands necessary to use the
CLCOVER 00057 components of VM/370; the Control Program, the
CLCOVER 00058 Conversational Monitor System, and the various subsystems
CLEDNOT 00001 that run under CMS.

CLEDNOT 00008 First Edition

CLEDNOT 00010 Changes are periodically made to the specifications herein;
CLEDNOT 00010 before using this publication in connection with the
CLEDNOT 00011 operation of IBM systems, refer to the lastest IBM
CLEDNOT 00012 System/360 and System/370 SRL newsletter, Order No.
CLEDNOT 00014 GN20-3060, for the editions that are applicable and current.

CLEDNOT 00015 Requests for copies of IBM publications should be made to
CLEDNOT 00017 your IBM representative or to the IBM branch office serving
CLEDNOT 00018 your locality.

CLEDNOT 00020 A form for reader's comments is provided at the back of this
CLEDNOT 00021 publication. If the form has been removed, address comments
CLEDNOT 00022 to IBM Corporation, Programming Publications, 545 Technology
CLEDNOT 00023 Square, Cambridge, Massachusetts, 02139.

CLEDNOT 00024

CLPREFAC 00006

PREFACE

CLPREFAC 00007
CLPREFAC 00008
CLPREFAC 00009

This publication is intended as an introduction, and user's guide and reference manual for, all users of the command languages (CP and CMS) of VM/370.

CLPREFAC 00011

Recommended Prerequisites:

CLPREFAC 00016
CLPREFAC 00017
CLPREFAC 00019
CLPREFAC 00020
CLPREFAC 00022
CLPREFAC 00023

- Introduction to Virtual Storage in System/370, Part I (Order No. GR-4260)
- IBM Virtual Machine Facility/370, Introduction (Order No. GC20-1800)
- IBM Virtual Machine Facility/370, Terminal User's Guide (Order No. GC20-1810)

CLPREFAC 00026
CLPREFAC 00026
CLPREFAC 00027

For commands for which additional prerequisite or supplemental information is needed, this fact is noted in the text of the command definition.

CLPREFAC 00029

The major divisions of this book are:

CLPREFAC 00032

- Introduction

CLPREFAC 00034

- CMS Commands

CLPREFAC 00036

- CP Commands

CLPREFAC 00038

- Appendixes

CLPREFAC 00040

- Glossary

CLPREFAC 00042

- Index

CLPREFAC 00046
CLPREFAC 00047
CLPREFAC 00048
CLPREFAC 00048
CLPREFAC 00049
CLPREFAC 00050
CLPREFAC 00051
CLPREFAC 00052

The Introduction describes the VM/370 environment, virtual machine concepts, and system components and provides background information necessary in order to access and use the VM/370 system. It discusses CP and CMS file management, procedures for establishing communication with the VM/370 system, general format of the commands and describes the command syntax, typing conventions, abbreviations, and other aids used to minimize entry requirements

CLPREFAC 00053
CLPREFAC 00055

The CMS Command section describes the format and use of the CMS commands. Examples are included.

CLPREFAC 00056
CLPREFAC 00057
CLPREFAC 00058

The CP command section describes the format and use of commands which are executed in the control program environment.

CLPREFAC 00060
CLPREFAC 00060
CLPREFAC 00062

The Appendixes present a detailed sample terminal session, discuss terminal usage, provide supplemental information about file management facilities, discuss the CMS Batch

CLPREFAC 00063
CLPREFAC 00064
Facility, and discuss compatibility of VM/370 with the CP/67-CMS system available for the System/360 Model 67.

CLPREFAC 00065
CLPREFAC 00067
The Glossary contains definitions of terms that appear in the text of this publication.

CLPREFAC 00068
CLPREFAC 00069
The Index contains the location of terms and subjects discussed in the text.

CLPREFAC 00071
CLPREFAC 00072
CLPREFAC 00073
In this publication, the term 3330 is used to refer to both the IBM 3330-A1 Disk Storage Facility and the IBM 3330-A2 Disk Storage Facility.

NOTE

Information in this publication (if any) about the following is for planning purposes only:

- The Virtual=Real Performance Option
- The Dedicated Channel Performance Option
- The Virtual and Real Channel-to-Channel Adapter
- The CMS Batch Facility
- The Program Product, PL/I Optimizing Compiler

CLTABCON 00008 **CONTENTS**

CLFIG 00005

FIGURES

CLVIRCON 00009 **INTRODUCTION**

CLVIRCON 00013 **VIRTUAL MACHINE CONCEPT**

CLVIRCON 00014 VM/370 is a system which manages the resources of a single
CLVIRCON 00015 computer such that multiple computing systems appear to
CLVIRCON 00016 exist. Each such computing system is said to run in a
CLVIRCON 00017 virtual machine.

CLVIRCON 00018 VM/370 consists of (1) a control program (CP), which
CLVIRCON 00019 controls the real computer and allows multiple computing
CLVIRCON 00020 systems to run under its control, and (2) the Conversational
CLVIRCON 00022 Monitor System (CMS), which runs in a virtual machine. Each
CLVIRCON 00023 virtual machine is a functional simulation of the real
CLVIRCON 00024 System/370 in which CP is running.

CLVIRCON 00026 VM/370 maintains a file called the "user directory" which
CLVIRCON 00026 contains information about the configuration of each virtual
CLVIRCON 00028 machine that may run in the real machine. The virtual
CLVIRCON 00029 machine configuration includes counterparts to the
CLVIRCON 00030 components of a real System/370:

CLVIRCON 00033 - virtual system console

CLVIRCON 00035 - virtual storage

CLVIRCON 00037 - virtual CPU

CLVIRCON 00039 - virtual Input/Output devices

CLVIRCON 00044 It is the function of the control program of VM/370 to make
CLVIRCON 00045 these components appear real to the operating system which
CLVIRCON 00047 is controlling the work flow of the virtual machine.

CLVIRCON 00049 Virtual System Console: Since each VM/370 virtual machine
CLVIRCON 00050 user is, in effect, controlling his own operating system, he
CLVIRCON 00051 must have the means to control it in the same way the
CLVIRCON 00051 operator of a real machine does. The operation of a virtual
CLVIRCON 00052 machine is normally controlled by a terminal connected to
CLVIRCON 00053 the real System/370 by communication lines although the real
CLVIRCON 00055 system console or alternate system console may also be used.
CLVIRCON 00055 The remote terminal is used initially to access or log into
CLVIRCON 00056 VM/370. A user becomes active by identifying himself to
CLVIRCON 00057 VM/370 by keying in a pre-established user identification
CLVIRCON 00057 (userid). The userid is associated with a particular
CLVIRCON 00058 virtual machine configuration by an appropriate entry in the
CLVIRCON 00059 user directory. The user must also enter the correct
CLVIRCON 00060 password for his userid; at this time, printing of typed-in
CLVIRCON 00060 characters is inhibited (if the terminal is so equipped) or

masking (a combination of over-struck characters) is provided as an aid to maintaining the security of the password.

Once VM/370 has been properly accessed, the remote terminal provides the user with facilities equivalent to those of the operator's console for his virtual machine. The terminal acts as the operator's console, and the control program provides a set of commands to simulate the control panel of the real System/370. These commands are available to the user to provide facilities such as displaying and storing in virtual storage, IPLing a virtual device, or performing a system reset or restart. Further terminal communication would be under control of the virtual machine's operating system until the control program is again entered. CP commands can be entered when the user is in CP command or "console function" mode. This mode is normally entered by the use of the terminal's Attention (ATTN) key (or its equivalent), but can also be entered via special CMS commands, use of special interfaces, or upon abnormal termination of the virtual machine's operating system.

Virtual Storage: The storage size of the virtual machine is defined in its directory entry and may differ among virtual machines. This storage can appear as real storage to whatever program executes in the virtual machine. CP manages the real main storage resource as virtual storage, utilizing the System/370 Dynamic Address Translation facilities this virtual storage management is transparent to the virtual machine. As an example, three virtual machines requiring 256K each could be run on a single real computing system having 384K of real storage. Because there is not room in real storage for all users' virtual storage, the system uses a technique called "paging". Virtual storage is divided into 4096-byte blocks of real storage called pages. All but currently active pages are kept by the control program on direct access storage; this direct access space is preallocated but is assigned only on a demand basis. As active and inactive pages change status they are moved in and out of real storage on demand. While the paging operation is being performed for one virtual machine, another virtual machine can be executing. The paging operation, and resultant allocation of real storage to a given user's pages, is transparent to the user.

Because the virtual storage appears real to the virtual machine, operating systems such as DOS or OS, which cannot directly manage virtual storage, can be run without change under control of VM/370, and thus utilize the facilities and advantages of virtual storage. When an operating system such as CMS, OS/VS1, VS2, or DOS/VS (which can manage virtual storage via the Dynamic Address Translation facility of a System/370) is run in a virtual machine, the storage of the virtual machine is itself managed as virtual storage.

CLVIRCON 00099 The minimum virtual storage size that can be specified for a
CLVIRCON 00100 user is 4K (4096 bytes or 1 page). The maximum virtual
CLVIRCON 00100 storage size that can be specified is 16M (16,777,216 bytes
CLVIRCON 00101 or 4096 pages). Virtual storage size can be defined for any
CLVIRCON 00102 size between 4K and 16M in 4K increments. Each virtual
CLVIRCON 00102 machine defined in the user directory has an "initial" and a
CLVIRCON 00103 "maximum" virtual storage size specification. The "initial"
CLVIRCON 00104 size is automatically assigned to the user when he initially
CLVIRCON 00105 accesses the VM/370 system. The user can alter the size of
CLVIRCON 00106 virtual storage available to him up to his "maximum" by
CLVIRCON 00107 using the CP command DEFINE.

CLVIRCON 00108 Virtual CPU: VM/370 provides a CPU resource to each active
CLVIRCON 00110 virtual machine user through time-slicing techniques. The
CLVIRCON 00111 virtual machine can be single task or multi-task. Thus
CLVIRCON 00111 single or multiple copies of CMS, DOS/360 and VS, OS/MFT,
CLVIRCON 00113 MVT, VS1, VS2, PCP, and BPS and VM/370 can all run in
CLVIRCON 00114 concurrently executing virtual machines.

CLVIRCON 00116 Virtual Input/Output: All virtual machine I/O operations are
CLVIRCON 00117 initiated by the virtual machine's operating system, but the
CLVIRCON 00118 channel programs involved are normally handled by CP, which
CLVIRCON 00119 must translate them into real machine I/O operations. This
CLVIRCON 00120 requires two translations accomplished as follows: CP
CLVIRCON 00121 intercepts all user I/O when STARTIO (SIO) is issued. It
CLVIRCON 00121 "maps" or converts virtual device addresses into real device
CLVIRCON 00122 addresses, translates virtual storage addresses into real
CLVIRCON 00123 storage addresses, ensures that all necessary pages are in
CLVIRCON 00124 real storage, builds the required channel program (that is,
CLVIRCON 00125 the appropriate number of channel command words) for the
CLVIRCON 00125 user, and issues SIO when the channel is free. The virtual
CLVIRCON 00127 machine does not regain control from CP from the time it
CLVIRCON 00128 issues an SIO until CP issues the real SIO and delivers the
CLVIRCON 00128 resulting condition code to the virtual machine. In the
CLVIRCON 00129 meantime, other virtual machines may be executing. When CP
CLVIRCON 00130 receives an interrupt indicating I/O completion, it
CLVIRCON 00132 simulates the proper I/O interrupt when control is returned
CLVIRCON 00133 to the virtual machine.

CLVIRCON 00134 Physical DASD devices may be logically subdivided into
CLVIRCON 00135 smaller units called "minidisks" for the use of virtual
CLVIRCON 00136 machines; thus, when a virtual machine user refers to a
CLVIRCON 00137 disk, he is actually accessing a portion of a physical disk.
CLVIRCON 00137 For example, a real 2314 pack may be divided into three
CLVIRCON 00139 virtual disks or "minidisks".

CLVIRCON 00142 Virtual machine unit record I/O is normally spooled by CP
CLVIRCON 00143 onto a special spooling disk. Thus, any card deck to be
CLVIRCON 00143 "read" by a virtual machine would, in the normal case, have
CLVIRCON 00144 been read in by CP before the user requests it for his
CLVIRCON 00144 virtual machine, or it would have been transferred to that
CLVIRCON 00145 user from another user's files. If the cards are to be
CLVIRCON 00146 entered through the real card reader, the card deck must be

preceded by a special VM/370 header card containing the owner's userid, so that CP can know who owns the deck and properly identify, label, and provide DASD storage space for the card-image file. Later, when the virtual machine has read the card images, a card reader end-of-file is simulated. Punch and printer output, similarly spooled, is not queued for physical output until CP is notified of end-of-file in one of three ways:

- the VM/370 user logs off the system (end-of-file is assumed).

- the CP command CLOSE; entered either explicitly, or implicitly by the CMS commands, PRINT and PUNCH; specifies the virtual address of the device to be closed.

- CP detects an invalid Channel Command Word (CCW) addressed to the device (end-of-file is assumed).

Further output for a closed device is assumed to start a new file. So that the system operator can separate physical output, VM/370 precedes all printed and punched output files with a record containing the userid.

CLENVIR 00003

VM/370 ENVIRONMENT

CLENVIR 00005 Each input line typed at the terminal by a user is
CLENVIR 00006 transmitted to the VM/370 system, where it is processed
CLENVIR 00007 (examined, and accepted or rejected) by a given routine.
CLENVIR 00007 The particular routine by which input is processed is
CLENVIR 00008 determined by which portion of the system has control at the
CLENVIR 00009 time the input line is entered. Each portion of the system
CLENVIR 00010 to which input may be entered constitutes a unique
CLENVIR 00010 environment, and only a subset of all possible input is
CLENVIR 00011 acceptable to any given environment. Figure 1 shows the
CLENVIR 00012 environments of the VM/370 system and shows how each
CLENVIR 00013 environment is entered.

CLENVIR 00015 The input-processing routines fall into three main
CLENVIR 00015 categories: input is received by either the Control Program
CLENVIR 00016 (CP environment), a central CMS service routine (CMS Command
CLENVIR 00017 environment), or a particular CMS command (DEBUG and EDIT
CLENVIR 00017 environments). Input lines which are acceptable to the CP
CLENVIR 00019 and CMS environments are referred to as commands.

CLENVIR 00020 Certain CMS commands cause CMS subenvironments to be
CLENVIR 00021 entered. Examples of these are the DEBUG and EDIT commands.
CLENVIR 00021 Lines acceptable to the environments of these commands are
CLENVIR 00022 referred to as "subcommands", or merely "input", depending
CLENVIR 00022 on the particular mode which is entered when the command is
CLENVIR 00023 issued.

CLENVIR 00027 1) If the EDIT command is entered for a file which already
CLENVIR 00027 exists, the EDIT mode is entered, allowing the contents
CLENVIR 00028 of the existing file to be examined, added to, deleted,
CLENVIR 00029 changed, or re-arranged. The user may cause the input
CLENVIR 00030 mode to be entered so that he can add to the existing
CLENVIR 00031 file.

CLENVIR 00033 2) If an EDIT command is issued for a file which does not
CLENVIR 00034 currently exist, the EDIT mode is entered, and the user
CLENVIR 00035 may cause the INPUT mode to be entered so that he may
CLENVIR 00036 directly create a file on disk, thus eliminating the
CLENVIR 00037 keypunch step. The INPUT mode accepts any input typed
CLENVIR 00037 at the terminal, and this input becomes a part of the
CLENVIR 00038 file being created. Because no check is made to
CLENVIR 00038 determine the acceptibility of input to this
CLENVIR 00039 environment, lines which are keyed in are termed
CLENVIR 00040 "input".

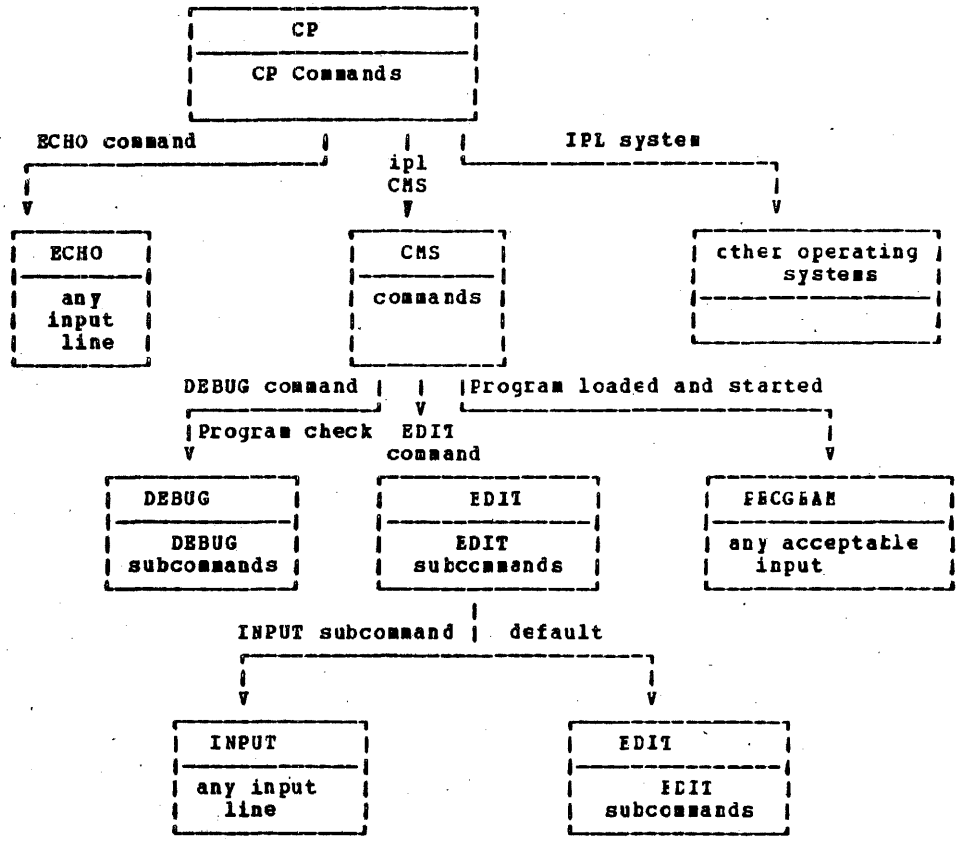
CLENVIR 00043 The ECHO environment is entered when the CP command ECHO is
CLENVIR 00043 keyed in. All data lines entered in the ECHO environment
CLENVIR 00044 are transmitted unchanged back to the terminal from which
CLENVIR 00045 they were received.

CLENVIR 00047 To verify which environment the user is in, the RETURN key
CLENVIR 00048 (or its equivalent) can be depressed with no characters

CLENVIR 00049 entered.

CLENVIR 00050 Various actions by the user cause control to pass from one
CLENVIR 00051 environment to another. These actions are specified in
CLENVIR 00052 detail throughout this guide. The Attention key (or its
CLENVIR 00053 equivalent) can always be used to transfer control to the CP
CLENVIR 00054 environment from any of the other environments.

CLENVIR 00062 VM/370



CLENVIR 00103 Figure 1. VM/370 Environments

CLFILM 00006

CMS FILE MANAGEMENT

CLFILM 00007
CLFILM 00008
CLFILM 00008
CLFILM 00009

The Conversational Monitor System (CMS) uses a specially designed file management scheme to manipulate and maintain user and system data files on direct access storage devices (disks).

CLFILM 00010
CLFILM 00011
CLFILM 00012

Note: The terms "user disk", "disk", and "disk space" are used in this discussion to denote virtual disks rather than physical disks.

CLFILM 00015

DISK ARRANGEMENT

CLFILM 00016
CLFILM 00017
CLFILM 00019
CLFILM 00019
CLFILM 00021
CLFILM 00021
CLFILM 00022
CLFILM 00023
CLFILM 00024
CLFILM 00024
CLFILM 00025
CLFILM 00025
CLFILM 00027
CLFILM 00028

Each user has access to a maximum of nine user disks plus the CMS System disk, which is shared among all users. Each user disk has a name in the form of a six character label associated with it. Within the defined disk space, records are automatically allocated on demand whenever a CMS file is to be created or expanded. When a CMS file is deleted the space originally occupied by the records of the file is automatically returned to "available" status. CMS disks are pre-formatted into 800-byte physical blocks; logical records are defined within these fixed-length physical blocks. Each disk space has an index or directory which lists the contents (files) of that disk and contains pointers to the 800-byte blocks which constitute the files on the disk. Blocks of a single file need not be contiguous.

CLFILM 00029
CLFILM 00030
CLFILM 00031
CLFILM 00031
CLFILM 00032
CLFILM 00034

A user file directory is created for each disk the user may refer to. This directory contains a pointer to a File Status Table (FST) for each file on the disk. The FST refers to a field which contains a record of where the 800-byte data blocks for the file are located on the disk. Figure 2 shows the relationship between these items.

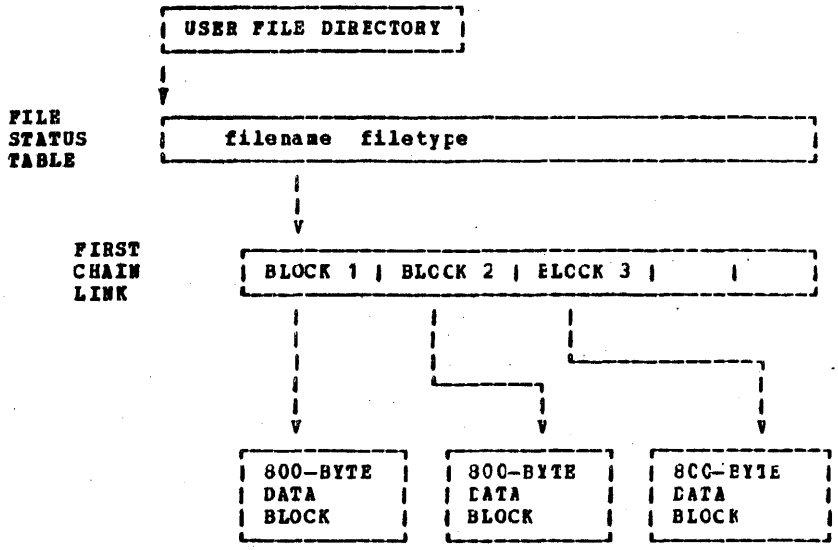


Figure 2. CMS Disk File Structure

Directory Naming

Each of the nine user file directories and the system disk file directory has a name in the form of a single alphabetic character, A, B, C, D, E, F, G, S, Y, or Z. When a user requests that a disk be made available to his system, he assigns a name to the directory of that disk. The 'S' directory denotes the system disk.

Disk Extensions

A file directory, in addition to encompassing the files of the disk which it directly represents, may logically be used to refer to the disk files of another disk. The latter directory is referred to as an extension of the former directory, with the former known as the parent directory.

CLFILM 00090

FILE NAMING

CLFILM 00091
CLFILM 00092
CLFILM 00092
CLFILM 00093
CLFILM 00094

Each file created under CMS control or created for use by CMS commands, is represented by a file identifier by which a user refers to the file. A file identifier consists of three components--filename, filetype, and filemode, in the following format:

CLFILM 00097

filename filetype filemode

CLFILM 00101
CLFILM 00103

filename **fn** is a one to eight alphameric character name assigned (in most cases) by the user.

CLFILM 00106
CLFILM 00108
CLFILM 00109
CLFILM 00110
CLFILM 00111

filetype **ft** is a one to eight alphameric character name used as a descriptor or as a qualifier of the filename. Refer to "Appendix D: Reserved Filetype Descriptions" for restrictions concerning the assignment of a filetype.

CLFILM 00114
CLFILM 00116
CLFILM 00116
CLFILM 00116
CLFILM 00117
CLFILM 00118

filemode **fm** is a two character field. The first character is alphabetic and denotes the name of the file directory of the disk on which the file resides. The second character is numeric and denotes the mode by which the file is accessed. Valid access modes are shown in Figure 3.

CLFILM 00127
CLFILM 00128
CLFILM 00129
CLFILM 00130
CLFILM 00131
CLFILM 00132
CLFILM 00133
CLFILM 00134
CLFILM 00135
CLFILM 00136
CLFILM 00137
CLFILM 00138
CLFILM 00139
CLFILM 00140
CLFILM 00141
CLFILM 00142
CLFILM 00143
CLFILM 00144
CLFILM 00145
CLFILM 00146

MODE	MEANING	ACCESS ALLOWED
0	private	file is accessible only when the directory is used in read/write status.
1	R/W	file is accessible in read/write status only if the directory is in read/write status.
2	R/O	file cannot be modified in place.
3	R/E	file will be erased after it is read.
4	OS	file is in the format produced by the simulated OS access methods under CMS.
5	R/W	same as Mode 1.
6-9		reserved.

CLFILM 00149

Figure 3. Access Modes for CMS files

DIRECTORY SEARCH CRITERIA

When a CMS virtual machine is capable of using two or more file directories, a problem of search order arises. If, in a specific CMS command, the user has the ability of designating which directory to examine for the file, there is no doubt as to where and how to search directories. But in some cases, if the user does not or cannot designate a directory, there must be a set search pattern for scanning directories to find a specified data file or available disk space.

Standard Search Order

To find a file, or space for writing a file, the user disks are searched in a system-defined alphabetical order: A, B, C, D, E, F, G, S, Y, and Z.

When one file directory is made an extension of another directory by the CMS command ACCESS, the extension directory is searched immediately after the parent. Thus, the standard search order may be changed for the duration of the terminal session.

Filemode Specification

The filemode for a particular file may be indicated in one of the four ways shown in Figure 4.

explicitly specified	fm	filemode entered
implicitly specified	*	* or = entered
	=	
specified by default	default	filemode left blank
not specified	null	command issued which cannot contain a file-mode specification

Figure 4. Methods of Filemode Specification

The following rules have been established to govern the action of all commands relative to the four methods of specifying the filemode (fm; *, =; blank; null) and the two

CLFILM 00211 possible access modes (Read, Write).

CLFILM 00213 Reading:

CLFILM 00216 fm when a mode designation such as B2 or B is
CLFILM 00216 specified within the command line, the unique
CLFILM 00217 file directory and all its extensions are
CLFILM 00217 searched. If the specified directory is itself
CLFILM 00219 an extension, its parent is not searched.

CLFILM 00221 default if the command entered does not contain a
CLFILM 00222 filemode (that is, the user chooses to leave the
CLFILM 00223 filemode specification blank), the disk with a
CLFILM 00224 mode designation of "A" is searched.

CLFILM 00226 *,null a search is made of all active disks in the
CLFILM 00227 predetermined, hierarchical order.

CLFILM 00230 Writing:

CLFILM 00232 fm when a mode designation such as A1 or A is
CLFILM 00233 specified, that unique disk is written upon. If
CLFILM 00234 the specified mode letter designates a read/only
CLFILM 00235 disk, the command aborts.

CLFILM 00237 = when "=" is entered, it is used for commands
CLFILM 00238 which have read/write functions to indicate
CLFILM 00238 that writing should be done to the disk from
CLFILM 00239 which the reading took place.

CLFILM 00242 default, for commands with read/write functions, writing
CLFILM 00243 null,* is attempted to the disk or the parent of the
CLFILM 00244 disk from which reading was done. If no room is
CLFILM 00245 available on either of the disks mentioned or if
CLFILM 00246 both disks are read/only, writing is done only
CLFILM 00247 to the primary disk.

CLFILM 00250 "Appendix C: Disk Determination" presents a table relating
CLFILM 00251 CMS commands, methods of filemode specification, and the
CLFILM 00252 criteria used in choosing a file directory for reading and
CLFILM 00253 writing.

CLFILM 00257 RESERVED FILETYPES

CLFILM 00258 Filetype (ft) is an eight alphameric character component of
CLFILM 00259 the CMS file identification and is used a filename
CLFILM 00260 descriptor or qualifier. For certain commands in which only
CLFILM 00261 a filename may be issued, CMS assumes a filetype for the
CLFILM 00262 specified filename.

CLFILM 00263 "Appendix D: Reserved Filetype Descriptions" contains a

```
CLFILM   00264
CLCHARST 00001
```

summary of filetypes which have special significance tc the CMS system.

CLCHARST 00009

CHARACTER SET USAGE

CLCHARST 00011
CLCHARST 00012
CLCHARST 00013

CP and CMS commands may be entered using a combination of characters from six different character sets. The contents of each of the character sets is described in Figure 5.

CLCHARST 00022
CLCHARST 00023
CLCHARST 00024
CLCHARST 00025
CLCHARST 00026
CLCHARST 00027
CLCHARST 00028
CLCHARST 00029
CLCHARST 00030
CLCHARST 00031
CLCHARST 00032
CLCHARST 00033
CLCHARST 00034
CLCHARST 00035
CLCHARST 00036
CLCHARST 00037
CLCHARST 00038
CLCHARST 00039
CLCHARST 00040
CLCHARST 00041
CLCHARST 00042
CLCHARST 00043
CLCHARST 00044

Character Set	Contents	
Separator	Blank	
National	Dollar Sign	$
	Pound Sign	#
	At Sign	@
Alphabetic	Upper Case	A - Z
	Lower Case	a - z
Numeric	Numbers	0 - 9
Alphameric	National	$, #, @
	Alphabetic	A - Z
		a - z
	Numeric	0 - 9
Special		all other characters

CLCHARST 00046

Figure 5. Character Sets

CLNAMCON 00005 **NAMING CONVENTIONS**

CLNAMCON 00009 1. All command names, keyword parameters, filenames, and
CLNAMCON 00011 filetypes must meet the following requirements:

CLNAMCON 00016 a. All characters must be alphameric.

CLNAMCON 00018 b. The length of each field may not exceed eight
CLNAMCON 00019 characters. If the field exceeds eight characters,
CLNAMCON 00020 it is truncated without any indication to the
CLNAMCON 00021 user.

CLNAMCON 00024 2. Passwords and userids may be any combination of up to
CLNAMCON 00025 eight characters, including all numeric and special
CLNAMCON 00026 characters.

CLNAMCON 00028 3. Volume serial labels may be any combination of up to
CLNAMCON 00029 six alphameric characters.

CLNAMCON 00031 4. No name can contain a character normally used by VM/370
CLNAMCON 00032 for logical line-editing as discussed in "Appendix B:
CLNAMCON 00032 Terminal Considerations" in the discussion of Line
CLNAMCON 00033 Editing

CLCOMSYN 00006 ## COMMAND SYNTAX

CLCOMSYN 00008 A command line consists of a command name, usually followed
CLCOMSYN 00009 by one or more positional operands, and possibly by an
CLCOMSYN 00010 option list. A command name is typically a familiar English
CLCOMSYN 00010 word, usually a verb, that describes the function of the
CLCOMSYN 00011 command. The positional operands provide specific
CLCOMSYN 00012 information required for the command to perform the
CLCOMSYN 00013 requested operation. The option list begins with a left
CLCOMSYN 00013 parenthesis and indicates which command options are to be
CLCOMSYN 00014 used. The right or closing parenthesis is optional and may
CLCOMSYN 00016 be omitted if no other information is to be entered. The
CLCOMSYN 00016 user may enter a left parenthesis and a carriage return
CLCOMSYN 00018 which signals that default options are in effect. This is
CLCOMSYN 00019 equivalent to ending the input line after the positional
CLCOMSYN 00019 parameters. However; in the RUN command, if he enters a
CLCOMSYN 00021 left parenthesis followed by a carriage return, null
CLCOMSYN 00022 arguments will be assumed; in COPYFILE and FILEDEF commands,
CLCOMSYN 00023 an error message will be generated. The general command
CLCOMSYN 00024 format is as follows:

CLCOMSYN 00035 ┌──┐
CLCOMSYN 00036 | command name | [operands...] [(options...[)]] |
CLCOMSYN 00037 └──┘

CLCOMSYN 00042 For example, to assemble a source file named TEST and print
CLCOMSYN 00043 the listing, type

CLCOMSYN 00046 ```
CLCOMSYN 00048 ASSEMBLE TEST (PRINT)
CLCOMSYN 00050 - or -
ASSEMBLE TEST (PRINT
```

CLCOMSYN 00055 Command parameters must be separated from one another by one
CLCOMSYN 00055 or more separator characters (blanks) and the command name
CLCOMSYN 00056 must be separated from the first parameter by one or more
CLCOMSYN 00057 blanks. Any parameters entered other than those required are
CLCOMSYN 00058 treated as comments. See Notes 1 through 3 below for
CLCOMSYN 00059 exceptions to this. There are two types of command
CLCOMSYN 00060 parameters:

CLCOMSYN 00063 1. Positional parameters
CLCOMSYN 00065 2. Keyword parameters

CLCOMSYN 00068 ### POSITIONAL PARAMETERS

CLCOMSYN 00070 Positional parameters must be entered first in the parameter
CLCOMSYN 00071 string, and they must be in a specific order.

CLCOMSYN 00073 For the purpose of syntax checking, positional parameters
CLCOMSYN 00074 are divided into parameters that include delimiters as part
CLCOMSYN 00075 of their definition (delimiter-dependent parameters) and
```

CLCOMSYN 00076 parameters that do not include delimiters as part of their
CLCOMSYN 00077 definition (delimiter-independent parameters).

CLCOMSYN 00080 Delimiter Dependent Parameters

CLCOMSYN 00082 Those parameters that include delimiters as part of their
CLCOMSYN 00083 definition are called delimiter-dependent parameters.
CLCOMSYN 00083 Delimiter-dependent parameters used in the command language
CLCOMSYN 00084 are:

CLCOMSYN 00086 1. FILEID
CLCOMSYN 00087 2. STRING
CLCOMSYN 00088 3. SSYMBOL
CLCOMSYN 00089 4. LABEL

CLCOMSYN 00094 FILEID

CLCOMSYN 00095 A fileid consists of all the information necessary to
CLCOMSYN 00097 uniquely identify a file. The format is as follows:

CLCOMSYN 00099 fn ft fm

CLCOMSYN 00100 filename (fn) is the user-supplied name of the file.
CLCOMSYN 00102 The filename may be up to eight alphameric characters.

CLCOMSYN 00103 filetype (ft) is a descriptive qualifier for the file.
CLCOMSYN 00105 It may be up to eight alphameric characters.

CLCOMSYN 00106 filemode (fm) specifies in which logical directory the
CLCOMSYN 00107 file exists, and the mode by which the file may be
CLCOMSYN 00108 accessed.

CLCOMSYN 00110 An asterisk or an equal sign may be used in some
CLCOMSYN 00111 commands in place of fn, ft, or fm. An asterisk
CLCOMSYN 00111 indicates that all files at that level are to be
CLCOMSYN 00112 considered. For example, "fn * fm" means that the
CLCOMSYN 00113 specified function is to be performed for all files
CLCOMSYN 00114 with the given fn and fm. An equal sign indicates that
CLCOMSYN 00115 the name is to be the same as the previously entered
CLCOMSYN 00116 name at the same level.

CLCOMSYN 00119 STRING

CLCOMSYN 00121 A string is a group of characters between two alike
CLCOMSYN 00122 self-defining delimiter characters, such as

CLCOMSYN 00124 /string/

CLCOMSYN 00126 or, the group of characters between a self-defining
CLCOMSYN 00126 delimiter character and the end of a logical line, such
CLCOMSYN 00127 as

CLCOMSYN 00129 /string

CLCOMSYN 00131 The same self-defining delimiter character can be used
CLCOMSYN 00132 to delimit two contiguous strings, such as

CLCOMSYN 00134 /string/string/ or /string/string

CLCOMSYN 00135 A null string, which indicates that a positional
CLCOMSYN 00136 parameter has not been entered, is defined as two
CLCOMSYN 00137 contiguous delimiters or a delimiter and the end of a
CLCOMSYN 00138 logical line.

CLCOMSYN 00141 SSYMBOL

CLCOMSYN 00142 A special symbol (SSYMBOL) consists of an ampersand
CLCOMSYN 00143 followed by up to seven alphameric characters. An
CLCOMSYN 00144 SSYMBOL may be either an argument number or a symbolic
CLCOMSYN 00145 name. An argument number consists of an ampersand
CLCOMSYN 00146 followed by an integer from 0 to 255. A symbolic name
CLCOMSYN 00147 consists of an ampersand followed by up to seven
CLCOMSYN 00148 alphameric characters, the first of which must be an
CLCOMSYN 00149 alphabetic or national character.

CLCOMSYN 00153 LABEL

CLCOMSYN 00155 A label consists of up to seven characters preceded by
CLCOMSYN 00156 a dash, such as

CLCOMSYN 00158 -label1

CLCOMSYN 00162 Delimiter-Independent Parameters

CLCOMSYN 00164 A positional parameter that is not dependent on delimiters
CLCOMSYN 00165 is treated as a character string. It may consist of any
CLCOMSYN 00166 characters except a blank or carriage return.

CLCOMSYN 00168 KEYWORD PARAMETERS

CLCOMSYN 00170 Keyword parameters are specific names or symbols that have a
CLCOMSYN 00171 particular meaning to the system. Keywords may be coded in
CLCOMSYN 00173 any order within the command field in which they are used.

CLCOMSYN 00174 Keyword parameters can have other parameters associated with
CLCOMSYN 00175 them. These parameters, known as subfields, must be
CLCOMSYN 00176 separated from the keyword by a blank. A typical keyword
CLCOMSYN 00177 with a subfield is:

CLCOMSYN 00180 LINECNT 55
CLCOMSYN 00182 keyword subfield

CLCOMSYN 00184 Note 1: Except where noted, extraneous information entered
CLCOMSYN 00185 in a CP command line after any required operand or selected
CLCOMSYN 00186 options is treated as a comment.

The following CP commands will not accept comments in the command line due to the structure or function of the command. Any extraneous information will be rejected with an error message.

ACNT	ENABLE	SPOOL
BACKSPAC	FLUSH	START
CHANGE	IPL	STCP
CLOSE	LOADBUF	STORE
DCP	MSG	TERMINAL
DISABLE	ORDER	TRACE
DISPLAY	PURGE	WNG
DMCP	REPEAT	*
DUMP		

Note 2: In CMS commands if, after all possible positional parameters are entered, text is entered that is not a left parenthesis (indicating options), an error message is generated.

Note 3: In CMS commands, comments may be entered after the right parenthesis which indicates the end of the option list except in the ERASE command, where no comments are allowed.

Note 4: If conflicting or duplicate options are specified in CMS commands, the last one entered is in effect except in the following commands:

COPYFILE, ERASE, FILEDEF, FORMAT

CLNOTCON 00009 **NOTATIONAL CONVENTIONS**

CLNOTCON 00011 The notation used to define the command syntax in this
CLNOTCON 00013 publication is described in the following paragraphs:

CLNOTCON 00016 1. Minimum truncation and abbreviation

CLNOTCON 00017 Command names may be specified by typing any part of
CLNOTCON 00018 the name from the minimum truncation to the full name.
CLNOTCON 00018 The minimum truncation, if any, is noted below the full
CLNOTCON 00019 name of the command.

CLNOTCON 00023 Example: ASSEMBLE
CLNOTCON 00025 A

CLNOTCON 00029 indicates that A, AS, ASS, ASSE, ASSEM, ASSEMB,
CLNOTCON 00030 ASSEMBL, and ASSEMBLE are all valid specifications for
CLNOTCON 00031 this command name.

CLNOTCON 00032 Options may be specified in the same manner. A minimum
CLNOTCON 00033 truncation, if any, is noted below the explanation of
CLNOTCON 00034 each option of a particular command. If no minimum
CLNOTCON 00035 truncation is noted, the entire word must be entered.

CLNOTCON 00036 A second series of characters below the command name or
CLNOTCON 00037 option indicates that an abbreviation, (that is, a
CLNOTCON 00038 collection of characters which is not a simple
CLNOTCON 00039 truncation) is also valid.

CLNOTCON 00041 Example: READER name
CLNOTCON 00042 R minimum truncation
CLNOTCON 00043 RDR valid abbreviation

CLNOTCON 00049 2. The following symbols are used to define the command
CLNOTCON 00050 format and should never be typed when the actual
CLNOTCON 00051 command is entered.

CLNOTCON 00053 underscore _
CLNOTCON 00054 braces { }
CLNOTCON 00055 brackets [ ]
CLNOTCON 00056 ellipsis ...
CLNOTCON 00057 logical OR |

CLNOTCON 00062 3. Uppercase letters and words, and the symbols listed
CLNOTCON 00063 below, should be typed as specified in the statement
CLNOTCON 00064 format.

CLNOTCON 00066 asterisk *
CLNOTCON 00067 comma ,
CLNOTCON 00068 hyphen -
CLNOTCON 00069 equal sign =
CLNOTCON 00070 parentheses ( )
CLNOTCON 00071 period .

CLNOTCON 00072 colon :

CLNOTCON 00077 4. Lowercase letters, words, and symbols appearing in the
CLNOTCON 00078 command format represent variables for which specific
CLNOTCON 00079 information should be substituted. For example, "fn ft
CLNOTCON 00080 fm" could be typed in as "MYFILE EXEC A1."

CLNOTCON 00083 5. There are two ways that choices are represented in the
CLNOTCON 00085 command formats. One is by stacking them:

CLNOTCON 00087 A
CLNOTCON 00088 B
CLNOTCON 00089 C

CLNOTCON 00092 The other is by separating them with logical OR
CLNOTCON 00093 characters.
CLNOTCON 00094
CLNOTCON 00095 A|B|C

CLNOTCON 00100 6. Punctuation marks joining lowercase letters, words, and
CLNOTCON 00101 symbols indicate that only one value is to be
CLNOTCON 00102 substituted for the variable.

CLNOTCON 00106 Examples: 1. If "record-length" appears in a
CLNOTCON 00107 command definition, only one value
CLNOTCON 00107 should be substituted (for example,
CLNOTCON 00108 20).

CLNOTCON 00111 2. If "n'cyl" appears in a command
CLNOTCON 00111 definition, only one value should be
CLNOTCON 00112 substituted (for example, 5).

CLNOTCON 00116 7. An underscore indicates an assumed default option. If
CLNOTCON 00117 an underscored choice is selected, it need not be typed
CLNOTCON 00118 when the command is entered.

CLNOTCON 00123 Example: The representations

CLNOTCON 00125 A
CLNOTCON 00126 B or A|B|C
CLNOTCON 00127 C

CLNOTCON 00130 indicates that either A, B, or C may be selected.
CLNOTCON 00132 However, if B is selected, it need not be specified.

CLNOTCON 00135 8. The use of braces denotes choices, one of which must be
CLNOTCON 00140 selected.

Example: The representations

or {A|B|C}

indicate that the user must specify either A, B, or C. If a list of choices is enclosed by neither brackets or braces, it is to be treated as if enclosed by braces.

9) The use of brackets denotes choices, one of which may be selected.

Example: The representations

or [A|B|C]

indicate that the user may code A, B, or C, or he may omit the field.

10. An ellipsis indicates that the preceding item or group of items may be repeated more than once in succession.

Example: The representation

(option...)

indicates that more than one option may be coded within the parentheses.

CLLOGON 00005

LOGGING PROCEDURES

CLLOGON 00007 This section describes the procedures which must be
CLLOGON 00008 performed at the terminal to begin and terminate use of the
CLLOGON 00009 VM/370 system. Before the VM/370 facilities are made
CLLOGON 00009 available to a user, he must identify himself to the control
CLLOGON 00010 program (CP) by giving his userid and his password (two
CLLOGON 00010 identifiers which are assigned to him at the time he is
CLLOGON 00011 authorized to use the system). This identification
CLLOGON 00012 procedure is referred to as Login. When Login is completed,
CLLOGON 00012 a command may be issued to initialize CMS, as described
CLLOGON 00013 below.

CLLOGON 00014 When the user has completed his use of the system, he
CLLOGON 00016 signals this fact by issuing a LOGOUT command to CP. The
CLLOGON 00016 interval of time between VM/370 Login and VM/370 Logout is
CLLOGON 00018 referred to as a terminal session.

CLLOGON 00019 Note: During the Login procedure, CP uses the teleprocessing
CLLOGON 00021 line time-out feature when reading the userid and password.
CLLOGON 00022 If a user fails to type any character for a specified length
CLLOGON 00022 of time (normally 28 seconds), the line will time-out and be
CLLOGON 00023 disabled.

CLLOGON 00025

Login Procedure

CLLOGON 00026 After the terminal has been prepared for use as described in
CLLOGON 00028 the IBM Virtual Machine Facility/370, Terminal User's Guide
CLLOGON 00030 (Order No. GC20-1810) the procedure described below must be
CLLOGON 00030 performed in order to gain access to the VM/370 system.
CLLOGON 00032 (Input may be entered in either uppercase or lowercase).

CLLOGON 00035 1. A communications link to the computer must be
CLLOGON 00036 established as described in the Terminal User's Guide.

CLLOGON 00038 2. The system acknowledges that a communication link has
CLLOGON 00039 been established by typing one of the following
CLLOGON 00040 messages:

CLLOGON 00043 vm/370 online xxxxxxxx

CLLOGON 00045 -or-

CLLOGON 00047 xxxxxxxxx vm/370 online

CLLOGON 00048 The xxxxxxxx portion of the message consists of
CLLOGON 00049 meaningless characters and should be ignored. The
CLLOGON 00050 message types in lowercase letters.

3. At this point, the system must be notified that someone wishes to use it. To do this, the user must press the ATTN key (or its equivalent) once.

4. The system responds by unlocking the keyboard on an IBM 2741 or 1052, or waiting for input on the Teletype* Model 33 or 35. On the Teletype terminals, a period (.) is typed.

5. The user then identifies himself to CP by using the LOGIN command. Refer to the CP Command Section for a description of the various forms of the LOGIN command. If the user has entered the command name incorrectly, one of the following messages will type:

```
restart   xxxxxxx

     -or-

xxxxxxx   restart
```

The xxxxxxx portion of the message consists of meaningless characters and should be ignored. The user should repeat Step 5.

6. If the LOGIN command has been entered correctly, the system responds with one of the following messages:

```
ENTER PASSWORD:          ENTER PASSWORD:
                   -or-
                         ********
```

where ******** is a mask to safeguard the user's password.

7. The first form of the above message is received if

   a. the user's terminal is an IBM 2741 with the Print Inhibit feature. The user should enter his password; it does not print as he is typing it.

   b. the user did not specify the MASK option in the LOGIN command. The user should press the Carriage return key (or its equivalent) to indicate that he wishes the mask to type. The mask then types and the user should enter his password over the printed mask.

---

* Teletype is a trademark of the Teletype Corporation, Skokie, Illinois.

CLLOGON 00118 The second form of the message is typed if the user
CLLOGON 00119 specified the MASK option in the LOGIN command. The
CLLOGON 00121 user should type in his password over the printed mask.

CLLOGON 00125 8. If the password entered contains no errors, one or more
CLLOGON 00126 of the folowing messages types out:

CLLOGON 00132 a. LOGMSG setup time
CLLOGON 00134 b. LOGMSG messages with * as the first character
CLLOGON 00136 c. FILES messages (describing status of any virtual
CLLOGON 00137 unit record files)
CLLOGON 00139 d. LOGON LOGON time message

CLLOGON 00141 If any message other than one of the above types prints
CLLOGON 00143 out, the user should proceed to step 9.

CLLOGON 00145 Explanation of messages received at LOGON time:

CLLOGON 00147 a. If the VM/370 operator has previously set up any log
CLLOGON 00148 messages preceded by an asterisk (*), they are
CLLOGON 00149 automatically typed at this time. For example:

CLLOGON 00152 LOGMSG 09:34:54 02/15/72
CLLOGON 00154 *CP WILL BE UP 24 HOURS A DAY
CLLOGON 00156 *QUERY LOG FOR ADDITIONAL INFORMATION

CLLOGON 00158 Lower priority log messages may be present in the
CLLOGON 00159 VM/370 system, which do not have the preceding
CLLOGON 00160 asterisk. In order to see the additional entries,
CLLOGON 00161 the user must type the command:

CLLOGON 00163 QUERY LOGMSG

CLLOGON 00164 This results in the typing of all log messages
CLLOGON 00165 including * messages.

CLLOGON 00169 b. At this time, if there are any files in the
CLLOGON 00170 virtual reader, or output files for the printer
CLLOGON 00171 or punch, the message:

CLLOGON 00173 FILES: xxx RDR, xxx PRT, xxx PUN is typed.

CLLOGON 00176 c. The system then responds with the following
CLLOGON 00177 message:

CLLOGON 00179 LOGON AT hh:mm:ss zone weekday mm/dd/yy

CLLOGON 00181 where hh:mm:ss is the time of day, zone is the
CLLOGON 00182 time zone (for example, EST), weekday is the day
CLLOGON 00183 of the week, and mm/dd/yy is the date. This
CLLOGON 00184 message indicates that the password has been
CLLOGON 00184 accepted and the VM/370 LOGIN procedure is
CLLOGON 00185 complete. The control program (CP) environment

CLLOGON 00186 has been entered, and any CP command may be
CLLOGON 00186 issued. To load CMS into the user's virtual
CLLOGON 00187 machine, proceed to step 11.

CLLOGON 00193 9. Some of the reasons why VM/370 may reject a user's
CLLOGON 00194 request for access to VM/370 are:

CLLOGON 00201 • An invalid password was given.
CLLOGON 00204 • The userid is already being used by someone else.
CLLOGON 00207 • An invalid userid or no userid was given.
CLLOGON 00209 • The specified userid is not in the directory.
CLLOGON 00212 • VM/370 is already servicing the maximum number of
CLLOGON 00213 users.
CLLOGON 00216 • An error occurred while the directory entry for the
CLLOGON 00218 specified user was being read.

CLLOGON 00220 If the user receives a message indicating that any of
CLLOGON 00221 the above error conditions has occurred, he should take
CLLOGON 00223 appropriate action as indicated by the message.

CLLOGON 00227 10. Any operating system can now be loaded into the virtual
CLLOGON 00228 machine. To load CMS, the user should proceed to step
CLLOGON 00228 11. To load another operating system, the IPL command
CLLOGON 00229 should be issued to CP, specifying the device from
CLLOGON 00230 which the system is to be loaded. For example:

CLLOGON 00232 IPL 293

CLLOGON 00234 -or-

CLLOGON 00236 IPL 00C

CLLOGON 00237 If the device that is IPLed contains an operating
CLLOGON 00238 system (such as OS or DOS) the remote terminal becomes
CLLOGON 00239 the operator's console for that operating system.

CLLOGON 00242 11. CMS Initialization

CLLOGON 00246 a. To initialize CMS, one of the following forms of
CLLOGON 00247 the IPL command must be issued, followed by a
CLLOGON 00248 carriage return (or its equivalent).

CLLOGON 00250 IPL cuu

CLLOGON 00252 -or-

CLLOGON 00254 IPL CMS

CLLOGON 00255 This causes a copy of the CMS nucleus to be
CLLOGON 00257 brought into storage from disk. The first form of
CLLOGON 00258 the command indicates that the CMS nucleus is to
CLLOGON 00258 be retrieved from the device at virtual device
CLLOGON 00259 address cuu.

CLLOGON 00262 b. The message:

CLLOGON 00264 CMS Vx.y mm/dd/yy hh:mm:ss

CLLOGON 00265 is typed, where x is the version level and y is
CLLOGON 00266 the modification level, and the keyboard is
CLLOGON 00266 unlocked. The CMS Command environment has control
CLLOGON 00268 at this point, and any CMS command may be issued.

CLLOGON 00271 ## Logout Procedures

CLLOGON 00272 When the user has finished using the system and wishes to
CLLOGON 00273 end his terminal session, he should do so by logging out
CLLOGON 00274 from CP. If the user is not already in the CP environment
CLLOGON 00275 at the time he wishes to log out, he may enter this
CLLOGON 00276 environment by pressing the ATTN key (or its equivalent)
CLLOGON 00276 twice. The keyboard is unlocked and the user types LOGOUT,
CLLOGON 00278 followed by a carriage return. The system responds with

CLLOGON 00280 CONNECT=hh:mm:ss VIRTCPU=mm:ss.hs TOTCPU=mm:ss.hs
CLLOGON 00282 LOGOFF AT hh:mm:ss zone weekday mm/dd/yy

CLLOGON 00286 and the connection to the computer is broken if on a
CLLOGON 00287 switched line. The connect time is in hours, minutes, and
CLLOGON 00287 seconds; the virtual CPU, and total CPU times are in
CLLOGON 00288 minutes, seconds, and hundredths of a second. The logout
CLLOGON 00289 procedure is then completed, and the user may turn the
CLLOGON 00290 terminal power off.

CLLOGON 00291 If the user desires to end the terminal session, but not
CLLOGON 00292 lose the switched line connection with the computer (so that
CLLOGON 00293 another user may log in from the terminal), he types

CLLOGON 00296 LOGOUT HOLD

CLLOGON 00299 If the word HOLD is keyed in followed by pressing the RETURN
CLLOGON 00301 key (or its equivalent), CP does not disconnect the terminal
CLLOGON 00301 from the VM/370 computer, but will type out the "VM/370
CLFUNCSM 00001 online" message again as in Step 2 of the login procedure.

CLFUNCSM 00006 ## FUNCTIONS OF VM/370 COMMANDS

CLFUNCSM 00007 Figures 6 through 14 present a functional summary of
CLFUNCSM 00009 commands available in the VM/370 system. They are described
CLFUNCSM 00010 in detail in the following sections.

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Begin terminal session (identify user to VM/370 system).	LOGIN		CP
End terminal session.	LOGOUT		CP
Communicate with other VM/370 users and with the system operator.	MSG		CP
Connect a terminal to a multi-access virtual machine.	DIAL		CP
Disconnect a user's terminal from a virtual machine.	DISCONN		CP
Test terminal hardware.	ECHO		CP
Control terminal input and output.			
• Indicate if accounting data is to be received at the terminal.	SET	ACNT	CP
• Indicate if messages from other users are to be received at the terminal.	SET	MSG	CP
	SET	WNG	CP
• Control line editing functions.	SET	LINEDIT	CP
• Control format of messages received at the terminal.	SET	RECTYPE	CMS
	SET	EMSG	CP
• Get information about terminal control parameters.	QUERY	TERMINAL	CP
	QUERY		CMS

Figure 6. Commands to Control Terminal Session (Part 1 of 2)

CLFUNCSM 00070
CLFUNCSM 00071
CLFUNCSM 00072
CLFUNCSM 00073
CLFUNCSM 00074
CLFUNCSM 00075
CLFUNCSM 00076
CLFUNCSM 00077
CLFUNCSM 00078
CLFUNCSM 00079
CLFUNCSM 00080
CLFUNCSM 00081
CLFUNCSM 00082
CLFUNCSM 00083
CLFUNCSM 00084
CLFUNCSM 00085
CLFUNCSM 00086
CLFUNCSM 00087
CLFUNCSM 00088
CLFUNCSM 00089
CLFUNCSM 00090
CLFUNCSM 00091
CLFUNCSM 00092
CLFUNCSM 00093
CLFUNCSM 00094
CLFUNCSM 00095

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
• Specify method of password entry.	TERMINAL	MASK	CP
• Specify use of additional translation tables.	TERMINAL	API	CP
• Specify terminal line size.	TERMINAL	LINESIZE	CP
• Specify ATTN key handling procedures.	TERMINAL	ATTN	CP
	SET	RUN	CP
• Specify characters to indicate CPU time interval reporting.	SET	BLIP	CMS
• Specify format of CMS READY message.	SET	RDYMSG	CMS
• Indicate character translations to be done during terminal input and output.	SET	INPUT	CMS
	SET	OUTPUT	CMS

CLFUNCSM 00097

Figure 6. Commands to Control Terminal Session (Part 2 of 2)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Create a source program file from the terminal.	EDIT		CMS
Invoke the System Assembler to assemble a source program.	ASSEMBLE		CMS
Invoke the BASIC Compiler.	BASIC		CMS
Create macro libraries to be used during assemblies or compilations.	MACLIB		CMS
Create subroutine libraries.	TXTLIB		CMS
Specify macro libraries to be searched during assemblies or compilations.	GLOBAL	MACLIB	CMS
Specify subroutine libraries to be searched during LOAD and INCLUDE functions.	GLOBAL	TXTLIB	CMS
Bring object code into main storage.	LOAD		CMS
	INCLUDE		CMS
Create a MODULE (core-image) file.	GENMOD		CMS
Bring MODULE files into main storage.	LOADMOD		CMS

Figure 7. Commands to Develop Programs and Process Data (Part 1

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Print a storage map of a MODULE file.	MODMAP		CMS
Build auxiliary module directories.	GENDIRT		CMS
Begin execution of programs which were previously loaded into main storage.	START		CMS
Simulate OS Data Definition (DD) JCL cards during program execution.	FILEDEF		CMS
Load and execute object (TEXT) files.	RUN		CMS
Compile, load, and execute source files.	RUN		CMS
Load and execute core-image (MODULE) files.	RUN		CMS

Figure 7. Commands to Develop Programs and Process Data (Part 2 of 2

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Stop execution at a specified virtual machine location.	ADSTOP		CP
	DEBUG	BREAK	CMS
Resume execution of a stopped virtual machine.	BEGIN		CP
	DEBUG	GO	CMS
Display virtual storage, registers, PSW, etc.	DISPLAY		CP
	DEBUG	GPR	CMS
	DEBUG	PSW	CMS
	DEBUG	X	CMS
Print the contents of virtual storage locations on the spooled printer.	DUMP		CP
	DEBUG	DUMP	CMS
Change the contents of registers and storage locations.	STORE		CP
	DEBUG	STORE	CMS
	DEBUG	SET	CMS
Trace virtual machine SVC calls.	SVCTRACE		CMS
Trace virtual machine instructions, I/O operations, SVC calls, etc.	TRACE		CP
Convert system ABEND dumps to printed output.	VDUMP*		CMS

*Refer to IBM Virtual Machine Facility/370 Operator's Guide, (Order No. GC20-1806) for a discussion of this command.

Figure 8. Commands to Test and Debug a Program

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Create a file from terminal input.	EDIT		CMS
Create a file from card input.	READCARD		CMS
	DISK	LOAD	CMS
Verify the existence of a file on disk.	STATE		CMS
Erase a file (or files) from disk.	ERASE		CMS
	ACCESS	ERASE	CMS
List names of files on disk and their attributes.	LISTFILE		CMS
Type the contents of a file at the terminal.	TYPE		CMS
Print the contents of a file or a member of a library on a spooled printer.	PRINT		CMS
Punch the contents of a disk file on a spooled punch.	PUNCH		CMS
	DISK	DUMP	CMS
Sort the records of a file into ascending order based on specified sort fields.	SORT		CMS

Figure 9. Commands to Maintain Data Files (Part 1 of 4)

CLFUNCSM 00278
CLFUNCSM 00279
CLFUNCSM 00280
CLFUNCSM 00281
CLFUNCSM 00282
CLFUNCSM 00283
CLFUNCSM 00284
CLFUNCSM 00285
CLFUNCSM 00286
CLFUNCSM 00287
CLFUNCSM 00288
CLFUNCSM 00289
CLFUNCSM 00290
CLFUNCSM 00291
CLFUNCSM 00292
CLFUNCSM 00293
CLFUNCSM 00294
CLFUNCSM 00295
CLFUNCSM 00296
CLFUNCSM 00297
CLFUNCSM 00298
CLFUNCSM 00299
CLFUNCSM 00300
CLFUNCSM 00301
CLFUNCSM 00302
CLFUNCSM 00303
CLFUNCSM 00304
CLFUNCSM 00305
CLFUNCSM 00306
CLFUNCSM 00307
CLFUNCSM 00308
CLFUNCSM 00309
CLFUNCSM 00310
CLFUNCSM 00311
CLFUNCSM 00312
CLFUNCSM 00313
CLFUNCSM 00314
CLFUNCSM 00315
CLFUNCSM 00316
CLFUNCSM 00317

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Copy one disk file to another disk file.	MOVEFILE		CMS
	COPYFILE		CMS
Combine several files into one file.	COPYFILE		CMS
Copy data from one device type to another device type.	MOVEFILE		CMS
Change the name of a CMS file.	RENAME		CMS
Compress a file by encoding multiple contiguous occurrences of a single character.	COPYFILE	PACK	CMS
Rearrange the contents of records in a disk file.	COPYFILE		CMS
Perform character translations on specified characters in a disk file.	COPYFILE	TRANS	CMS
Append one file to the end of another file.	COPYFILE	APPEND	CMS
Remove trailing blanks from records in a file.	COPYFILE	TRUNC	CMS

CLFUNCSM 00319

Figure 9. Commands to Maintain Data Files (Part 2 of 4)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Compare the contents of two disk files.	COMPARE		CMS
Change records in a file based on record sequence numbers.	UPDATE		CMS
Change attributes of a spooled file.	CHANGE		CP
Terminate spooling operations on a virtual unit record device.	CLOSE		CP
Load the virtual forms control buffer (FCB).	LOADVFCB		CP
Indicate order of processing for spooled files.	ORDER		CP
Remove spooled files from the system.	PURGE		CP
Change options in effect for spooling operations.	SPOOL		CP
Route spooled input files to another user.	TRANSFER		CP

Figure 9. Commands to Maintain Data Files (Part 3 of 4)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Copy data from tape to disk and from disk to tape.	TAPE		CMS
Convert OS partitioned data set (PDS) files on tape to CMS format on disk.	TAPPDS		CMS

Figure 9. Commands to Maintain Data Files (Part 4 of 4)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Logically connect a disk to a virtual machine	LINK		CP
Logically disconnect a device from a virtual machine.	DETACH		CP
Make files on a disk available to a user.	ACCESS		CMS
Remove accessibility to files.	RELEASE		CMS
Dump a disk to tape.	DDR	DUMP	CMS
Restore a disk from tape.	DDR	RESTORE	CMS
Format disk space in CMS format.	FORMAT		CMS
Format real or virtual disk volumes for OS or DOS use.	MINIDASD*		CMS

*Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1806) for a descussion of this command.

Figure 10. Commands for Disk Maintenance and Control

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Load a virtual machine operating system.	IPL		CP
Alter the virtual machine configuration.	DEFINE		CP
Enter control program commands from a CMS virtual machine.	CP		CMS
Communicate with other virtual machine users or with the system operator.	MSG		CP
Simulate an external interrupt for a virtual machine.	EXTERNAL		CP
Simulate 'not ready' for a virtual device.	NOTREADY		CP
Simulate functions of buttons on the main computer console.	SYSTEM		CP
Place virtual machine console in dormant state with keyboard locked	SLEEP		CP
Perform tape rewind action.	REWIND		CP
	TAPE	REW	CMS
Establish User Directory Entries.	DIR*		CMS

*Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1806) for a discussion of this command.

Figure 11. Commands for Virtual Machine Control (Part 1 of 3)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Simulate device end interrupt to a virtual machine device.	READY		CP
Reset pending interrupts for virtual devices.	RESET		CP
Define means of entering CMS command names	SYNCSYN		CMS
	SET	ABBREV	CMS
	SET	IMPEX	CMS
	SET	IMPCP	CMS

Figure 11. Commands for Virtual Machine Control (Part 2 of 3)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Indicate if I/O is to be done as specified by the virtual machine with no CCW translation by CP.	SET	NOTRANS	CP
Control timer updating.	SET	TIMER	CP
Change or set the number of loader tables for a CMS virtual machine.	SET	LDRTBLS	CMS
Indicate if pages of storage are to be released after the execution of certain CMS commands.	SET	RELPAGE	CMS
Get information about virtual machine status.	QUERY		CP
	QUERY		CMS

Figure 11. Commands for Virtual Machine Control (Part 3 of 3)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Create accounting records for logged on users and reset accounting data.	ACNT		CP
Logically connect or dedicate devices to a virtual machine or to CP.	ATTACH		CP
Logically disconnect devices from a virtual machine or from CP.	DETACH		CP
Disable communication lines.	DISABLE		CP
Enable communication lines.	ENABLE		CP
Force a specific user to log out from CP.	FORCE		CP
Terminate active channel program on a specific device.	HALT		CP

Figure 12. Commands for VM/370 Control (Part 1 of 2)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Control paging activity.	LOCK UNLOCK		CP CP
Request information about real and virtual machine characteristics.	QUERY		CP
Establish system parameters.	SET		CP
Terminate CP functions and checkpoint system.	SHUTDOWN		CP
Transmit high priority messages to users.	WNG		CP

Figure 12. Commands for VM/370 Control (Part 2 of 2)

CLFUNCSM 00602
CLFUNCSM 00603
CLFUNCSM 00604
CLFUNCSM 00605
CLFUNCSM 00606
CLFUNCSM 00607
CLFUNCSM 00608
CLFUNCSM 00609
CLFUNCSM 00610
CLFUNCSM 00611
CLFUNCSM 00612
CLFUNCSM 00613
CLFUNCSM 00614
CLFUNCSM 00615
CLFUNCSM 00616
CLFUNCSM 00617
CLFUNCSM 00618
CLFUNCSM 00619
CLFUNCSM 00620
CLFUNCSM 00621
CLFUNCSM 00622
CLFUNCSM 00623
CLFUNCSM 00624
CLFUNCSM 00625
CLFUNCSM 00626
CLFUNCSM 00627
CLFUNCSM 00628
CLFUNCSM 00629
CLFUNCSM 00630
CLFUNCSM 00631
CLFUNCSM 00632

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Restart or reposition the current output of an output spooling device.	BACKSPAC		CP
Change attributes of a closed spool file.	CHANGE		CP
Halt operations of specified spooling devices following completion of current activity.	DRAIN		CP
Cancel current output on a real unit record device.	FLUSH		CP
Cancel spool HOLD status.	FREE		CP
Defer spooled output of a particular user.	HOLD		CP
Load printer UCS or FCB buffer.	LOADBUF		CP

CLFUNCSM 00634

Figure 13. Commands for Spooling Control (Part 1 of 2)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Cause spooled files to be processed in a specific order.	ORDER		CP
Remove closed spool files from the system.	PURGE		CP
Request information about spool files.	QUERY		CP
Repeat printing or punching of current file on a specific output device.	REPEAT		CP
Force a printer to single space output.	SPACE		CP
Start spooling device after draining or changing output class.	START		CP
Direct spooled files to a user's card reader.	TRANSFER		CP

Figure 13. Commands for Spooling Control (Part 2 of 2)

FUNCTION	COMMAND	SUB-COMMAND, OPTION	TYPE
Display real storage at terminal.	DCP		CP
Dump real storage to virtual spooled printer.	DMCP		CP
Find locations of control blocks.	LOCATE		CP
Save virtual machine storage space on disk.	SAVESYS		CP
Perform intensive recording of device activity information.	SET	RECORD	CP
Dump error information which has been recorded by error recording routines.	CPEREP*		CP

* Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1806) for a discussion of this command.

Figure 14. Commands for System and Hardware Analysis

CLCMSINT 00005 # CMS COMMANDS

CLCMSINT 00008 ## COMMAND SUMMARY

CLCMSINT 00010 This section contains descriptions of the commands acceptable
CLCMSINT 00012 CMS environment. Figure 6 contains an alphabetical list
CLCMSINT 00014 commands and functions performed by each of the commands.

COMMAND	USAGE
ACCESS	define direct access space to a CMS virtual machine, and relate the disk space to a logical directory.
ASSEMBLE	assemble Assembler language source code
BASIC	compile and execute VM/370 BASIC programs
*COBOL	compile ANS Version 4 COBOL source code
COMPARE	compare all or part of records in two existing files
*CONVERT	convert free form FORTRAN statements to fixed form
COPYFILE	copy files according to specifications
CP	enter CP commands from CMS environment
**CPEREP	dump error information which has been recorded by VM/370 error recording routine
DDR	perform backup, restore, and copy operations on minidisks
DEBUG	enter DEBUG subenvironment

* These commands are used to invoke processors which are IBM Program Products. They are available through IBM for a license fee.

** Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1806) for a discussion of this command.

CLCMSINT 00068 Figure 15. CMS Command Summary (Part 1 of 4)

COMMAND	USAGE
**DIR	set up User Directory entries
DISK	perform disk-to-card and card-to disk operations for CMS data sets
EDIT	enter EDIT subenvironment
ERASE	delete files from user disks
EXEC	process special procedures made up of frequently-used sequences of commands
FILEDEF	provide simulation of OS JCL data definition (DD) statements
FORMAT	prepare disks in CMS 800-byte block format
*FORTGI	compile FORTRAN source code using GI compiler
*FORTHX	compile FORTRAN source code using H-extended compiler
GENDIRT	create auxiliary module directories
GENMOD	generate absolute non-relocatable file (MODULE files)
GLOBAL	define CMS libraries to be searched for macros and subroutines
*GOFORT	compile FORTRAN source code and execute program just compiled using Code and Go compiler
INCLUDE	bring additional TEXT files into storage
LISTFILE	list information about user CMS files

* These commands are used to invoke processors which are IBM Program Products. They are available through IBM for a license fee.

** Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1906) for a discussion of this command.

Figure 15. CMS Command Summary (Part 2 of 4)

COMMAND	USAGE
LOAD	bring TEXT files into storage and establish linkages
LOADMOD	bring a single MODULE file into storage
MACLIB	perform maintenance on macro libraries
**MINIDASD	format disk volumes in DOS or OS format
MODMAP	type load map of a MODULE file
MOVEFILE	move data from one device to another device of the same or different type
*PLIOPT	compile PL/I source code (using optimizing compiler)
PRINT	spool a specified file to the printer
PUNCH	spool a specified file to the punch
QUERY	request information about the virtual machine
READCARD	read data from spooled card input device
RELEASE	make a disk and its directory inaccessible to a virtual machine
RENAME	change the name of a CMS file or files
RUN	initiate series of functions to be performed on a file
SET	establish, set, or reset virtual machine characteristics

* These commands are used to invoke processors which are IBM Program Products. They are available through IBM for a license fee.

** Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1806) for a discussion of this command.

Figure 15. CMS Command Summary (Part 3 of 4)

COMMAND	USAGE
SORT	arrange a specified file in ascending order according to specified fields in the data record
START	begin execution of programs previously loaded
STATE	verify the existence of a file
SVCTRACE	record information about supervisor calls
SYNONYM	specify alternate names by which certain commands may be invoked
TAPE	performs tape-to-disk and disk-to-tape operations for CMS data sets
TAPPDS	load OS partitioned data set (PDS) files from tape to disk
TXTLIB	perform maintenance on text libraries
TYPE	type all or part of a file at the terminal
UPDATE	make changes in a file as defined by control cards in a second file
**VDUMP	convert system ABEND dumps to printed output

** Refer to IBM Virtual Machine Facility/370 Operator's Guide (Order No. GC20-1806) for a discussion of this command.

Figure 15. CMS Command Summary (Part 4 of 4)

Any of the commands in Figures 24-26 may be entered when the user is running CMS in his virtual machine and the terminal is idle and the keyboard is unlocked. If the keyboard is locked (a command is executing) the user may press the ATTN key (or its equivalent) and when the Attention interrupt occurs, enter the command. It will not be executed until the command that is currently executing completes.

In additions to the commands listed in Figures 24-26, there are four commands called Immediate Commands which are handled in a different manner from the others. They may be entered while another command is executing by pressing the ATTN key (or its equivalent) and will be executed immediately. The Immediate Commands are:

CLCMSINT 00247 • HO - Halt tracing

CLCMSINT 00249 • HT - Halt typing

CLCMSINT 00251 • HX - Halt execution

CLCMSINT 00253 • RT - Resume typing

CLCMSERR 00007 CMS Error Messages and Responses

CLCMSERR 00009 The responses from each CMS command are shown following the
CLCMSERR 00009 command discussion. Although it is not stated in the
CLCMSERR 00010 response section, the READY message is received after each
CLCMSERR 00011 command has finished execution. The format of the READY
CLCMSERR 00012 message is:

CLCMSERR 00014 R; T=x.xx/y.yy hh:mm:ss

CLCMSERR 00018 where x.xx is the virtual CPU time used since the last
CLCMSERR 00019 READY message was issued.
CLCMSERR 00021 y.yy is the total CPU time used since the last
CLCMSERR 00022 READY message was issued.

CLCMSERR 00024 and hh:mm:ss is the current real clock time in
CLCMSERR 00025 hours:minutes:seconds.

CLCMSERR 00028 The user may cause the READY message to be printed as R; by
CLCMSERR 00029 issuing the command:

CLCMSERR 00037 ┌──────────────────────┐
CLCMSERR 00038 | SET RDYMSG SMSG |
CLCMSERR 00039 └──────────────────────┘

CLCMSERR 00047 Refer to IBM Virtual Machine Facility/370, System Messages
CLCMSERR 00048 (Order No. GC20-1808) for a description of the messages
CLCMSERR 00049 produced by the CMS commands. All messages from CMS commands
CLACCESS 00001 have the prefix DMS.

CLACCESS 00012 **ACCESS**

CLACCESS 00014 The ACCESS command acquires disk space for the user and sets
CLACCESS 00015 up user file directories in main storage to be used during a
CLACCESS 00016 terminal session.

CLACCESS 00019
CLACCESS 00020 | ACCESS | {[ccu mode[/ext [fn [ft [fm]]]]] [(NOPROF|ERASE)] }|
CLACCESS 00021 | | [(NODISK) }|
CLACCESS 00022

CLACCESS 00027 ccu specifies the device address of the disk to be
CLACCESS 00029 made available to the terminal user. This field
CLACCESS 00030 must be specified if parameters other than the
CLACCESS 00031 default values are desired unless the NODISK
CLACCESS 00032 option is selected. The default value is 191.

CLACCESS 00034 mode specifies the 1 character mode letter which is to
CLACCESS 00035 be associated with the disk being accessed. This
CLACCESS 00035 field must be specified if ccu is specified unless
CLACCESS 00036 the NODISK option is selected. The default value
CLACCESS 00037 is A.

CLACCESS 00039 ext indicates the mode of the parent disk for which
CLACCESS 00040 the file directory of this disk is to be a
CLACCESS 00041 read/only extension. There must not be a blank
CLACCESS 00042 preceding or following the slash (/).

CLACCESS 00044 fn specifies the filename of the files to be selected
CLACCESS 00045 for inclusion in the file directory of the disk
CLACCESS 00046 being accessed. An asterisk coded in this field
CLACCESS 00046 means that all filenames are to be used in file
CLACCESS 00048 selection. See note 6.

CLACCESS 00050 ft specifies the filetype of the files to be selected
CLACCESS 00051 for inclusion in the file directory of the disk
CLACCESS 00051 being accessed. If an asterisk is coded in this
CLACCESS 00053 field, all filetypes are used in file selection.
CLACCESS 00054 See note 6.

CLACCESS 00056 fm specifies the filemode of the files to be selected
CLACCESS 00057 for inclusion in the file directory of the disk
CLACCESS 00057 being accessed. An asterisk coded in this field
CLACCESS 00058 means that all files on the disk specified by the
CLACCESS 00058 mode letter and device address are selected,
CLACCESS 00060 regardless of mode number. See note 6.

CLACCESS 00063 Note: An asterisk(*), preceded by any number of characters
CLACCESS 00064 for filename or filetype allows the specified characters to
CLACCESS 00065 be used as the leading characters for that identifier. For

example, ABC* for fn allows access to all files with filenames beginning with ABC.

Options

NOPROF specifies that the user's PROFILE EXEC file (see Note 1.) is not to be executed. This option is valid only if the command is the first one entered after an IPL of the CMS system. On subsequent ACCESS commands, this option is ignored.

ERASE specifies that the file directory created for the disk being accessed is to contain no entries, and the space previously used by files on that disk is to be available for new files. This option is valid only for disks in read/write mode. This does not alter the file directory on the disk but only the directory in main storage (see Note 4).

NODISK specifies that the user is to gain access to the CMS system without a user disk. This option is valid only in the first command after an IPL. If it is specified, no other parameters or options may be used.

Notes

1. The PROFILE EXEC file is an EXEC procedure which may perform initialization and set system parameters required by the user's processing. For example, if the user is assembling or compiling programs which use macros, the PROFILE EXEC procedure can issue the GLOBAL command to indicate where the system should search for the macros; if he is running OS programs, the PROFILE EXEC procedure can set up 'FILEDEF statements for the files he will be using.

2. If an ACCESS command is not entered as the first command after an IPL CMS command has been performed, the following command:

```
ACCESS 191 A
```

is automatically performed and the user's PROFILE EXEC file, if one exists, is executed.

3. The disk being accessed must have previously been formatted using the CMS command FORMAT. In order to format an unformatted disk, the user must first issue ACCESS (NODISK) then FORMAT ccu mode after IPLing.

4. If an ACCESS command with the ERASE option is entered by mistake, the user may regain access to the existing files on the disk either by issuing another ACCESS command without the ERASE option, or by issuing a RELEASE command for the disk. This is possible since the ERASE option does not alter the disk directory or disk files, but specifies that the directory built in main storage is to contain no entries. However, if a CMS command is executed which causes a new file to be written on the disk, the disk directory is updated and the files which were on the disk before the execution of the ACCESS command with ERASE option are no longer available.

5. If the initial ACCESS command (either explicit or implied) encounters an error, a message is typed at the terminal and the keyboard unlocks to accept another command; no disk has been accessed.

6. If a fileid is specified for a read/write disk that is accessed read/write, all files on the disk will be made available. A fileid can only be specified with disks which are accessed as read/only extensions. Files with a mode number of 0 will not be accessed.

Example 1

```
┌──────────────────────────────────┐
│ ACCESS   192   B                 │
│                                  │
└──────────────────────────────────┘
```

Action: this command makes available to the user for reading, all files on 192 (which is to become the user's B disk).

Example 2

```
┌──────────────────────────────────┐
│  AC      192    B/B ABC          │
│                                  │
└──────────────────────────────────┘
```

Action: this command provides read/only access to all files on the B disk with a filename of ABC.

Example 3

```
ACC    192    B/B       ABC*
```

Action: this command provides read/only access to the user to all files on the B disk whose filenames have ABC as the first three characters.

Example 4

```
ACCE    192    B/B   ABC   *   B2
```

Action: the user has read/only access to all files on the B disk that have a filename of ABC and are in read/only mode.

Example 5

```
ACCES      191    A/A
```

Action: the A disk is placed in read/only mode.

Example 6

```
ACCESS     194      D/A
```

Action: the D disk is made a read/only extension of the A disk. This changes the standard disk search order to

A, D, B, C, E, F, G, S, Y, Z

The A disk must have been accessed before this command is issued.

The effect of this command may be negated by entering the following command.

```
ACCESS    194    D
```

Example 7

```
ACC    (NOPROF)
```

Action: if this command is the first one entered after loading CMS, disk 191 is accessed as the user's A disk, and the user's PROFILE EXEC file is not executed.

Example 8

```
AC  191    A    (ERASE)
```

Action: disk 191 is made available to the user as his A disk, but the file directory built contains no entries. All space on the disk is now available for new files.

Responses

DMSACC723I 'mode (ccu)' R/O

The specified disk is attached to the CMS virtual machine in read/only mode.

DMSACC724I 'cuu1' REPLACES 'mode (cuu2)'

```
CLACCESS 00283    Before execution of the command the disk represented by ccu2
CLACCESS 00283    was the user's 'mode' disk.  ccu1  is now assigned that mode
CLACCESS 00285    letter. This message will be followed by message DMSACC726I.

CLACCESS 00287    DMSACC725I  'ccu' ALSO = 'mode' DISK

CLACCESS 00289    The disk specified  by ccu is the user's 'mode'  disk and he
CLACCESS 00289    has  issued an  ACCESS  command to  assign  it another  mode
CLACCESS 00290    letter.

CLACCESS 00294    DMSACC726I  'ccu mode [/ext]' RELEASED

CLACCESS 00295    The  disk  located at  virtual  address  ccu that  is  being
CLACCESS 00296    accessed as a read/write disk is already currently accessed.
CLACCESS 00297    The effect of  the previous ACCESS command  is cancelled and
CLACCESS 00298    the  disk  is  released  from  the  user's  virtual  machine
CLASSEMB 00001    configuration.
```

ASSEMBLE

The ASSEMBLE command invokes the VM/370 System Assembler to assemble the specified file. Assembler processing and output are controlled by the options selected.

```
┌──────────────────────────────────────────────────────┐
│ ASSEMBLE │ fn     [(option...)]                       │
│ A        │                                            │
│          │        options:  [XREF|NOXREF]             │
│          │                  [RENT|NORENT]             │
│          │                  [DECK|NODECK]             │
│          │                  [LOAD|NOLOAD]             │
│          │                  [ALGN|NOALGN]             │
│          │                  [OS|DOS]                  │
│          │                  [TEST|NOTEST]             │
│          │                  [LINECNT (nn|55)]         │
│          │                  [PRINT|DISK|NOPRINT]      │
│          │                  [TERM|NOTERM]             │
│          │                  [NUM|NONUM]               │
│          │                  [STMT|NOSTMT]             │
│          │                  [LIST|NOLIST]             │
└──────────┴─────────────────────────────────────────────┘
```

fn specifies the filename of the source file to be assembled. The file must have a filetype of ASSEMBLE and fixed-length, 80-character records.

Options

If duplicate or conflicting options are specified, the last one entered in the command line will be in effect.

XREF includes a cross-reference symbol table in the LISTING file.

NOXREF suppresses the cross-reference symbol table.

RENT a check for satisfaction of reentrancy requirements is made of the source file.

NORENT suppresses the check for satisfaction of reentrancy requirements.

DECK the output object module, fn TEXT, is spooled to the punch.

NODECK the output object module, fn TEXT, is not spooled to the punch.

LOAD a TEXT file is created on disk for the program that was assembled.

```
CLASSEMB 00077     NOLOAD     a TEXT file is not created on disk for the program
CLASSEMB 00079                that was assembled.

CLASSEMB 00082     ALGN       does not suppress alignment error diagnostic
CLASSEMB 00083                messages.

CLASSEMB 00087     NOALGN     suppresses alignment error diagnostic messages.

CLASSEMB 00090     LIST       produces a LISTING file.

CLASSEMB 00093     NOLIST     produces no LISTING file.

CLASSEMB 00095     OS         the assembler will produce OS compatible object
CLASSEMB 00096                programs.

CLASSEMB 00098     DOS        the Assembler will behave like DOS Assemblers D
CLASSEMB 00100                and F and produce DOS compatible object programs.
CLASSEMB 00101                If the DOS option is specified, the LOAD, TEST,
CLASSEMB 00102                and NOALGN options must not be specified; they are
CLASSEMB 00104                not compatible with the DOS option.

CLASSEMB 00105                No DOS macro libraries are provided with VM/370.
CLASSEMB 00107                DOS object programs will not execute under CMS.

CLASSEMB 00110     TEST       generates a symbol table as part of the TEXT file.

CLASSEMB 00113     NOTEST     does not generate a symbol table.

CLASSEMB 00116     PRINT      spools the LISTING file to the printer.

CLASSEMB 00119     DISK       puts the LISTING file on disk.

CLASSEMB 00122     NOPRINT    produces no LISTING file.

CLASSEMB 00124     STMT       the internal statement numbers are to appear with
CLASSEMB 00126                the diagnostic messages sent to the terminal. This
CLASSEMB 00127                option is meaningful only when used with TERM.

CLASSEMB 00129     NOSTMT     the internal statement numbers are not to be
CLASSEMB 00130                included.

CLASSEMB 00132     NUM        the sequence numbers assigned to each source
CLASSEMB 00133                statement in the input file are to be typed in
CLASSEMB 00135                diagnostic messages to the terminal. This option
CLASSEMB 00136                is meaningful only when used with TERM.

CLASSEMB 00139     NONUM      the sequence numbers are not included.

CLASSEMB 00142     TERM       specifies that diagnostic messages are to be typed
CLASSEMB 00143                at the terminal.

CLASSEMB 00146     NOTERM     no diagnostic messages are to be typed.
```

LINECNT nn|55
specifies the number of lines per output printer page. A default of 55 lines is assumed.

Files Created by the ASSEMBLE Command

Temporary Workfiles:

fn SYSUT1
fn SYSUT2
fn SYSUT3

These files are temporarily created for each assembly; any existing files with the same file identifiers are erased at the beginning of the assembly. These files are placed on the read/write disk with the most available space. Work space is automatically allocated as needed during the assembly and returned to available status when the assembly is complete. Insufficient space causes abnormal termination of the assembly.

Permanent files:

fn TEXT
fn LISTING

fn TEXT contains the output object module if the LOAD option is in effect. fn LISTING contains a listing of source statements, assembled machine code, and other associated information based on the options selected. This file is created unless the NOPRINT or NOLIST options are selected. The LISTING and TEXT files are placed on (1) the disk from which the source file was read, (2) its parent disk, or (3) the primary disk, unless the user has created a file definition for these files placing them on a non-DASD device. Failure to obtain sufficient space for these files results in abnormal termination of the assembly.

References

For a description of facilities available to users of VM/370 Assembler Language, see the following publication: IBM Virtual Machine Facility/370, Assembler Programmer's Guide (Order No. GC20-1802).

BASIC

The BASIC command invokes the VM/370 BASIC Language Processor to compile and execute the specified program.

```
┌──────────────────────────────────────┐
| BASIC |     fn      [(LONG)]         |
| BA    |                              |
└──────────────────────────────────────┘
```

fn specifies the filename of the file to be compiled and executed. The file must have a filetype of BASIC and contain fixed length 80-character records.

LONG is the option for long form precision. If not specified, short form is assumed.

References

For a description of the BASIC language see the IBM publication IBM Virtual Machine Facility/370, BASIC Language Reference Manual (Order No. GC20-1803).

Files Used by VM/370 BASIC Programs During Execution

Input and output files used during the execution of VM/370 BASIC programs must have a filetype of 'BASDATA', with a filename as designated by the OPEN statement in the program. The record format is the standard BASIC data record format (that is, undefined record, with a maximum record length of 3440 characters).

Responses

For responses from the BASIC processor, refer to the manual already noted under "References".

CLCOMPAR 00003 **COMPARE**

CLCOMPAR 00014 The COMPARE command is used to compare two disk files or
CLCOMPAR 00015 fixed or variable length format and type the contents of
CLCOMPAR 00017 corresponding unlike records at the terminal.

CLCOMPAR 00019
CLCOMPAR 00020 | COMPARE | fileid1 fileid2 [(COL nn-nn)] |
CLCOMPAR 00021 | COM |
CLCOMPAR 00022

CLCOMPAR 00027 fileid specifies the file identification of the two
CLCOMPAR 00028 files to be compared. All three components
CLCOMPAR 00029 (filename, filetype, and filemode) must be
CLCOMPAR 00030 specified for each fileid.

CLCOMPAR 00033 **Options**

CLCOMPAR 00035 COL nn-nn By using this option, the user can define any
CLCOMPAR 00037 1-LRECL contiguous portion of the corresponding records
CLCOMPAR 00038 for comparison. The comparison begins at
CLCOMPAR 00039 position nn of each record in both files. The
CLCOMPAR 00040 comparison proceeds up to and including position
CLCOMPAR 00040 nn of each record in both files. If nn is not
CLCOMPAR 00041 specified the default ending position is the
CLCOMPAR 00042 last character of each record. The "-" is
CLCOMPAR 00042 required and it may not be preceded or followed
CLCOMPAR 00043 by a blank. LRECL is the logical record length
CLCOMPAR 00044 of the file.

CLCOMPAR 00047 Example 1

CLCOMPAR 00049
CLCOMPAR 00050 | COMPARE | ABC XYZ A1 ABC MNO A1 |
CLCOMPAR 00051

CLCOMPAR 00055 **Action**: each record in file ABC XYZ A1 is compared with the
CLCOMPAR 00056 corresponding record in file ABC MNO A1. Comparison begins
CLCOMPAR 00057 at the first position of each record and proceeds for the
CLCOMPAR 00058 entire length of the record. Records which do not match are
CLCOMPAR 00059 typed on the terminal typewriter.

Example 2

```
| COMPARE |MYFILE ASSEMBLE A1 YOURFILE ASSEMELE A1 (COL 10-72)|
```

Action: positions 10-72 of each record in file MYFILE ASSEMBLE A1 are compared with corresponding positions of each record in file YOURFILE ASSEMBLE A1. Records in which these positions do not match are typed on the terminal typewriter.

Responses

If corresponding records in each file do not match, the record from the first file is typed, followed by the record from the second file.

CLCOPYF 00004 **COPYFILE**

CLCOPYF 00006 The COPYFILE command copies data from specified input files
CLCOPYF 00007 to the output files according to conversions and
CLCOPYF 00009 specifications indicated by the options selected. The manner
CLCOPYF 00009 in which the file identifiers are entered determines whether
CLCOPYF 00010 one output file will be created (single output mode) or
CLCOPYF 00012 multiple files will be created (multiple output mode).

CLCOPYF 00014 The COPYFILE command is used to:

CLCOPYF 00019 • copy one file to another
CLCOPYF 00021 • combine two or more files into a single output file
CLCOPYF 00023 • copy files into multiple output files

CLCOPYF 00025 Within these copy operations, the user can specify that the
CLCOPYF 00027 following actions are to be performed during the operation:

CLCOPYF 00030 • names of files copied may be typed at the terminal.
CLCOPYF 00031 • existing output files may be replaced by the new output
CLCOPYF 00032 files.
CLCOPYF 00033 • the record format and logical record length may be
CLCOPYF 00034 changed.
CLCOPYF 00036 • records from the input file(s) may be selectively copied
CLCOPYF 00037 based on either:
CLCOPYF 00041 a. record number.
CLCOPYF 00043 b. label field of record.
CLCOPYF 00045 • trailing fill characters may be removed from each
CLCOPYF 00046 record.
CLCOPYF 00048 • an input file may be compressed.
CLCOPYF 00049 • 026 key punch characters may be converted to
CLCOPYF 00051 corresponding 029 characters.
CLCOPYF 00052 • lowercase letters may be converted to uppercase and
CLCOPYF 00054 uppercase letters may be converted to lowercase.
CLCOPYF 00056 • data in an existing file may be overlaid with selected
CLCOPYF 00057 data from another file.
CLCOPYF 00059 • one file may be appended to another file.
CLCOPYF 00061 • selected positions within each record many be moved to
CLCOPYF 00062 specified positions in the records of another file.
CLCOPYF 00064 • a specified character string or hexadecimal character
CLCOPYF 00065 may be inserted into selected positions of each record
CLCOPYF 00066 in the output file.
CLCOPYF 00068 • character translations may be performed.

```
CLCOPYF  00083
CLCOPYF  00084    | COPYFILE | fileidi1 [fileidi2... ] [fileido] [(options)]|
CLCOPYF  00085    | COPY     |                                                |
CLCOPYF  00086    |          |  options:  [TYPE|NOTYPE]                       |
CLCOPYF  00087    |          |            [NEWDATE|OLDDATE]                   |
CLCOPYF  00088    |          |            [RECFM (F|V) ]                      |
CLCOPYF  00089    |          |            [LRECL nn]                          |
CLCOPYF  00090    |          |            [PROMPT|NOPROMPT]                   |
CLCOPYF  00091    |          |            [FROM n'rec|FRLABEL xxxxxxxx]       |
CLCOPYF  00092    |          |            [FOR n'rec|TOLABEL xxxxxxxx]        |
CLCOPYF  00093    |          |            [TRUNC|NOTRUNC]                     |
CLCOPYF  00094    |          |            [PACK|UNPACK]                       |
CLCOPYF  00095    |          |            [EBCDIC]                            |
CLCOPYF  00096    |          |            [UPCASE|LOWCASE]                    |
CLCOPYF  00097    |          |            [NEWFILE|REPLACE|OVLY|APPEND]      |
CLCOPYF  00098    |          |            [FILL (c|nn|40)]                    |
CLCOPYF  00099    |          |            [SPECS|NOSPECS]                     |
CLCOPYF  00100    |          |            [TRANS]                             |
CLCOPYF  00101    |__________|________________________________________________|
```

CLCOPYF 00106 **fileidi1** specifies the first (or only) input file.
CLCOPYF 00107 Filename, filetype and filemode must all be
CLCOPYF 00108 specified either by indicating a specific name
CLCOPYF 00109 or by coding an asterisk. Note, however, that
CLCOPYF 00110 all three components of fileidi1 cannot be
CLCOPYF 00111 specified by asterisks.

CLCOPYF 00115 **fileidi2** specifies additional input files. Filename,
CLCOPYF 00115 filetype and filemode must all be specified. In
CLCOPYF 00116 single ouput filemode, any of the three
CLCOPYF 00117 components may be specified by an asterisk while
CLCOPYF 00117 in multiple output filemode, an asterisk is
CLCOPYF 00118 illegal in the field. An equal (=) sign may be
CLCOPYF 00118 coded for any (or any part of) of the fileid
CLCOPYF 00119 components indicating that the component is to
CLCOPYF 00120 be the same as the corresponding component in
CLCOPYF 00121 fileidi1.

CLCOPYF 00123 **fileido** specifies the output files to be created.
CLCOPYF 00124 Filename, filetype and filemode must all be
CLCOPYF 00125 specified. To create multiple output files, an
CLCOPYF 00125 equal sign (=) must be coded in one or more of
CLCOPYF 00126 the component fields. If there is only one input
CLCOPYF 00127 fileid, fileido may be omitted, in which case it
CLCOPYF 00127 defaults to = = = . This causes the input file
CLCOPYF 00129 represented by fileidi1 to be replaced.

CLCOPYF 00131 The following options are the most used and are useful for
CLCOPYF 00132 most general applications. Minor variations of these
CLCOPYF 00134 options are described under the heading, "Other Options".

```
CLCOPYF  00136    Options

CLCOPYF  00140    TYPE          specifies that the names of the files copied are
CLCOPYF  00144    T             to be typed at the terminal.

CLCOPYF  00147    NOTYPE        specifies that the names of the files to be
CLCOPYF  00151    NOT           copied are not typed at the terminal.

CLCOPYF  00154    NEWDATE       specifies that the creation date of the new
CLCOPYF  00158    NEWD          files is the current date.

CLCOPYF  00161    OLDDATE       specifies that the creation date of the new
CLCOPYF  00165    OLDD          files is the same as that of the first input
CLCOPYF  00166                  file.

CLCOPYF  00172
CLCOPYF  00174    RECFM | F |   specifies the record format of the output
CLCOPYF  00176    REC   < V >   files. If not specified, the output record format
CLCOPYF  00179          L   J   is the same as that of the input.

CLCOPYF  00182    LRECL nn      specifies the logical record length of the
CLCOPYF  00186    LR            output files if it is to be different from that
CLCOPYF  00187                  of the input files. The maximum value of nn is
CLCOPYF  00188                  65535.

CLCOPYF  00191    PROMPT        specifies that the messages which request
CLCOPYF  00192    PR            specification or translation lists are to be
CLCOPYF  00193                  displayed.

CLCOPYF  00196    NOPROMPT      specifies that prompt messages for specification
CLCOPYF  00198    NOPR          and translation lists are not to be displayed.

CLCOPYF  00204    Copy Extent Options

CLCOPYF  00206    FROM recno    specifies the starting record number for each
CLCOPYF  00210    FR            input file in the copy operation.

CLCOPYF  00214    FRLABEL       xxxxxxxx
CLCOPYF  00220    FRL           xxxxxxxx specifies a character sequence which
CLCOPYF  00221                  will appear at the beginning of the first record
CLCOPYF  00221                  to be copied from each input file. Up to eight
CLCOPYF  00222                  characters may be specified. The character
CLCOPYF  00223                  sequence may not contain embedded blanks.

CLCOPYF  00228    FOR n'rec     specifies the number of records to be copied
CLCOPYF  00229                  from each input file.

CLCOPYF  00234    TOLABEL       xxxxxxxx
CLCOPYF  00237    TOL           xxxxxxxx specifies a character sequence which,
CLCOPYF  00238                  if at the beginning of a record, stops the copy
CLCOPYF  00238                  operation for that input file. The record
```

```
CLCOPYF  00239                 containing the given character is not copied.
CLCOPYF  00240                 Up to eight characters may be specified. The
CLCOPYF  00240                 character sequence may not contain embedded
CLCOPYF  00241                 blanks.

CLCOPYF  00244     TRUNC       specifies that trailing blanks (or fill
CLCOPYF  00248     TRU         characters) are to be removed when converting to
CLCOPYF  00249                 RECFM V output file record format.

CLCOPYF  00250                 The RECFM and LRECL options can be used to
CLCOPYF  00251                 specify the record format and logical record
CLCOPYF  00252                 lengths of the output files being created.

CLCOPYF  00253                 There are two record formats, F (fixed) or V
CLCOPYF  00254                 (variable). Fixed record format files have all
CLCOPYF  00254                 records the same size. Typical of such files
CLCOPYF  00255                 are card image files, where all records are 80
CLCOPYF  00256                 bytes long. Variable record format files have
CLCOPYF  00256                 records of varying sizes. Typical of such files
CLCOPYF  00257                 are SCRIPT files, where the length of each
CLCOPYF  00258                 record is usually different.

CLCOPYF  00259                 The logical record length applies only to fixed
CLCOPYF  00260                 record format (F) files. For example, for card
CLCOPYF  00261                 image files, the logical record length is 80.

CLCOPYF  00262                 The default record format and logical record
CLCOPYF  00263                 length are taken from the current or only first
CLCOPYF  00264                 input file, and may be overridden by the RECFM
CLCOPYF  00264                 or LRECL options. The RECFM option is followed
CLCOPYF  00265                 by the letter F or V, and the IRECL option is
CLCOPYF  00265                 followed by a decimal number giving the logical
CLCOPYF  00266                 record length. If the output file record format
CLCOPYF  00267                 is V, then the LRECL option if specified, is
CLCOPYF  00268                 ignored.

CLCOPYF  00269                 When the output file record format is F, then
CLCOPYF  00270                 the input records are truncated or padded, if
CLCOPYF  00271                 necessary, to the logical record length. When
CLCOPYF  00271                 the output file record format is V, the input
CLCOPYF  00272                 records are simply copied to the output file,
CLCOPYF  00273                 unless the TRUNC option is specified. In the
CLCOPYF  00273                 latter case, all blanks on the end of each
CLCOPYF  00274                 record are removed before the record is written
CLCOPYF  00275                 out.

CLCOPYF  00276                 For information on variations of this option,
CLCOPYF  00277                 see the discussion of the FILL option under
CLCOPYF  00278                 "Other Options."

CLCOPYF  00281     NOTRUNC     specifies that trailing blanks (or fill
CLCOPYF  00284     NOTR        characters) are not to be removed when
CLCOPYF  00286                 converting to RECFM V output file record format.
```

CLCOPYF 00290 **PACK** specifies that repetitively occurring
CLCOPYF 00293 **PA** characters are to be converted to compressed
CLCOPYF 00294 format. If the FILL option is not used, all
CLCOPYF 00295 occurrences of two or more blanks in the file
CLCOPYF 00295 are encoded as one character, and four or or
CLCOPYF 00296 more occurrences of any other character in the
CLCOPYF 00297 file are encoded as three characters. If a FILL
CLCOPYF 00297 character is specified, that character replaces
CLCOPYF 00298 the blank as the special packing character, and
CLCOPYF 00299 blanks are treated as any other non-fill
CLCOPYF 00300 character.

CLCOPYF 00301 Source files generally take up a great deal of
CLCOPYF 00302 disk space, largely because they contain a great
CLCOPYF 00302 many blanks. This is particularly wasteful in
CLCOPYF 00303 the case of source files which are seldom
CLCOPYF 00303 needed, but must be kept immediately available
CLCOPYF 00305 for the few times that they are needed.

CLCOPYF 00306 By using the PACK option of the COPYFILE
CLCOPYF 00307 command, the user may cause the system to encode
CLCOPYF 00307 his file in a special way so that multiple
CLCOPYF 00308 blanks are represented as a single character,
CLCOPYF 00308 and multiple occurrences of other characters
CLCOPYF 00309 also produce space savings.

CLCOPYF 00310 If the PACK option is specified, then the output
CLCOPYF 00311 file will be in a special format which can be
CLCOPYF 00311 decoded only by the UNPACK option of the
CLCOPYF 00312 COPYFILE command.

CLCOPYF 00313 As an example of the use of this option, the
CLCOPYF 00314 following command:

CLCOPYF 00317 COPYFILE * ASSEMBLE A1 (PACK

CLCOPYF 00319 causes all ASSEMBLE files on the A-disk to be
CLCOPYF 00320 packed.

CLCOPYF 00321 In the encoding process, all occurrences of two
CLCOPYF 00322 of more blanks in the file are encoded as one
CLCOPYF 00322 character, while four or more consecutive
CLCOPYF 00323 occurrences of any other character in the file
CLCOPYF 00324 are encoded as three characters. It is clear,
CLCOPYF 00324 therefore, that the greatest savings are made
CLCOPYF 00325 with respect to compressing blank characters.
CLCOPYF 00326 If the user knows that his file contains a
CLCOPYF 00326 preponderance of some other character, far more
CLCOPYF 00327 often then blanks, then he may specify that
CLCOPYF 00327 character with the FILL option, and that
CLCOPYF 00328 character will replace the blank as the "special
CLCOPYF 00329 packing character." Of course, four or more
CLCOPYF 00329 blanks will still be compressed, as will any
CLCOPYF 00330 other character. (The FILL option is discussed

```
CLCOPYF  00331                under "Other Options".)

CLCOPYF  00332                Note, however, that not every file should be
CLCOPYF  00333                packed. In fact, if the file does not contain
CLCOPYF  00333                very many occurrences of multiple characters,
CLCOPYF  00334                then it is possible for the packed file to be
CLCOPYF  00335                longer than the original file.

CLCOPYF  00336                WARNING:  A file in packed format should not be
CLCOPYF  00337                modified or tampered with in any way. If such a
CLCOPYF  00337                file is modified in any way, then the UNPACK
CLCOPYF  00338                routines will be unable to reconstruct the
CLCOPYF  00339                original file. Packed files should never be
CLCOPYF  00340                combined or split apart in any way.

CLCOPYF  00343    UNPACK      specifies that the PACK operation is to be
CLCOPYF  00347    UNP         reversed.


CLCOPYF  00350    Character Translation Options


CLCOPYF  00354    EBCDIC      specifies that the file was created with 026
CLCOPYF  00358    EB          keypunch characters (BCD), and that all such
CLCOPYF  00358                characters are to be converted to C29 keypunch
CLCOPYF  00359                characters (EBCDIC). The following conversions
CLCOPYF  00360                are made:

CLCOPYF  00362                          < to )
CLCOPYF  00364                          & to +
CLCOPYF  00366                          % to (
CLCOPYF  00368                          # to =
CLCOPYF  00370                          @ to '
CLCOPYF  00372                          ' to :

CLCOPYF  00376    UPCASE      specifies that all lowercase characters in each
CLCOPYF  00380    UP          record to be written to an output file are to be
CLCOPYF  00380                converted to uppercase before the record is
CLCOPYF  00381                written out.

CLCOPYF  00384    LOWCASE     specifies that all uppercase characters in each
CLCOPYF  00388    LO          record to be written to an output file are to be
CLCOPYF  00388                converted to lowercase before the record is
CLCOPYF  00389                written out.


CLCOPYF  00393    Other Options


CLCOPYF  00395    In addition to the options already described, the CCPYFILE
CLCOPYF  00396    command offers several more complex variations, which allow
CLCOPYF  00397    more flexibility in the copy operation.
```

```
CLCOPYF  00399     NEWFILE     if this option is in effect, the output files
CLCOPYF  00402     NEWF        must not have previously existed.  If one or
CLCOPYF  00403                 more do exist an error message is typed and the
CLCOPYF  00404                 COPYFILE command terminates.  This option is the
CLCOPYF  00404                 default so that the user cannot inadvertently
CLCOPYF  00406                 destroy an existing file on his disk.

CLCOPYF  00409
CLCOPYF  00410     REPLACE     specifies that only files with the same file
CLCOPYF  00413     REP         identifiers as those being created by the
CLCOPYF  00414                 COPYFILE command are to be erased and new ones
CLCOPYF  00415                 created.  REPLACE is the default option only
CLCOPYF  00416                 when the output file identification is "= = = ".

CLCOPYF  00419     OVLY        specifies that data in an existing output file
CLCOPYF  00423     OV          is to be overlaid with data from the input file.

CLCOPYF  00426     APPEND      specifies that data from the input file is to
CLCOPYF  00429     AP          be appended to the end of a file specified by
CLCOPYF  00430                 the output file identifier.  If no output file
CLCOPYF  00431                 exists, one is created.

CLCOPYF  00434     FILL c      specifies the padding and truncation character
CLCOPYF  00436     FI   hh     or the principal packing character for the
CLCOPYF  00440          40     PACK option.  It may be specified by typing a
CLCOPYF  00440                 single character, 'c', or by typing a two-digit
CLCOPYF  00441                 hexadecimal representation of a character.  The
CLCOPYF  00442                 default is 40 (the hexadecimal representation
CLCOPYF  00443                 for a blank in EBCDIC).

CLCOPYF  00444                 When the output file record format is F, then
CLCOPYF  00445                 the input records are truncated or padded, if
CLCOPYF  00446                 necessary, to the logical record length.  The
CLCOPYF  00446                 padding and truncation character is usually a
CLCOPYF  00447                 blank, but this default may be overridden by
CLCOPYF  00447                 means of the FILL option, to specify a new
CLCOPYF  00449                 padding and truncation character.

CLCOPYF  00450                 When the output file record format is V, then
CLCOPYF  00451                 the input records will simply be copied to the
CLCOPYF  00451                 output file, unless the TRUNC option is
CLCOPYF  00452                 specified.  In the latter case, all blanks on the
CLCOPYF  00453                 end of each record are removed before the record
CLCOPYF  00453                 is written out.  If it is desired to truncate
CLCOPYF  00454                 some character other then blanks, then the FILL
CLCOPYF  00455                 option may be used.

CLCOPYF  00456                 The PACK routine treats the blank as a "special
CLCOPYF  00457                 packing character", and makes the greatest
CLCOPYF  00458                 savings with respect to multiple occurrences of
CLCOPYF  00458                 that character.  If desired, the FILL option may
CLCOPYF  00459                 be used to change the "special packing
CLCOPYF  00460                 character" to any desired character.
```

CLCOPYF 00464 SPECS specifies that the user is to be asked for a
CLCOPYF 00467 SP specification list which defines the manner in
CLCOPYF 00469 which data is to be copied.

CLCOPYF 00470 Use of the SPECS option causes the COPYFILE
CLCOPYF 00471 command to request a specification list that
CLCOPYF 00471 defines the way in which the output records are
CLCOPYF 00472 to be formed.

CLCOPYF 00473 If the SPECS option is used, then CCPYFILE
CLCOPYF 00474 issues the following prompting message:

CLCOPYF 00476 DMSCPY601R ENTER SPECIFICATION LIST

CLCOPYF 00477 The keyboard unlocks, and the user may type in
CLCOPYF 00478 the specification list.

CLCOPYF 00479 The format of the specification list is as
CLCOPYF 00480 follows:

CLCOPYF 00485 source target <source target> ...

CLCOPYF 00486 The source specification may be any one of the
CLCOPYF 00487 following:

CLCOPYF 00492 a. A pair of columns of the input file,
CLCOPYF 00494 specified in the format "nn-mm", two decimal
CLCOPYF 00495 number values separated by a hyphen. This
CLCOPYF 00495 format is used to cause the specified columns
CLCOPYF 00496 of the input file to be copied to the output
CLCOPYF 00497 file. For each record that is copied, the
CLCOPYF 00497 value of mm is compared with the length of
CLCOPYF 00498 the record. If it is larger, then it is
CLCOPYF 00498 truncated to the record length. For example,
CLCOPYF 00499 the source specification "23-40" causes
CLCOPYF 00499 columns 23 through 40 of the input file
CLCOPYF 00501 record to be copied to the output file.

CLCOPYF 00505 b. A string of characters, delimited by
CLCOPYF 00507 non-alphameric characters. Such a
CLCOPYF 00508 specification causes the specified string of
CLCOPYF 00508 characters to be placed in the output file
CLCOPYF 00510 record. For example, a source specification
CLCOPYF 00510 of "/AbCd 1234/" causes the string "AbCd
CLCOPYF 00512 1234" to be placed in each record of the
CLCOPYF 00512 output file. The letters in the string of
CLCOPYF 00513 characters are not automatically raised to
CLCOPYF 00513 uppercase; if the user wishes capital
CLCOPYF 00515 letters, he must type them in as such.

CLCOPYF 00519 c. A string of characters, specified by the
CLCOPYF 00522 letter "H" followed by an even number of
CLCOPYF 00522 hexadecimal digits entered using numeric

```
CLCOPYF  00523          characters and either upper or lower case
CLCOPYF  00524          alphabetic characters.  This type of
CLCOPYF  00524          specification is useful when type "b"
CLCOPYF  00526          described in the preceding paragraph is
CLCOPYF  00526          unusable because the desired string contains
CLCOPYF  00527          characters which cannot be typed.  For
CLCOPYF  00527          example, the specification, "hb7c1ff" causes
CLCOPYF  00527          the characters given by the hexadecimal
CLCOPYF  00528          values X'B7', X'C1', and X'FF' to be placed
CLCOPYF  00529          in the output file.

CLCOPYF  00532       The target specification has only one format: a
CLCOPYF  00533       decimal number value representing a column
CLCOPYF  00534       number in the output record.

CLCOPYF  00535       For example, consider the following
CLCOPYF  00536       specification list:

CLCOPYF  00540            1-5 10 *ABC* 3  *XYZ* 20 H0C 25

CLCOPYF  00541       This specification list contains four sets of
CLCOPYF  00542       specifications, and causes data to be placed
CLCOPYF  00542       into the output record in the following manner:
CLCOPYF  00543       First, columns 1 through 5 of the input record
CLCOPYF  00543       are placed in columns 10 through 14 of the
CLCOPYF  00544       output record.  Next, the characters "ABC" are
CLCOPYF  00545       placed in columns 3 through 5 of the output
CLCOPYF  00545       record.  Next, the characters "XYZ" are placed
CLCOPYF  00546       in columns 20 through 22 of the output record.
CLCOPYF  00546       Finally, the character X'00' is placed in column
CLCOPYF  00548       25 of the output record.

CLCOPYF  00549       If the RECFM of the output file is V, then the
CLCOPYF  00550       length of the output record is determined by the
CLCOPYF  00550       position of the rightmost byte of data that was
CLCOPYF  00551       placed into the output record.  If the RECFM of
CLCOPYF  00551       the output file is F, then the record is padded
CLCOPYF  00553       or truncated to the logical record length.

CLCOPYF  00554       Columns of the output record for which no data
CLCOPYF  00554       is specified are filled with blanks.  If some
CLCOPYF  00555       other character is desired, then the FILL option
CLCOPYF  00555       may be used to specify a different character.
CLCOPYF  00556       In the example shown above, columns 1-2, 4-9,
CLCOPYF  00557       15-19 and 23-24 of the output file record will
CLCOPYF  00557       contain blanks.  In addition, if the output
CLCOPYF  00558       record format is F, then there will be blanks in
CLCOPYF  00559       columns 26 through to the end of the record.

CLCOPYF  00560       The SPECS option is particularly useful in
CLCOPYF  00560       conjunction with the OVLY option.  If these
CLCOPYF  00561       options are used together, then the
CLCOPYF  00561       specification list specifies how the output file
CLCOPYF  00562       record is to be overlaid.  For example, if the
```

CLCOPYF 00562 specification list "1-10 20" were used with the
CLCOPYF 00563 OVLY option, then columns 1-10 of each input
CLCOPYF 00563 file record would overlay columns 20-29 of each
CLCOPYF 00564 record of the existing output file, and the
CLCOPYF 00564 other columns of the output file records would
CLCOPYF 00565 remain unchanged.

CLCOPYF 00566 The specification list can be continued onto
CLCOPYF 00567 additional lines by typing '++' at the end of
CLCOPYF 00567 the first line. When these two characters are
CLCOPYF 00568 encountered when a source-specification is
CLCOPYF 00568 expected, all scanning of that line ceases, and
CLCOPYF 00569 the keyboard unlocks so that a new line may be
CLCOPYF 00570 entered.

CLCOPYF 00573 NOSPECS indicates that the user does not wish to enter a
CLCOPYF 00577 NOSP specification list.

CLCOPYF 00580 TRANS specifies that the user is to be asked for a list
CLCOPYF 00583 TRA of character conversions which are to be made as
CLCOPYF 00584 the file is copied.

CLCOPYF 00585 The TRANS option allows the user to specify his
CLCOPYF 00586 own list of translations. If specified in
CLCOPYF 00587 conjunction with any of the other three
CLCOPYF 00587 translation options (UPCASE, LOWCASE or EBCDIC),
CLCOPYF 00588 then the user's list will override any
CLCOPYF 00588 individual translations specified by those
CLCOPYF 00589 options.

CLCOPYF 00590 If the TRANS option is specified, then the
CLCOPYF 00591 COPYFILE command types out the prompting
CLCOPYF 00592 message:

CLCOPYF 00594 DMSCPY602R ENTER TRANSLATION LIST

CLCOPYF 00595 and unlocks the keyboard. The user may then
CLCOPYF 00596 enter his translation list.

CLCOPYF 00597 The translation list consists of as many pairs
CLCOPYF 00598 of characters as the user desires, separated by
CLCOPYF 00598 blanks. Each character may be specified either
CLCOPYF 00599 by typing the character itself or by typing a
CLCOPYF 00599 two-digit hexadecimal equivalent (the latter is
CLCOPYF 00600 particularly useful for characters which cannot
CLCOPYF 00601 be typed).

CLCOPYF 00602 For example, the following translation list may
CLCOPYF 00603 be typed in:

CLCOPYF 00605 * - A f0 00 ff

CLCOPYF 00606 This translation list specifies that the
CLCOPYF 00607 character '*' is to be translated to '-', the

CLCOPYF 00607 character 'A' is to be translated to X'F0', and
CLCOPYF 00608 the character X'00' is to be translated to
CLCOPYF 00609 X'FF'. If the preceding translation list had
CLCOPYF 00610 been specified in conjunction with the LOWCASE
CLCOPYF 00611 option, then the translation 'A' to X'F0' would
CLCOPYF 00612 override the LOWCASE translation, 'A' to 'a'.

CLCOPYF 00613 The translation list can be continued onto
CLCOPYF 00614 additional lines by typing '++' as the last two
CLCOPYF 00615 characters of the present line. When these two
CLCOPYF 00616 characters are encountered where the first
CLCOPYF 00616 character of a character pair is expected, then
CLCOPYF 00617 all scanning of that line ceases, and the
CLCOPYF 00617 keyboard unlocks so that a new line may be
CLCOPYF 00618 entered.

CLCOPY2 00004 Option Incompatibility

CLCOPY2 00006 Figure 16 shows combinations of options which may not be
CLCOPY2 00008 specified together in the same COPYFILE command.

CLCOPY2 00009 If the option in the first column is specified, none of the
CLCOPY2 00011 options in the second column may be coded.

TYPE	NOTYPE
NOTYPE	TYPE
NEWDATE	OLDDATE, APPEND
OLDDATE	NEWDATE, APPEND
RECFM	PACK, UNPACK, APPEND
LRECL	PACK, UNPACK, APPEND
PROMPT	NOPROMPT
NOPROMPT	PROMPT
FROM	FRLABEL, PACK, UNPACK
FRLABEL	FROM, PACK, UNPACK
FOR	TOLABEL, PACK, UNPACK
TOLABEL	FOR, PACK, UNPACK
TRUNC	NOTRUNC, PACK, UNPACK
NOTRUNC	TRUNC, PACK, NOPACK
PACK	RECFM, LRECL, TRUNC, UNPACK, EBCDIC, UPCASE, LOWCASE, OVLY, APPEND, SPECS, TRANS, FROM FRLABEL, FOR, TOLABEL
UNPACK	RECFM, LRECL, TRUNC, PACK, EBCDIC, UPCASE, LOWCASE, OVLY, APPEND, SPECS, TRANS, FROM, FRLABEL, FOR, TOLABEL
EBCDIC	PACK, UNPACK
UPCASE	PACK, UNPACK
LOWCASE	PACK, UNPACK
NEWFILE	REPLACE, OVLY, APPEND
REPLACE	NEWFILE, OVLY, APPEND
OVLY	NEWFILE, REPLACE, APPEND, PACK, UNPACK
APPEND	NEWDATE, OLDDATE, RECFM, LRECL, NEWFILE, REPLACE, OVLY, PACK, UNPACK
SPECS	PACK, UNPACK, NOSPECS
NOSPECS	PACK, UNPACK, SPECS
TRANS	PACK, UNPACK

Figure 16. COPYFILE Option Incompatibilities

Example 1

```
COPYFILE    OLD FILE A1        NEW FILE A1
```

Action: This command copies the file OLD FILE A1 to a new file created by this command and named NEW FILE A1.

CLCOPY2 00078 Example 2

```
CLCOPY2 00080 ┌──────────────────────────────────────────────────┐
CLCOPY2 00082 │ COPYFILE * FILE A1          BIG FILE A1          │
CLCOPY2 00083 └──────────────────────────────────────────────────┘
```

CLCOPY2 00086 Action: This command copies all files with filetype of FILE
CLCOPY2 00087 and filemode of A1 and combines them into a single output
CLCOPY2 00088 file named BIG FILE A1. If the user has files A FILE A1, B
CLCOPY2 00089 FILE A1 and C FILE A1, they are copied to BIG FILE A1 and
CLCOPY2 00090 are not erased.

CLCOPY2 00094 Example 3

```
CLCOPY2 00096 ┌──────────────────────────────────────────────────┐
CLCOPY2 00098 │ COPYFILE OLD * B1          NEW = B1              │
CLCOPY2 00104 └──────────────────────────────────────────────────┘
```

CLCOPY2 00107 Action: This command copies a group of files, each of which
CLCOPY2 00108 has a filetype of OLD and a filemode of B1 to a new group of
CLCOPY2 00109 files each with a filetype of NEW and a filemode of B1. If
CLCOPY2 00110 the user has files named OLD NAME B1, OLD BIRTHDA B1, OLD
CLCOPY2 00111 ADDRESS B1, they will be copied respectively to files named
CLCOPY2 00112 NEW NAME B1, NEW BIRTHDA B1, and NEW ADDRESS B1.

CLCOPY2 00115 Example 4

```
CLCOPY2 00118 ┌──────────────────────────────────────────────────┐
CLCOPY2 00119 │ COPYFILE X Y A1 P Q A1   BIG FILE A1             │
CLCOPY2 00120 └──────────────────────────────────────────────────┘
```

CLCOPY2 00123 Action: This command combines files X Y A1 and P Q A1 into a
CLCOPY2 00124 single output file named BIG FILE A1.

CLCOPY2 00127 Example 5

```
CLCOPY2 00129 ┌──────────────────────────────────────────────────┐
CLCOPY2 00131 │ COPYFILE X * A1          = FILE =                │
CLCOPY2 00137 └──────────────────────────────────────────────────┘
```

CLCOPY2 00140 Action: All files with a filename of X and a filemode of A1
CLCOPY2 00142 are combined into a single output file named X FILE A1.

CLCOPY2 00150 Example 6

```
CLCOPY2 00152 ┌──────────────────────────────────────────────────┐
CLCOPY2 00153 │ COPYFILE X * A1 = T A1 BIG FILE A1               │
CLCOPY2 00154 └──────────────────────────────────────────────────┘
```

CLCOPY2 00157 Action: All files with filename X and filemode A1 are
CLCOPY2 00158 combined with all files with a filetype T and filemode A1 to
CLCOPY2 00159 form a single file named BIG FILE A1. This command could
CLCOPY2 00160 also have been entered as:

```
CLCOPY2 00162 COPYFILE X * A1 = T = BIG FILE =
```

CLCOPY2 00164 Example 7

```
CLCOPY2 00166 ┌──────────────────────────────────────────────────┐
CLCOPY2 00167 │ COPYFILE X * A1 P Q A1 BIG = A1                  │
CLCOPY2 00168 └──────────────────────────────────────────────────┘
```

CLCOPY2 00172 Action: Each file with a filename of X and a filemode of A1
CLCOPY2 00173 is combined with P Q A1 to create a file named BIG * A1. One
CLCOPY2 00173 file is created for each existing file with a filename X and
CLCOPY2 00174 filemode A1.

CLCOPY2 00177 Example 8

```
CLCOPY2 00179 ┌──────────────────────────────────────────────────┐
CLCOPY2 00180 │ COPYFILE X * A1 P = A1 BIG = A1                  │
CLCOPY2 00181 └──────────────────────────────────────────────────┘
```

CLCOPY2 00184 Action: Each file with a filename of X and a filemode of A1
CLCOPY2 00185 is combined with a file with a filename of P and a filemode
CLCOPY2 00186 of A1 and a filetype corresponding to that of the first
CLCOPY2 00187 input file to produce a file with a filename of BIG, a
CLCOPY2 00188 corresponding filetype, and a filemode of A1. This command
CLCOPY2 00189 could also have been entered as:

```
CLCOPY2 00191 COPYFILE X * A1 P = = BIG = =
```

CLCOPY2 00194 Example 9

```
CLCOPY2 00196 ┌──────────────────────────────────────────────┐
CLCOPY2 00197 │ COPYFILE BIGNAME A A1 XYZ= == =              │
CLCOPY2 00198 └──────────────────────────────────────────────┘
```

CLCOPY2 00201 Action: File BIGNAME A A1 is copied tc a file named
CLCOPY2 00202 XYZBIGNA AA A1.

CLCOPY2 00204 Example 10

```
CLCOPY2 00207 ┌──────────────────────────────────────────────┐
CLCOPY2 00208 │ COPYFILE A B A1 C D A1 (FROM 10 FOR 25)      │
CLCOPY2 00209 └──────────────────────────────────────────────┘
```

CLCOPY2 00212 Action: Twenty five records are copied from file A E A1 to
CLCOPY2 00214 file C D A1, beginning with the tenth record of A B A1.

CLCOPY2 00216 Example 11

```
CLCOPY2 00218 ┌──────────────────────────────────────────────────────┐
CLCOPY2 00219 │ COPYFILE OLD B A1 NEW B A1 (FRLABEL MYPROG TOLABEL FINIS) │
CLCOPY2 00220 └──────────────────────────────────────────────────────┘
```

CLCOPY2 00223 Action: Records are copied from file OLD B A1 to NEW B A1.
CLCOPY2 00224 Copying begins with a record containing the characters
CLCOPY2 00224 'MYPROG' and continues until a record containing the
CLCOPY2 00227 characters 'FINIS' is encountered.

CLCOPY2 00229 Example 12

```
CLCOPY2 00231 ┌──────────────────────────────────────────────┐
CLCOPY2 00232 │ COPYFILE * ASSEMBLE * (PACK                  │
CLCOPY2 00233 └──────────────────────────────────────────────┘
```

CLCOPY2 00236 Action: All files with a filetype of ASSEMBLE are converted
CLCOPY2 00237 to packed format.

CLCOPY2 00239 Example 13

```
CLCOPY2 00241 ┌──────────────────────────────────────────────────┐
CLCOPY2 00242 │ COPYFILE  MY FILE A1  YOUR FILE A1  (SPECS       │
CLCOPY2 00243 └──────────────────────────────────────────────────┘
```

CLCOPY2 00246 Action: The file named MY FILE A1 is to be copied to a new
CLCOPY2 00247 file named YOUR FILE A1 and is to be modified according to
CLCOPY2 00248 certain specifications before the new file is written. The
CLCOPY2 00249 following message types at the terminal:

CLCOPY2 00251 DMSCPY601R ENTER SPECIFICATION LIST

CLCOPY2 00252 The keyboard unlocks so that the specification list can be
CLCOPY2 00253 entered.

CLCOPY2 00254 For example, if the following specification list were
CLCOPY2 00255 entered:

CLCOPY2 00257 1-5 10 /ABC/ 3 H00 1

CLCOPY2 00258 Positions 1-5 of the input record will be copied to
CLCOPY2 00259 positions 10-14 of the output record, the character string
CLCOPY2 00260 "ABC" will be placed in positions 3-5 of the output record,
CLCOPY2 00261 and the character X'00' will be placed in the first position
CLCOPY2 00262 of the output record. All other positions in the record
CLCOPY2 00263 will contain blanks unless a FILL character is specified, in
CLCOPY2 00263 which case the FILL character will be placed in all
CLCOPY2 00264 unspecified positions.

CLCOPY2 00266 Example 14

```
CLCOPY2 00268 ┌──────────────────────────────────────────────────┐
CLCOPY2 00269 │ COPYFILE  OLD FORM A1  NEW FORM A1  (SPECS OVLY  │
CLCOPY2 00270 └──────────────────────────────────────────────────┘
```

CLCOPY2 00274 Action: The output file already exists and is to be overlaid
CLCOPY2 00275 with certain portions of the input record or character
CLCOPY2 00276 strings specified in a specification list.

CLCOPY2 00280 Responses

CLCOPY2 00283 DMSCPY601R ENTER SPECIFICATION LIST:

CLCOPY2 00285 This message requests the specification list which is to be
CLCOPY2 00286 entered in conjunction with the SPECS option.

CLCOPY2 00288 DMSCPY602R ENTER TRANSLATION LIST:

CLCOPY2 00290 This message requests the translation list which is to be
CLCOPY2 00291 entered in conjunction with the TRANS option.

CLCOPY2 00297 DMSCPY721I COPY 'fn ft fm' {TO |APPEND| OVLY} 'fn ft fm' {{CLD|NEW} FILE}

CLCOPY2 00301 This message appears in conjunction with the TYPE option.
CLCOPY2 00303 It indicates the names of the input file and output file.

CLCPFUNC 00004 **CP**

CLCPFUNC 00007 The CP command transmits commands to the VM/370 control
CLCPFUNC 00008 program environment without forcing the user to leave the
CLCPFUNC 00009 CMS environment.

CLCPFUNC 00019
CLCPFUNC 00021 | CP | [line] |
CLCPFUNC 00023

CLCPFUNC 00028 line any of the CP commands which are permitted for the
CLCPFUNC 00029 user's CP command privilege class. If this field is
CLCPFUNC 00029 omitted the user is placed in CP mode and may then
CLCPFUNC 00030 enter CP commands without the "CP" preceding the
CLCPFUNC 00030 command. To return to CMS, he must enter the CP
CLCPFUNC 00031 command BEGIN.

CLCPFUNC 00037 **Responses**

CLCPFUNC 00039 All responses are from the CP command which was issued.

CLCPFUNC 00041 Example

CLCPFUNC 00043
CLCPFUNC 00044 | CP TERMINAL LINESIZE 80 |
CLCPFUNC 00045

CLCPFUNC 00048 **Action:** The TERMINAL command will be passed to the control
CLCPFUNC 00050 program for processing.

CLDDR 00008 **DDR**

CLDDR 00010 The DDR command provides the user with the ability to:

CLDDR 00013 1. Save data from direct access storage devices on
CLDDR 00014 magnetic tape.

CLDDR 00016 2. Restore data which has been saved on magnetic tape to a
CLDDR 00017 direct access storage device.

CLDDR 00019 3. Copy data from one device to another device of the same
CLDDR 00020 type. Data may be reordered on a cylinder basis. A
CLDDR 00021 tape-to-tape copy may be done on data saved by this
CLDDR 00022 program.

CLDDR 00025 4. Print specified records.

CLDDR 00027 5. Type specified records at the terminal.

CLDDR 00037 ┌──────────────────────┐
CLDDR 00038 | DDR | [fn ft [fm]] |
CLDDR 00039 └──────────────────────┘

CLDDR 00044 fn ft fm specifies the file identifier of a file which
CLDDR 00045 contains DDR control statements. If no file
CLDDR 00045 identifier is given, the user is requested to
CLDDR 00047 enter control statements at the terminal.

CLDDR 00050 DDR Control Statements: These statements may be either be
CLDDR 00050 in an existing file or they may be entered at the terminal
CLDDR 00051 during execution.

CLDDR 00053 I/O Definition Statements

CLDDR 00054 INPUT or OUTPUT card – used to describe all tape and DASD
CLDDR 00055 units used. For DUMP, RESTORE, and COPY operations, one
CLDDR 00056 INPUT and one OUTPUT card is required. For PRINT and TYPE
CLDDR 00057 operations, only an INPUT card is required.

CLDDR 00060 ┌──┐
CLDDR 00061 | | | | | |
CLDDR 00062 | | | ┌ ┐ | ┌ ┐ ┐ |
CLDDR 00063 | INPUT|OUTPUT | ccu type |volser| |([SKIP xx][MODE den]|REWIND|) |
CLDDR 00064 | IN |OUT | |altape| | |UNLOAD| | |
CLDDR 00065 | | | └ ┘ | └ ┘ | |
CLDDR 00066 | | | └ ┘ |
CLDDR 00067 └──┘

```
CLDDR   00072   INPUT     specifies that the device described is tc be used
CLDDR   00074   IN        as the input device.

CLDDR   00077   OUTPUT    specifies that the device described is tc be used
CLDDR   00080   OUT       as the output device.

CLDDR   00083   ccu       specifies the unit address of the device.

CLDDR   00085   type      specifies the device type. Accepted values for
CLDDR   00086             DASD are 2314, 2319, 3330, and 2305-2. Accepted
CLDDR   00086             values for tape are 2400, 2420, and 3420 (7 track
CLDDR   00088             tapes are not supported).

CLDDR   00090   volser    specifies the volume serial number of a DASD
CLDDR   00091             device.

CLDDR   00093   altape    specifies the address of an alterrate tape drive
CLDDR   00094             to be used if more than one tape is involved in
CLDDR   00095             the operation.

CLDDR   00098   Options

CLDDR   00099   Options, which apply only to tape, may be specified without
CLDDR   00101   a ccu, type, and altape, implying that only the tape
CLDDR   00101   specifications are to be changed. All options specified in
CLDDR   00102   a previous INPUT or OUTPUT card remain in effect unless
CLDDR   00103   overridden.

CLDDR   00106   SKIP xx   specifies that the tape is to be forward spaced
CLDDR   00110   SK   00    xx files. xx is set to zero when the tape is
CLDDR   00111             repositioned.

CLDDR   00114   MODE den  specifies the density and mode settings for an
CLDDR   00115   MO        output tape.

CLDDR   00118             den     mode      density
CLDDR   00119             1600    9 track   1600 BPI
CLDDR   00120             16

CLDDR   00122             800     9 track   800 BPI
CLDDR   00124             8

CLDDR   00128   REWIND    specifies that the tape is to rewound at the
CLDDR   00130   RE        completion of the current function.

CLDDR   00133   UNLOAD    specifies that the tape is to be unloaded at the
CLDDR   00135   UN        completion of the current function.

CLDDR   00138   LEAVE     specifies that the tape is to be positioned after
CLDDR   00141   LE        the tape mark at the completion of the current
CLDDR   00142             function.

CLDDR   00145   SYSPRINT card - used to describe a printer device. It is
CLDDR   00146   used in conjunction with the PRINT function and the DUMP,
```

RESTORE, and COPY functions.

```
┌──────────────────────┐
| SYSPRINT |  ccu      |
| SY       |           |
└──────────────────────┘
```

ccu — specifies the virtual address of the device which is to receive printer output. If this card is not included, output is directed to the printer at address 00E.

Function Control Statements

The (DUMP|COPY|RESTORE) card describes the extents of the operation.

```
┌──────────────────────┬──────────────────────────────────────────────┐
|                      | ┌                                          ┐ |
| DUMP|COPY|RESTORE    | |cyl1 [TO] [cyl2 [REORDER] [TC][cyl3]]     | |
| DU   CO   RE         | |                                          | |
|                      | <CPVOL                                     > |
|                      | |                                          | |
|                      | |ALL                                       | |
|                      | └                                          ┘ |
└──────────────────────┴──────────────────────────────────────────────┘
```

DUMP (DU) — specifies that data is to be transferred from a DASD device to a magnetic tape device.

COPY (CO) — specifies that data is to be copied from one device to another device of the same type (that is, from tape to tape or from disk to disk).

RESTORE (RE) — specifies that data previously placed on magnetic tape by the DDR DUMP function is to be restored to a DASD device.

cyl1 — specifies the first cylinder to be dumped, copied, or restored.

TO (T) — is an optional reserved keyword.

cyl2 — specifies the last cylinder upon which the specified operation is performed. If cyl2 is not specified, only the cylinder specified in cyl1 will be operated upon.

REORDER (R) — is an optional reserved keyword used only if cyl3 is to be specified. This indicates that when

CLDDR 00218 the tape is restored, the data will be placed on
CLDDR 00218 the disk at a different cylinder than the one on
CLDDR 00220 which it originally resided.

CLDDR 00222 **cyl3** specifies the cylinder at which reordering is to
CLDDR 00224 begin. If not specified, the default is cyl1.

CLDDR 00226 **CPVOL** specifies that cylinder 0 and all allocated
CLDDR 00227 directory, and permanent disk space are to be
CLDDR 00228 copied, dumped, or restored. This indicates that
CLDDR 00228 the disk is in CP format, that is, it must have
CLDDR 00229 been formatted by the CP FORMAT/ALLOCATE program.
CLDDR 00231 Refer to the IBM Virtual Machine Facility/370
CLDDR 00231 Operator's Guide (Order No. GC20-1806) for an
CLDDR 00232 explanation of this program.

CLDDR 00235 **ALL** specifies that the operation is to be performed on
CLDDR 00237 **AL** all cylinders.

CLDDR 00240 PRINT or TYPE statement - produces a hexadecimal and EBCDIC
CLDDR 00241 translation of each record specified.

CLDDR 00244
CLDDR 00245 | | cc1 [hh1 [rr1]] [TO cc2 [hh2 [rr2]]] |
CLDDR 00246 | PRINT TYPE | 0 0 |
CLDDR 00247 | PR TY | |
CLDDR 00248 | | |
CLDDR 00249 | | |
CLDDR 00250

CLDDR 00254 **PRINT** specifies that output is to be directed to the
CLDDR 00256 **PR** SYSPRINT device.

CLDDR 00258 **TYPE** specifies that the output is to be directed to the
CLDDR 00260 **TY** console.

CLDDR 00262 **cc1** specifies the cylinder at which printing or typing
CLDDR 00263 is to begin.

CLDDR 00265 **hh1** specifies the track at which printing or typing is
CLDDR 00267 to begin. If omitted, a value of zero is assumed.

CLDDR 00269 **rr1** specifies the ID of the record at which printing
CLDDR 00270 or typing is to begin. If omitted, home address,
CLDDR 00271 record zero is assumed.

CLDDR 00274 **TO** is a keyword which delimits the starting and
CLDDR 00276 **T** ending address. If not specified, only one
CLDDR 00277 cylinder, track, or record will be printed or
CLDDR 00277 typed depending upon the specification of starting
CLDDR 00278 address.

CLDDR 00280 **cc2** specifies the last cylinder to be printed or

```
CLDDR   00281              typed. If the keyword TO is specified, cc2 must be
CLDDR   00282              specified.

CLDDR   00284   hh2        specifies the last  track of cc2 tc  be printed or
CLDDR   00285              typed. If not  specified, the default is  the last
CLDDR   00286              track on cc2.

CLDDR   00288   rr2        specifies the last record ID  of hh2 to be printed
CLDDR   00289              or typed.  If not  specified,  the  default  is  the
CLDDR   00290              entire track.


CLDDR   00294   Responses


CLDDR   00296   DMSDDR711R VOLID IS volse1 NOT volse2
CLDDR   00298   DO YOU WISH TO CONTINUE? RESPOND YES NO OR REREAD:

CLDDR   00300   volse1     is the volume serial number from the vol1 label on
CLDDR   00301              the DASD unit.

CLDDR   00303   volse2     is  the volume  serial number  from  the input  or
CLDDR   00304              output card.

CLDDR   00307   The volume serial number read from  the device at ccu is not
CLDDR   00308   same as that specified on the INPUT or OUTPUT card.

CLDDR   00310   DMKDDR716R NO VOL1 LABEL FOUND FOR xxxxxx
CLDDR   00312   DO YOU WISH TO CONTINUE? RESPOND YES NO OR REREAD:

CLDDR   00315   xxxxxx     is the  volume serial  number of  the DASD  device
CLDDR   00316              from the input or the output card.

CLDDR   00319   The DASD device at ccu contains no volume serial number.

CLDDR   00321   DMKDDR717R DATA DUMPED FROM vol1 TO BE RESTORED TO vol2
CLDDR   00323   DO YOU WISH TO CONTINUE? RESPOND YES NO OR REREAD:

CLDDR   00326   vol1       is the  volume serial number  from the  input tape
CLDDR   00327              header record (volume dumped).

CLDDR   00330   vol2       is the volume  serial number from the  output DASD
CLDDR   00331              device.

CLDDR   00333   This  message  is  printed for  verification  of  the  input
CLDDR   00334   parameters.

CLDDR   00336   ENTER CYLINDER EXTENTS
CLDDR   00338   ENTER:

CLDDR   00339   This message is received only if  the user is entering input
CLDDR   00340   from the console.

CLDDR   00342   END OF CYL XXX HD XX, MOUNT NEXT TAPE
```

```
CLDDR    00344    RESTORING xxxxxx

CLDDR    00347    xxxxxx    is the volume serial number of the disk dumped.

CLDDR    00350    The RESTORE operation has begun.

CLDDR    00352    COPYING xxxxxx

CLDDR    00354    xxxxxx    is the volume serial number descrited by the input
CLDDR    00355              unit.

CLDDR    00358    The COPY operation has begun.

CLDDR    00360    DUMPING xxxxxx

CLDDR    00362    xxxxxx    is the volume serial number descrited by the input
CLDDR    00363              unit.

CLDDR    00366    The dumping operation has begun.

CLDDR    00368    PRINTING xxxxxx

CLDDR    00371    xxxxxx    is the volume serial number descrited by the input
CLDDR    00372              unit.

CLDDR    00375    The PRINT operation has begun.

CLDDR    00377    END OF DUMP

CLDDR    00379    The DUMP has ended.

CLDDR    00381    END OF RESTORE

CLDDR    00383    The RESTORE operation has ended.

CLDDR    00385    END OF COPY

CLDDR    00387    The COPY operation has ended.

CLDDR    00389    END OF PRINT

CLDDR    00391    The PRINT operation has ended.

CLDDR    00393    END OF JOB

CLDDR    00395    All specified operations have completed.

CLDDR    00397    ENTER:

CLDEBUG  00001    prompts the user to enter input.
```

CLDEBUG 00005 **DEBUG**

CLDEBUG 00007 The DEBUG command provides the user with online facilities
CLDEBUG 00008 for debugging programs and provides an entry in CMS for
CLDEBUG 00010 handling external interrupts, program interrupts, and
CLDEBUG 00011 unrecoverable errors. The subcommands which may be issued in
CLDEBUG 00011 the DEBUG environment allow the user to examine and change
CLDEBUG 00012 the contents of certain control words and registers as well
CLDEBUG 00013 as portions of his virtual storage. The facilities of DEBUG
CLDEBUG 00014 are made available to the user when:

CLDEBUG 00018 - the DEBUG command is issued
CLDEBUG 00020 - an external interrupt occurs
CLDEBUG 00022 - a break point is encountered during program execution
CLDEBUG 00023 which causes a program interrupt

CLDEBUG 00026 Once DEBUG has been entered due to any of the above
CLDEBUG 00026 circumstances, the user is said to be in the DEBUG
CLDEBUG 00027 environment. The only valid input in this environment is the
CLDEBUG 00028 group of DEBUG subcommands.

CLDEBUG 00029 When the DEBUG environment is entered, the contents of all
CLDEBUG 00030 general registers, the channel status word, and the channel
CLDEBUG 00030 address word are saved so they may be examined and changed
CLDEBUG 00031 before being restored when leaving the DEBUG environment.
CLDEBUG 00032 If DEBUG is entered via an interrupt, the old program status
CLDEBUG 00033 word for that interrupt is also saved. If DEBUG is the first
CLDEBUG 00034 command entered after an ABEND occurs, the contents of all
CLDEBUG 00035 general registers, the CSW, the CAW, and the old PSW are
CLDEBUG 00036 available from the time of the ABEND.

CLDEBUG 00047
CLDEBUG 00048 | DEBUG |
CLDEBUG 00049

CLDEBUG 00053 DEBUG SUBCOMMANDS: For a complete description of the DEBUG
CLDEBUG 00055 subcommands refer to IBM Virtual Machine Facility/ 70,
CLDEBUG 00056 Programmer's Guide to Debugging (Order No. GC20-1807). The
CLDEBUG 00059 formats are provided here for reference only.

CLDEBUG 00062
CLDEBUG 00063 | BREAK | { symbol }
CLDEBUG 00064 | BR | id { }
CLDEBUG 00065 | | { hexloc }
CLDEBUG 00066
CLDEBUG 00069 Stop program execution at specified break point.

```
┌──────────────────────────────────────────────────────────────┐
| CAW   |                                                      |
└──────────────────────────────────────────────────────────────┘
```

Type Channel Address Word contents at time DEBUG was entered.

```
┌──────────────────────────────────────────────────────────────┐
| CSW   |                                                      |
└──────────────────────────────────────────────────────────────┘
```

Type contents of Channel Status Word at time DEBUG was entered.

```
┌──────────────────────────────────────────────────────────────┐
| DEFINE |                        ┌─ ─┐                        |
| DEF    | symbol hexloc          | n |                        |
|        |                        | 4 |                        |
|        |                        └─ ─┘                        |
└──────────────────────────────────────────────────────────────┘
```

Assign a symbolic name to a specific core address.

```
┌──────────────────────────────────────────────────────────────┐
|        | ┌─           ┌─       ─┐ ┐                           |
| DUMP   | | symbol1    | symbol2 | |                           |
| DU     | | hexloc     | hexloc2 | |      ident                |
|        | |            |    *    | |                           |
|        | └─           └─       ─┘ ┘                           |
└──────────────────────────────────────────────────────────────┘
```

Dump the contents of specified portions of virtual storage to the virtual spooled printer.

```
┌──────────────────────────────────────────────────────────────┐
|        | ┌─       ─┐                                          |
| GO     | | symbol  |                                          |
|        | | hexloc  |                                          |
|        | └─       ─┘                                          |
└──────────────────────────────────────────────────────────────┘
```

Leave the DEBUG environment and continue execution at the specified instruction or at the instruction indicated by the address portion of the PSW.

```
CLDEBUG  00122  ┌──────────────────────────────────────────────────────────┐
CLDEBUG  00123  | GPR    | reg1  [reg2]                                    |
CLDEBUG  00124  └──────────────────────────────────────────────────────────┘
CLDEBUG  00127  Type the contents of the specified general registers.


CLDEBUG  00129  ┌──────────────────────────────────────────────────────────┐
CLDEBUG  00130  | HX     |                                                 |
CLDEBUG  00131  └──────────────────────────────────────────────────────────┘
CLDEBUG  00134  Return to CMS command environment.


CLDEBUG  00136  ┌──────────────────────────────────────────────────────────┐
CLDEBUG  00137  | ORIGIN| { symbol }                                       |
CLDEBUG  00138  | OR    | { hexloc }                                       |
CLDEBUG  00139  └──────────────────────────────────────────────────────────┘
CLDEBUG  00141  Specify a base address to be added to locations specified in
CLDEBUG  00142  other DEBUG subcommands.


CLDEBUG  00144  ┌──────────────────────────────────────────────────────────┐
CLDEBUG  00145  | PSW    |                                                 |
CLDEBUG  00146  └──────────────────────────────────────────────────────────┘
CLDEBUG  00149  Type contents of the old Program Status Word.


CLDEBUG  00151  ┌──────────────────────────────────────────────────────────┐
CLDEBUG  00152  | RETURN|                                                  |
CLDEBUG  00153  | RET    |                                                 |
CLDEBUG  00154  └──────────────────────────────────────────────────────────┘
CLDEBUG  00157  Exit from DEBUG environment to CMS command environment.


CLDEBUG  00159  ┌──────────────────────────────────────────────────────────┐
CLDEBUG  00160  | SET    | { CAW hexinfo                          }        |
CLDEBUG  00161  |        | { CSW hexinfo [hexinfo]                }        |
CLDEBUG  00162  |        | { PSW hexinfo [hexinfo]                }        |
CLDEBUG  00163  |        | { GPR reg      hexinfo [hexinfo] }             |
CLDEBUG  00164  └──────────────────────────────────────────────────────────┘
CLDEBUG  00167  Change the contents of the specified location or register.
```

```
┌──────────────────────────────────────────────────────────────┐
| STORE | { symbol }                                           |
| ST    | {        } hexinfo                                   |
|       | { hexloc }                                           |
└──────────────────────────────────────────────────────────────┘
```

Store information in a specified virtual location.

```
┌──────────────────────────────────────────────────────────────┐
| TIN   | { CMS }                                              |
|       | { DEB }                                              |
└──────────────────────────────────────────────────────────────┘
```

Indicate means of handling I/O.

```
┌──────────────────────────────────────────────────────────────┐
|       |             ┌─ ─┐                                    |
| X     | { symbol }  | n |                                    |
|       | { hexloc }  | 4 |                                    |
|       |             └─ ─┘                                    |
└──────────────────────────────────────────────────────────────┘
```

Examine virtual storage locations.

Notes:

1. The CMS commands HT and HO are not recognized in the DEBUG environment.

2. The floating-point registers may not be examined or changed in the DEBUG environment.

Responses:

```
                                ┌                        ┐
                                |                        |
DMSDBG728I   DEBUG ENTERED      <EXTERNAL INT            >
                                |BREAKPOINT nn AT hexloc |
                                └                        ┘
```

This specifies that the DEBUG environment was entered for the reason stated.

CLDISK 00004 **DISK**

CLDISK 00007 The DISK command performs two functions:

CLDISK 00011 1. Punches disk files to the virtual spooled card punch in
CLDISK 00011 a special format which will allow the punched deck to
CLDISK 00012 be restored to disk in the form of the original disk
CLDISK 00013 file.

CLDISK 00015 2. Restores punched decks of a special format to a disk
CLDISK 00016 file. These decks must have been created by the DISK
CLDISK 00017 DUMP command.

CLDISK 00029
CLDISK 00030 | DISK | {DUMP fn ft [fm] }
CLDISK 00031 | | {}
CLDISK 00032 | | {LOAD }
CLDISK 00033

CLDISK 00038 DUMP indicates that the specific file is to be
CLDISK 00039 punched. The file may have either fixed or
CLDISK 00039 variable-length records. After all data is
CLDISK 00040 punched, an end-of-file card is created with an
CLDISK 00040 N in column 5. This card contains directory
CLDISK 00041 information, and must remain in the deck. The
CLDISK 00042 original disk file is retained.

CLDISK 00044 LOAD specifies that one or more card files are to be
CLDISK 00045 read from the spooled card reader and written as
CLDISK 00046 CMS files on the user's disk. The DISK LOAD
CLDISK 00046 operation reads a physical card reader deck
CLDISK 00047 consisting of any number of logical decks
CLDISK 00047 previously punched by DISK DUMP. File
CLDISK 00049 designations are obtained from the card stream.
CLDISK 00049 An existing file with the same designation as
CLDISK 00050 one of those in the card stream is erased and
CLDISK 00050 replaced. DISK LOAD loads files onto the
CLDISK 00051 primary read/write disk.

CLDISK 00053 fn ft fm specifies the file identification: filename,
CLDISK 00054 filetype, and filemode.

CLDISK 00058 Example 1

CLDISK 00060
CLDISK 00061 | DISK | DUMP MYFILE OLD A1
CLDISK 00062

CLDISK 00066 Action: The specified file, MYFILE OLD A1, is written to the
CLDISK 00066 virtual spooled punch. It is followed by an end-of-file
CLDISK 00067 card.

CLDISK 00069 Example 2

CLDISK 00072 ┌──────────────────────────────┐
CLDISK 00073 | DISK | LOAD |
CLDISK 00074 └──────────────────────────────┘

CLDISK 00078 Action: All files which were previcusly read into the
CLDISK 00078 virtual card reader are to be loaded onto disk; each file
CLDISK 00079 must be followed by an end-of-file card as created by the
CLDISK 00081 DISK DUMP function.

CLDISK 00084 Responses

CLDISK 00086 DISK DUMP gives no response except the READY message.

CLDISK 00087 DISK LOAD gives the file identifier for each file
CLDISK 00088 encountered in the input deck. The READY message is typed
CLEDIT 00001 after transfer is completed.

CLEDIT 00003 **EDIT**

CLEDIT 00006 The EDITOR allows the CMS user to:

CLEDIT 00010 • Create, from the terminal, sequential files consisting of
CLEDIT 00012 either fixed or variable length records.

CLEDIT 00014 • Provide, on the basis of certain known filetypes,
CLEDIT 00015 automatic selection of record length, record format,
CLEDIT 00016 logical tab settings, serialization, linemode, and
CLEDIT 00017 lowercase to uppercase translation.

CLEDIT 00020 • Add, delete, or change any part of a CMS file.

CLEDIT 00022 • Extract all or part of a CMS file and make it a new file,
CLEDIT 00024 or embed it in another file.

CLEDIT 00026 • Allow searching and changing of portions of the file
CLEDIT 00027 through context-directed searches or by using a specific
CLEDIT 00028 line number.

CLEDIT 00031 • Receive automatic prompting with line numbers.

CLEDIT 00033 • Allow any or all of the file to be typed out at the
CLEDIT 00034 user's terminal.

CLEDIT 00036 • Provide an interface to the CMS EXEC interpreter to allow
CLEDIT 00037 a 'macro' facility.

CLEDIT 00047 ┌──────────────────────────────────┐
CLEDIT 00048 │ EDIT │ fn ft [fm] [(LRECL nn)] │
CLEDIT 00049 │ E │ │
CLEDIT 00050 └──────────────────────────────────┘

CLEDIT 00057 fn ft specify the filename and filetype of the file to
CLEDIT 00058 be created or edited. If a file with the
CLEDIT 00058 specified filename and filetype does not exist,
CLEDIT 00059 the CMS EDITOR assumes that the file is being
CLEDIT 00059 created, and after the input environment is
CLEDIT 00060 entered, information typed by the user becomes
CLEDIT 00061 input to that file. If a file with the same
CLEDIT 00062 filename and filetype does exist, the EDIT
CLEDIT 00063 environment is entered, enabling the user to
CLEDIT 00063 issue EDIT subcommands and to modify the
CLEDIT 00064 specified file. Both fn and ft must be
CLEDIT 00065 specified.

CLEDIT 00067 fm specifies the filemode of the file to be created
CLEDIT 00068 or edited. If fm is specified, EDIT searches
CLEDIT 00069 only the indicated disk. If fm is specified as
CLEDIT 00069 an asterisk, EDIT searches all disks for the
CLEDIT 00070 existence of that file. A blank specification
CLEDIT 00070 for filemode causes searching of only the

CLEDIT 00071 primary disk. If the file is found, its mode is
CLEDIT 00071 saved, EDIT later writes the altered file back
CLEDIT 00072 to that same disk. If an existing file is not
CLEDIT 00072 found, the newly-created file is placed on the
CLEDIT 00073 disk specified by fm or on the user's primary
CLEDIT 00074 disk.

CLEDIT 00076 **LRECL nn** specifies the record length of the file to be
CLEDIT 00077 created or edited. If no record length is given,
CLEDIT 00078 the following default values are assumed:

CLEDIT 00083 <u>Editing Existing files</u>

CLEDIT 00085 F format:
CLEDIT 00087 Existing record length is kept regardless
CLEDIT 00088 of format.

CLEDIT 00091 <u>Creating New files</u>

CLEDIT 00093 V format:
CLEDIT 00095 Filetype LISTING: 121

CLEDIT 00098 F format:
CLEDIT 00100 All filetypes: 80

CLEDIT 00103 If the record length derived in this way is
CLEDIT 00104 smaller that that of the existing file, then the
CLEDIT 00105 EDITOR types out an error message, and must be
CLEDIT 00105 invoked again with an adequate record length
CLEDIT 00106 given. The maximum record length supported by
CLEDIT 00107 the editor is 133.

CLEDIT 00112 <u>Edit Subcommands</u>

CLEDIT 00114 Refer to the <u>IBM Virtual Machine Facility/370, EDIT Guide</u>
CLEDIT 00115 (Order No. GC20-1805) for a functional description of each
CLEDIT 00116 of the edit subcommands. The formats are given here for
CLEDIT 00117 reference only.

```
CLEDIT 00129  ┌──────────────────────────────────────────────────────┐
CLEDIT 00130  | ALTER    |                          ┌           ┐   |
CLEDIT 00131  | AL       |  {char1}  {char2}        | n  ┌    ┐ |   |
CLEDIT 00132  |          |  {hex1 }  {hex2 }        | *  | G  | |   |
CLEDIT 00133  |          |                          | 1  | *  | |   |
CLEDIT 00134  |          |                          └    └    ┘ ┘   |
CLEDIT 00135  └──────────────────────────────────────────────────────┘
```

CLEDIT 00138 Scan the first 'n' records of the file altering the
CLEDIT 00139 specified character either once in each line or for all
CLEDIT 00140 occurrences in the line.

```
CLEDIT   00143   ┌──────────────────────────────────────────┐
CLEDIT   00144   │ BOTTOM  │                                │
CLEDIT   00145   │ B       │                                │
CLEDIT   00146   └──────────────────────────────────────────┘

CLEDIT   00150   Make the last line of the file the current line.

CLEDIT   00153   ┌──────────────────────────────────────────┐
CLEDIT   00154   │ CASE    │ ┌   ┐                          │
CLEDIT   00155   │         │ │ U │                          │
CLEDIT   00156   │         │ │ M │                          │
CLEDIT   00157   │         │ └   ┘                          │
CLEDIT   00158   └──────────────────────────────────────────┘

CLEDIT   00161   Indicate if translation to uppercase is to be done or
CLEDIT   00162   display current status.

CLEDIT   00165   ┌──────────────────────────────────────────────────┐
CLEDIT   00166   │ CHANGE  │                   ┌   ┌           ┐ ┐  │
CLEDIT   00167   │ C       │ /string1/string2  │ / │ n ┌   ┐ │ │  │
CLEDIT   00168   │         │                   │   │ 1 │ G │ │ │  │
CLEDIT   00169   │         │                   │   │ * │ * │ │ │  │
CLEDIT   00170   │         │                   └   └   └   ┘ ┘ ┘  │
CLEDIT   00171   └──────────────────────────────────────────────────┘

CLEDIT   00175   Change string1 to string2 for 'n' records or to EOF, either
CLEDIT   00175   for the first occurrence in each line or for all
CLEDIT   00178   occurrences.

CLEDIT   00181   ┌──────────────────────────────────────────┐
CLEDIT   00182   │ CMS     │                                │
CLEDIT   00183   │         │                                │
CLEDIT   00184   └──────────────────────────────────────────┘

CLEDIT   00189   Enter CMS subset command mode

CLEDIT   00192   ┌──────────────────────────────────────────┐
CLEDIT   00193   │ DELETE  │ ┌   ┐                          │
CLEDIT   00194   │ DEL     │ │ n │                          │
CLEDIT   00195   │         │ │ 1 │                          │
CLEDIT   00196   │         │ │ * │                          │
CLEDIT   00197   │         │ └   ┘                          │
CLEDIT   00198   └──────────────────────────────────────────┘

CLEDIT   00202   Delete 'n' lines or to the end of the file (*).
```

```
┌──────────────────────────────────────────────────────────┐
| DOWN        | ┌   ┐                                       |
| D           | | n |                                       |
|             | | 1 |                                       |
|             | └   ┘                                       |
└──────────────────────────────────────────────────────────┘
```

Point to the n'th line from the current line.

```
┌──────────────────────────────────────────────────────────┐
| FILE        | [fn [ft [fm]]]                             |
|             |                                            |
└──────────────────────────────────────────────────────────┘
```

Save the file being edited on disk cr change its identifiers. Return to CMS.

```
┌──────────────────────────────────────────────────────────┐
| FIND        | line                                       |
|             |                                            |
└──────────────────────────────────────────────────────────┘
```

Search the file for the given line.

```
┌──────────────────────────────────────────────────────────┐
| FMODE       | [fm]                                       |
|             |                                            |
└──────────────────────────────────────────────────────────┘
```

Reset or display the filemode.

```
┌──────────────────────────────────────────────────────────┐
| FNAME       | [fn]                                       |
|             |                                            |
└──────────────────────────────────────────────────────────┘
```

Reset or display the filename.

```
┌──────────────────────────────────────────────────────────┐
| GETFILE |      ┌      ┌            ┌     ┌     ┐  ┐  ┐  ┐ |
| GE      | fn   | ft   | fm         | m   | n   |  |  |  | |
|         |      | *    | *          | 1   | *   |  |  |  | |
|         |      └      └            └     └     ┘  ┘  ┘  ┘ |
└──────────────────────────────────────────────────────────┘
```

Insert some or all of the given file follcwing the current line.

```
┌──────────────────────────────────────────────────────────────┐
│             │ ┌        ┐                                     │
│ IMAGE       │ │ ON     │                                     │
│             │ │ OFF    │                                     │
│             │ │ CANON  │                                     │
│             │ └        ┘                                     │
└──────────────────────────────────────────────────────────────┘
```

Expand text into line images or display current setting.

```
┌──────────────────────────────────────────────────────────────┐
│ INPUT       │ [line]                                         │
│ I           │                                                │
└──────────────────────────────────────────────────────────────┘
```

Insert 'line' in the file or enter INPUT mode.

```
┌──────────────────────────────────────────────────────────────┐
│              │  ┌      ┐                                     │
│ LINEMODE     │  │LEFT  │                                     │
│ LINE         │  │RIGHT │                                     │
│              │  │OFF   │                                     │
│              │  └      ┘                                     │
└──────────────────────────────────────────────────────────────┘
```

Set or display current line setting.

```
┌──────────────────────────────────────────────────────────────┐
│ LOCATE      │ /string [/]                                    │
│ L           │                                                │
└──────────────────────────────────────────────────────────────┘
```

Scan file from next line for first occurence of 'string'.

```
┌──────────────────────────────────────────────────────────────┐
│ LONG        │                                                │
│             │                                                │
└──────────────────────────────────────────────────────────────┘
```

Enter LONG error message mode.

```
┌──────────────────────────────────────────────────────────────┐
│ NEXT        │ ┌   ┐                                          │
│ N           │ │ n │                                          │
│             │ │ 1 │                                          │
│             │ └   ┘                                          │
└──────────────────────────────────────────────────────────────┘
```

Point to the n'th line from current line.

```
CLEDIT  00339    ┌──────────────────────────────────────────┐
CLEDIT  00340    | OVERLAY | line                           |
CLEDIT  00341    | O       |                                |
CLEDIT  00342    └──────────────────────────────────────────┘

CLEDIT  00346    Replace all or part of the current line.

CLEDIT  00349    ┌──────────────────────────────────────────┐
CLEDIT  00350    | PRESERVE|                                |
CLEDIT  00351    | PRE     |                                |
CLEDIT  00352    └──────────────────────────────────────────┘

CLEDIT  00356    Save current mode settings.

CLEDIT  00359    ┌──────────────────────────────────────────┐
CLEDIT  00360    | PROMPT  |  [ incr ]                      |
CLEDIT  00361    |         |                                |
CLEDIT  00362    └──────────────────────────────────────────┘

CLEDIT  00365    Set or  display line number  increment.  Initial  setting is
CLEDIT  00366    10.

CLEDIT  00370    ┌──────────────────────────────────────────┐
CLEDIT  00371    | QUIT    |                                |
CLEDIT  00372    |         |                                |
CLEDIT  00373    └──────────────────────────────────────────┘

CLEDIT  00377    Terminate EDIT session with no updates since last 'SAVE'.

CLEDIT  00380    ┌──────────────────────────────────────────┐
CLEDIT  00381    |         | ┌   ┐                          |
CLEDIT  00382    | RECFORM | | F |                          |
CLEDIT  00383    | REC     | | V |                          |
CLEDIT  00384    |         | └   ┘                          |
CLEDIT  00385    └──────────────────────────────────────────┘

CLEDIT  00389    Set or display record format for subsequent files.

CLEDIT  00392    ┌──────────────────────────────────────────┐
CLEDIT  00393    | REPEAT  | ┌   ┐                          |
CLEDIT  00394    |         | | n |                          |
CLEDIT  00395    |         | | 1 |                          |
CLEDIT  00396    |         | | * |                          |
CLEDIT  00397    |         | └   ┘                          |
CLEDIT  00398    └──────────────────────────────────────────┘

CLEDIT  00402    Execute the following OVERLAY subcommand n times.
```

```
CLEDIT  00405    ┌──────────┬──────────┐
CLEDIT  00406    │ REPLACE  │ [line]   │
CLEDIT  00407    │ R        │          │
CLEDIT  00408    └──────────┴──────────┘

CLEDIT  00411    Replace the  current line with  line or replace  the current
CLEDIT  00412    line and enter INPUT mode.


CLEDIT  00415    ┌──────────┐
CLEDIT  00416    │ RESTORE  │
CLEDIT  00417    │ RES      │
CLEDIT  00418    └──────────┘

CLEDIT  00422    Restore mode settings to values last preserved.


CLEDIT  00425    ┌──────────┐
CLEDIT  00426    │ RETURN   │
CLEDIT  00427    │          │
CLEDIT  00428    └──────────┘

CLEDIT  00432    Return to EDIT environment from CMS subset.


CLEDIT  00434    ┌──────────┬──────────────────┐
CLEDIT  00435    │          │ ┌              ┐ │
CLEDIT  00436    │  REUSE   │ │ edit subcommand │
CLEDIT  00437    │          │ └              ┘ │
CLEDIT  00438    └──────────┴──────────────────┘

CLEDIT  00443    Stack  (LIFO) the  last edit  subcommand  not starting  with
CLEDIT  00444    REUSE or ? and then execute any given edit subcommand.


CLEDIT  00448    ┌──────────┬──────────────────┐
CLEDIT  00449    │ SAVE     │ [fn [ft [fm]]]   │
CLEDIT  00450    │          │                  │
CLEDIT  00451    └──────────┴──────────────────┘

CLEDIT  00455    Save the file on disk and stay in EDIT environment.


CLEDIT  00458    ┌──────────┬──────────────────────────────────┐
CLEDIT  00459    │ SERIAL   │ 8    OFF                     9   │
CLEDIT  00460    │ SER      │ 8                        ┌   ┐ 9   │
CLEDIT  00461    │          │ 8 8  seq  9              │incr│ 9   │
CLEDIT  00462    │          │ 8 8  ON   9              │ 10 │ 9   │
CLEDIT  00463    │          │ 8 8  ALL  9              └   ┘ 9   │
CLEDIT  00464    └──────────┴──────────────────────────────────┘

CLEDIT  00470    Turn serialization on or off in columns 73-80.
```

```
┌────────────────────────────────────────────────────────────┐
| SHORT     |                                                |
|           |                                                |
└────────────────────────────────────────────────────────────┘
```

Enter SHORT error message mode.

```
┌────────────────────────────────────────────────────────────┐
| STACK     | ┌                   ┐                          |
|           | |         n         |                          |
|           | |         1         |                          |
|           | |edit subcommand    |                          |
|           | └                   ┘                          |
└────────────────────────────────────────────────────────────┘
```

Stack 'n' lines, beginning with current line, in terminal input buffer.

```
┌────────────────────────────────────────────────────────────┐
| TABSET    | tabs...                                        |
| TABS      |                                                |
└────────────────────────────────────────────────────────────┘
```

Set the given tabs.

```
┌────────────────────────────────────────────────────────────┐
|           |  ┌   ┐                                         |
| TRUNC     |  | n |                                         |
|           |  | * |                                         |
|           |  └   ┘                                         |
└────────────────────────────────────────────────────────────┘
```

Set or display the column of truncation. * means end of logical record.

```
┌────────────────────────────────────────────────────────────┐
|           | ┌          ┌            ┐ ┐                    |
|           | |    m     |            | |                    |
| TYPE      | |    1     |     n      | |                    |
| T         | |    -     |     *      | |                    |
|           | |    *     |            | |                    |
|           | └          └            ┘ ┘                    |
└────────────────────────────────────────────────────────────┘
```

Type 'm' lines beginning with either the current line, the nth line, or the beginning of the file.

```
CLEDIT   00533    ┌───────────────────────────────────────────────────────┐
CLEDIT   00534    | TOP |                                                 |
CLEDIT   00535    |     |                                                 |
CLEDIT   00536    └───────────────────────────────────────────────────────┘

CLEDIT   00540    Point to the beginning of the file.


CLEDIT   00542    ┌───────────────────────────────────────────────────────┐
CLEDIT   00543    |           | ┌   ┐                                     |
CLEDIT   00544    | UP        | | n |                                     |
CLEDIT   00545    | U         | | 1 |                                     |
CLEDIT   00546    |           | └   ┘                                     |
CLEDIT   00547    └───────────────────────────────────────────────────────┘

CLEDIT   00551    Point to the line 'n' lines above the current one.


CLEDIT   00554    ┌───────────────────────────────────────────────────────┐
CLEDIT   00555    |           | ┌   ┐ ┌   ┐                               |
CLEDIT   00556    | VERIFY    | |ON | |n  |                               |
CLEDIT   00557    | V         | |OFF| |*  |                               |
CLEDIT   00558    |           | └   ┘ └   ┘                               |
CLEDIT   00559    └───────────────────────────────────────────────────────┘

CLEDIT   00562    Set, display, or reset verify mode.   * means end of logical
CLEDIT   00563    record.


CLEDIT   00565    ┌───────────────────────────────────────────────────────┐
CLEDIT   00566    |           | ┌                   ┐                     |
CLEDIT   00567    | X         | | edit subcommand   |                     |
CLEDIT   00568    | Y         | |       n           |                     |
CLEDIT   00569    |           | |       1           |                     |
CLEDIT   00570    |           | └                   ┘                     |
CLEDIT   00571    └───────────────────────────────────────────────────────┘

CLEDIT   00574    Assign to  X or Y the  given edit subcommand or  execute the
CLEDIT   00576    previously assigned subcommand n times.


CLEDIT   00578    ┌───────────────────────────────────────────────────────┐
CLEDIT   00579    |           | ┌           ┌           ┐ ┐               |
CLEDIT   00580    | ZONE      | |     m     |     n     | |               |
CLEDIT   00581    | Z         | |     1     |           | |               |
CLEDIT   00582    |           | |     *     |     *     | |               |
CLEDIT   00583    |           | └           └           ┘ ┘               |
CLEDIT   00584    └───────────────────────────────────────────────────────┘

CLEDIT   00587    Set or display the columns between  which editing is to take
CLEDIT   00588    place.
```

CLEDIT 00592–00596

?	

CLEDIT 00599 Type out the last edit subcommand which did not begin with
CLEDIT 00600 REUSE or ?

CLEDIT 00602–00606

nnnnn	[text]

CLEDIT 00609 Locate the line specified by the given line number and
CLEDIT 00611 insert text if given.

CLEDIT 00617–00620

$DUP	n

CLEDIT 00622 Duplicate the current line n times.

CLEDIT 00624–00627

$MOVE	m {UP } n {DOWN}

CLEDIT 00629 Move m lines up or down n lines.

CLEDIT 00636 <u>Responses</u>

CLEDIT 00638 NEW FILE:

CLEDIT 00640 The specified file does not exist.

CLEDIT 00642 EDIT:

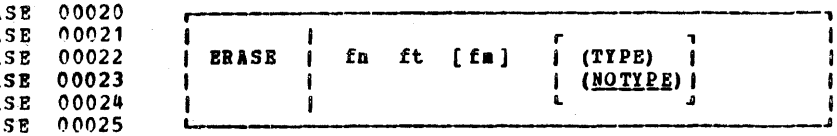
CLEDIT 00643 The EDIT environemnt is entered. The logical tab settings
CLEDIT 00644 may be either those defined by the user or those assumed
CLEDIT 00646 from the filetype. An EDIT subcommand may not be issued.

```
CLEDIT   00649    INPUT:

CLEDIT   00650    The input environment is entered by issuing the EDIT
CLEDIT   00652    subcommands REPLACE or INPUT with no parameters.  All
CLEDIT   00652    subsequent input lines will be accepted as input to the
CLEDIT   00653    file.


CLEDIT   00655    ?EDIT: line

CLEDIT   00656    "line" has been entered in the EDIT environment and is an
CLEDIT   00657    invalid EDIT subcommand or EDIT macro subcommand.  This
CLEDIT   00659    message appears while in LONG error message mode.


CLEDIT   00662    ¬

CLEDIT   00663    An invalid subcommand has been entered in the EDIT
CLEDIT   00665    environment while in SHORT error message mode.


CLEDIT   00667    ¬$

CLEDIT   00669    An invalid EDIT macro subcommand has been entered while in
CLERASE  00001    SHORT error message mode.
```

CLERASE 00005 **ERASE**

CLERASE 00007 The ERASE command deletes a file or a related group of files
CLERASE 00008 from a user's read/write disk. The file to be deleted must
CLERASE 00009 not be on a read/only disk.

CLERASE 00020
CLERASE 00021
CLERASE 00022
CLERASE 00023
CLERASE 00024
CLERASE 00025



CLERASE 00030 fn specifies the filename of the files to be erased.
CLERASE 00031 An asterisk may be coded in this position to
CLERASE 00031 indicate that all filenames are to be used. This
CLERASE 00032 field must be specified, either with a name or an
CLERASE 00033 asterisk.

CLERASE 00035 ft specifies the filetype of the files to be erased.
CLERASE 00036 An asterisk may be coded in this position to
CLERASE 00036 indicate that all filetypes are to be used. This
CLERASE 00037 field must be specified, either with a name or an
CLERASE 00038 asterisk.

CLERASE 00040 fm specifies the filemode of the files to be erased.
CLERASE 00041 If this field is omitted, the primary read/write
CLERASE 00042 disk is searched for the files to be erased.

CLERASE 00045 **Options**

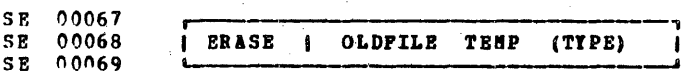
CLERASE 00048 TYPE indicates that the file identifier for each file
CLERASE 00050 T erased is to be typed at the terminal.

CLERASE 00054 NOTYPE file identifiers are not to be typed at the
CLERASE 00058 N terminal.

CLERASE 00060 **Note**: If asterisk is specified for filename and filetype
CLERASE 00061 then filemode must be specified. The filemode must include
CLERASE 00063 both a mode letter and number.

CLERASE 00065 Example

CLERASE 00067
CLERASE 00068
CLERASE 00069



CLERASE 00072 **Action**: The file with the identifier OLDFILE TEMP, if
CLERASE 00073 located on the primary read/write disk, is erased. Its file

```
CLERASE  00075    identifier is typed at the terminal.


CLERASE  00080    Responses

CLERASE  00082    fn      ft      fm

CLERASE  00083    If the TYPE  option has been specified, the  fileid for each
CLEXEC1  00001    file erased is typed.
```

CLEXEC1 00005 **EXEC**

CLEXEC1 00008 The EXEC command executes one or more CMS commands or EXEC
CLEXEC1 00008 control commands contained in a specified file, allowing a
CLEXEC1 00009 sequence of commands to be executed by issuing a single
CLEXEC1 00010 command. If this command is entered from the CMS command
CLEXEC1 00011 mode but not nested within another EXEC procedure, the
CLEXEC1 00013 initial word 'EXEC' may be omitted.

CLEXEC1 00022 ┌──────────────────────────────────────┐
CLEXEC1 00023 | EXEC | fn [args...] |
CLEXEC1 00024 | EX | |
CLEXEC1 00025 └──────────────────────────────────────┘

CLEXEC1 00032 fn specifies the filename of a file containing one or
CLEXEC1 00032 more CMS commands to be executed. The filetype of
CLEXEC1 00033 the file must be EXEC and the file must be fixed
CLEXEC1 00034 format with a logical record length not exceeding
CLEXEC1 00034 130 characters. EXEC files can be created with the
CLEXEC1 00035 EDIT command or by a user's program. EXEC files
CLEXEC1 00036 created by the CMS EDITOR have a logical record
CLEXEC1 00037 length of 80 characters. Each EXEC file may contain
CLEXEC1 00039 a maximum of 4096 lines.

CLEXEC1 00042 args specify the arguments to replace the numeric
CLEXEC1 00042 variables in the EXEC file specified. Within an
CLEXEC1 00043 EXEC file, up to thirty symbolic variables may be
CLEXEC1 00044 used (each one indicated by an ampersand (&)
CLEXEC1 00044 followed by an integer ranging from one to thirty)
CLEXEC1 00045 to indicate values which are to be replaced when the
CLEXEC1 00046 EXEC file is executed. The arguments are assigned
CLEXEC1 00047 to symbolic variables in the order in which they
CLEXEC1 00047 appear in the argument list. For example, each time
CLEXEC1 00049 an &1 appears in an EXEC line, the first argument
CLEXEC1 00050 specified with the EXEC command temporarily replaces
CLEXEC1 00050 the &1, the second argument specified with the EXEC
CLEXEC1 00051 command replaces &2, and so on, to argument N of the
CLEXEC1 00052 EXEC command.

CLEXEC1 00053 If the double quotation mark (") is used in place of
CLEXEC1 00054 an argument, the corresponding variable (&N) is
CLEXEC1 00054 ignored in all the commands which refer to that
CLEXEC1 00055 variable. If the specified EXEC file contains more
CLEXEC1 00055 variables than arguments given with the EXEC
CLEXEC1 00055 command, the higher numbered variables are assumed
CLEXEC1 00056 to be missing, and are ignored when the commands are
CLEXEC1 00057 executed.

CLEXEC1 00061 **EXEC FILES**

CLEXEC1 00063 EXEC files consist of two types of statements:

CLEXEC1 00066 1. nonexecutable - begins with an asterisk (*) and may or
CLEXEC1 00067 may not contain text. These statements are intended
CLEXEC1 00068 for use as comment statements and are ignored during
CLEXEC1 00069 execution.

CLEXEC1 00071 2. executable - does not begin with an asterisk. These
CLEXEC1 00072 statements consist of sequences of strings of
CLEXEC1 00074 contiguous non-blank characters separated by blanks.

CLEXEC1 00078 **Interpretation of Executable Statements**

CLEXEC1 00079 Executable statements are interpreted, one at a time,
CLEXEC1 00081 according to the following steps.

CLEXEC1 00084 1. Except for those commands that take a 'line' (an
CLEXEC1 00085 arbitrary, unsubstituted, collection of words) as an
CLEXEC1 00086 argument, the words forming a statement are
CLEXEC1 00087 'tokenized'. That is, each word is treated as an eight
CLEXEC1 00087 character quantity and is padded with blanks or
CLEXEC1 00089 truncated, as necessary.

CLEXEC1 00091 2. The tokens are searched for the names of any EXEC
CLEXEC1 00092 variable, which are replaced by their values. There is
CLEXEC1 00093 an exception if the token is the target of an
CLEXEC1 00094 assignment; in this case the name of the variable is
CLEXEC1 00095 retained. Tokens forming a statement are searched for
CLEXEC1 00096 the names of EXEC variables, which are replaced by
CLEXEC1 00097 their values as follows.

CLEXEC1 00102 a. Each token is scanned for ampersands, starting with
CLEXEC1 00104 the rightmost character of the token.

CLEXEC1 00106 b. If an ampersand is found, then it, with the rest of
CLEXEC1 00108 the token to the right, is taken as a name. Then:

```
CLEXEC1 00111          IF it is the name of an active variable,
CLEXEC1 00112               it is replaced (in the token) by its value;
CLEXEC1 00113          ELSE
CLEXEC1 00114               IF it is the name of an EXEC keyword,
CLEXEC1 00115                    it is left unaltered;
CLEXEC1 00116               ELSE
CLEXEC1 00117                    it is replaced (in the token) by blanks.
```

CLEXEC1 00122 (An EXEC keyword is a control word, a built-in
CLEXEC1 00123 function, or either of the special tokens &$ and
CLEXEC1 00124 &*.) The token so formed is padded with blanks,

```
CLEXEC1  00124          or truncated, as necessary, to maintain a length
CLEXEC1  00125          of eight.

CLEXEC1  00129      c. Scanning resumes at the next character to the left,
CLEXEC1  00130          and the procedure is repeated from (b) above,
CLEXEC1  00131          until the token is exhausted. There is an
CLEXEC1  00131          exception if the token is the target of an
CLEXEC1  00132          assignment; in this case scanning for ampersands
CLEXEC1  00132          effectively stops on the second character of the
CLEXEC1  00133          token.

CLEXEC1  00135      Note that any characters which are substituted are not
CLEXEC1  00136      scanned for ampersand. They are, however, included in
CLEXEC1  00136      the next name if another ampersand is found to the
CLEXEC1  00137      left.

CLEXEC1  00138      This processing makes it possible to simulate the
CLEXEC1  00140      effects of subscripted variables. For example, the
CLEXEC1  00141      sequence:

CLEXEC1  00143              &X = 123
CLEXEC1  00145              &TYPE ABC &X ABC&X 000000&X

CLEXEC1  00147      yields the printed line:

CLEXEC1  00149              ABC 123 ABC123 00000012

CLEXEC1  00151      The sequence:

CLEXEC1  00153              entered                 after substitution
CLEXEC1  00154              &I = 2
CLEXEC1  00157              ...
CLEXEC1  00158              &X&I = 5                &X2 = 5
CLEXEC1  00159              &I = &I - 1             &I = 1
CLEXEC1  00160              &X&I = &I + 1           &X1 = 2
CLEXEC1  00161              ...
CLEXEC1  00162              &X = &X&I + &X&X&I      &X = &X1 + &X2 or &X = 2 + 5
CLEXEC1  00163              &TYPE ANSWER IS &X

CLEXEC1  00165      yields the printed line:

CLEXEC1  00167              ANSWER IS 7


CLEXEC1  00174   3. If at this point the token is entirely blank, it is
CLEXEC1  00175      discarded from the statement, so that the next token is
CLEXEC1  00177      deemed to immediately follow the previous one.

CLEXEC1  00180   4. The statement is analyzed syntactically, and executed.
```

CLEXEC1 00185 Types of executable statements:

CLEXEC1 00186 There are four classes of executable statements: 1) null
CLEXEC1 00187 statements, 2) CMS commands, 3) assignment statements, and
CLEXEC1 00189 4) control statements. The four classes are described below.

CLEXEC1 00192 1. Null statements.

CLEXEC1 00193 A null statement is an executable statement in which
CLEXEC1 00195 the number of tokens is zero.

CLEXEC1 00198 2. CMS commands.

CLEXEC1 00200 An executable statement is a CMS command if the first
CLEXEC1 00201 token does not start with an ampersand or asterisk. It
CLEXEC1 00201 is executed immediately. When execution is finished,
CLEXEC1 00202 control returns to the EXEC file, and the completion
CLEXEC1 00203 code from the CMS command is placed in the special EXEC
CLEXEC1 00204 variable &RETCODE.

CLEXEC1 00207 3. Assignment statements.

CLEXEC1 00208 An executable statement is an assignment statement if
CLEXEC1 00209 the first token starts with an ampersand and the second
CLEXEC1 00210 token is an equal sign. The first token is taken as the
CLEXEC1 00211 name of an EXEC variable, and is assigned the value of
CLEXEC1 00212 the expression which follows the equal sign. The
CLEXEC1 00213 expression may be any of the following:

CLEXEC1 00216 a. A single token, such as ABC

CLEXEC1 00219 b. An arithmetic expression, consisting of a sequence
CLEXEC1 00220 of tokens which possess positive or negative
CLEXEC1 00220 integral values and are separated by plus or minus
CLEXEC1 00222 signs, such as 3 - 4 + -11 - 00.

CLEXEC1 00224 c. A built-in function followed by its arguments, such
CLEXEC1 00225 as &SUBSTR &1 2 1.

CLEXEC1 00230 4. Control statements.

CLEXEC1 00231 An executable statement is a control statement if the
CLEXEC1 00232 first token is an EXEC control word and the second
CLEXEC1 00233 token is not an equal sign. Examples of control words
CLEXEC1 00234 are:

CLEXEC1 00237 &GOTO
CLEXEC1 00239 &EXIT
CLEXEC1 00241 &IF

CLEXEC1 00243 Control statements begin with a control word, which is
CLEXEC1 00244 usually followed by a list of tokens, and in some cases
CLEXEC1 00245 by additional lines of data. The control words, and

CLEXEC1 00246 the rules for their use, are as follows.

CLEXEC1 00257
CLEXEC1 00258
CLEXEC1 00259

```
| &ARGS | [arg1 [arg2 ... ]] |
```

CLEXEC1 00264 Redefine the arguments &1, &2, ... with the value of
CLEXEC1 00264 'arg1', 'arg2', ..., and reset the variable &INDEX to
CLEXEC1 00265 the number of arguments so set. The remaining
CLEXEC1 00274 arguments are set to blanks (see &READ ARGS).

CLEXEC1 00276
CLEXEC1 00277
CLEXEC1 00278

```
| &BEGPUNCH | [ALL] |
```

CLEXEC1 00280 line1
CLEXEC1 00282 line2
CLEXEC1 00284 ...

CLEXEC1 00286
CLEXEC1 00287
CLEXEC1 00288

```
| &END |
```

CLEXEC1 00291 Punch 'line1', 'line2', ... into the card-punch,
CLEXEC1 00292 without tokenizing them. The lines are normally
CLEXEC1 00292 truncated at column 72 and padded with blanks to fill
CLEXEC1 00293 an eighty-column card; truncation can be avoided by
CLEXEC1 00293 specifying the option 'ALL'. The statement is
CLEXEC1 00294 terminated by a line in which the string '&END' starts
CLEXEC1 00296 in column 1 (see &PUNCH).

CLEXEC1 00298
CLEXEC1 00299
CLEXEC1 00300
CLEXEC1 00301
CLEXEC1 00302
CLEXEC1 00303

```
|           | [FIFO] [ALL] |
| &BEGSTACK | [LIFO]       |
```

CLEXEC1 00305 LIFO
CLEXEC1 00307 line1
CLEXEC1 00309 line2
CLEXEC1 00312 ...

CLEXEC1 00314
CLEXEC1 00315
CLEXEC1 00316

```
| &END |
```

CLEXEC1 00319 Stack 'line1', 'line2', ..., in the console input
CLEXEC1 00320 buffer without tokenizing them. The lines are normally
CLEXEC1 00321 stacked FIFO (first in, first out), but this can be
CLEXEC1 00321 changed by specifying the option 'LIFO' (last in, first
CLEXEC1 00322 out). The lines are normally truncated at column 72,
CLEXEC1 00323 but this can be avoided by specifying the option 'ALL'.
CLEXEC1 00323 The statement is terminated by a line in which the
CLEXEC1 00325 string '&END' starts in column 1.

```
CLEXEC1  00328          ┌──────────────────┐
CLEXEC1  00329          │ &BEGTYPE │ [ALL] │
CLEXEC1  00330          └──────────────────┘
CLEXEC1  00332          line1
CLEXEC1  00334          line2
CLEXEC1  00337          ...
CLEXEC1  00339          ┌──────────────────┐
CLEXEC1  00340          │ &END │           │
CLEXEC1  00341          └──────────────────┘

CLEXEC1  00344          Type 'line1', 'line2', ... at the console, without
CLEXEC1  00346          tokenizing them. The lines are normally truncated at
CLEXEC1  00346          column 72, but this can be avoided by specifying the
CLEXEC1  00347          option 'ALL'. The statement is terminated by a line
CLEXEC1  00349          which the the string '&END' starts in column 1.


CLEXEC1  00352             Note: The Logical Line End character is a
CLEXEC1  00353             hexadecimal '15'. CMS routines will not process the
CLEXEC1  00354             symbolic line end character '#'.


CLEXEC1  00358          ┌──────────────────┐
CLEXEC1  00359          │ &CONTINUE │      │
CLEXEC1  00360          └──────────────────┘

CLEXEC1  00363          A no-op used in conjunction with an EXEC label (for
CLEXEC1  00364          example, -LAB &CONTINUE) to provide a branch address
CLEXEC1  00365          for &ERROR, &GOTO, and other conditional branching
CLEXEC1  00366          statements.


CLEXEC1  00374          ┌──────────────────────────────────────────┐
CLEXEC1  00375          │           │ ┌   ┐ ┌    ┐ ┌    ┐        │
CLEXEC1  00376          │ &CONTROL  │ │OFF│ │TIME│ │PACK│        │
CLEXEC1  00377          │           │ │ERR│ │NTIM│ │NPAC│        │
CLEXEC1  00378          │           │ │CMS│ └    ┘ └    ┘        │
CLEXEC1  00379          │           │ │ALL│                      │
CLEXEC1  00380          │           │ └   ┘                      │
CLEXEC1  00381          └──────────────────────────────────────────┘

CLEXEC1  00385          Set, until further notice, the characteristics of the
CLEXEC1  00386          summary of execution, which is automatically printed at
CLEXEC1  00387          the console.

CLEXEC1  00390          OFF       Do not type CMS commands as they are executed
CLEXEC1  00391                    in this EXEC procedure, nor any return codes
CLEXEC1  00392                    which may result.

CLEXEC1  00394          ERROR     Type out CMS commands which result in a
CLEXEC1  00395                    non-zero return code, and type the return
CLEXEC1  00396                    code.
```

```
CLEXEC1  00398          CMS       Type out each CMS command as it is executed,
CLEXEC1  00400                    and type the return code if it is not zero.

CLEXEC1  00402          ALL       Type out every executable statement as it is
CLEXEC1  00403                    executed, and type any non-zero return codes
CLEXEC1  00404                    from CMS commands.

CLEXEC1  00406          TIME      Include the time-of-day value with each CMS
CLEXEC1  00407                    command printed in the summary.  Effective
CLEXEC1  00408                    only if CMS or ALL is set.

CLEXEC1  00410          NOTIME    Do not include the time-of-day value with CMS
CLEXEC1  00411                    commands.


CLEXEC1  00416          PACK      Pack the lines of the summary so that surplus
CLEXEC1  00418                    blanks are removed.

CLEXEC1  00421          NOPACK    Do not pack the lines of the summary.

CLEXEC1  00424          On entry to an EXEC file, the default settings for the
CLEXEC1  00425          file are:

CLEXEC1  00428               CMS  NOTIME  PACK

CLEXEC1  00430          Each group of options remains set until explicitly
CLEXEC1  00431          reset.


CLEXEC1  00439          ┌──────────┬──────────┐
CLEXEC1  00440          │ &ERROR   │ action   │
CLEXEC1  00441          └──────────┴──────────┘

CLEXEC1  00445          Until instructed otherwise, execute 'action' following
CLEXEC1  00446          any CMS command which yields an error return code (that
CLEXEC1  00447          is, a return code which is not zero).  The action may
CLEXEC1  00447          be any executable statement.  What happens next depends
CLEXEC1  00448          upon the type and consequences of the action.  If it is
CLEXEC1  00449          itself a CMS command which also yields an error return
CLEXEC1  00450          code, then the EXEC interpreter types an error message
CLEXEC1  00450          and exits from the file; otherwise (unless the action
CLEXEC1  00451          causes a transfer of control) execution resumes at the
CLEXEC1  00452          line following the CMS command which caused the action
CLEXEC1  00452          to be executed.  On entry to an EXEC file, the action
CLEXEC1  00454          is set to the null statement (that is, &CONTINUE).


CLEXEC1  00462          ┌──────────┬──────────────┐
CLEXEC1  00463          │          │ ┌          ┐ │
CLEXEC1  00464          │ &EXIT    │ │return-code│ │
CLEXEC1  00465          │          │ │    0     │ │
CLEXEC1  00466          │          │ └          ┘ │
CLEXEC1  00467          └──────────┴──────────────┘
```

```
CLEXEC1  00472          Exit from the EXEC file with the given return code.

CLEXEC1  00482          ┌──────────────────────────────────────────────────┐
CLEXEC1  00483          │ &IF │ {tok1} {EQ} {tok2}                         │
CLEXEC1  00484          │     │ {&$  } {NE} {&$  }                         │
CLEXEC1  00485          │     │ {&*  } {LT} {&*  } executable statement    │
CLEXEC1  00486          │     │        {LE}                                │
CLEXEC1  00487          │     │        {GT}                                │
CLEXEC1  00488          │     │        {GE}                                │
CLEXEC1  00489          └──────────────────────────────────────────────────┘

CLEXEC1  00493          If    the    condition    is    satisfied,    execute
CLEXEC1  00494          'executable-statement'; otherwise proceed to the next
CLEXEC1  00495          line.  &$ means 'any of the arguments &1, &2, ..., &n',
CLEXEC1  00495          and &* means 'all of the arguments &1, &2, ..., &n',
CLEXEC1  00496          where n = &INDEX. The comparison is numeric if both of
CLEXEC1  00497          the comparands are numeric; otherwise they are treated
CLEXEC1  00498          as character strings.

CLEXEC1  00508          ┌──────────────────────┐
CLEXEC1  00509          │ &GOTO │ {TOP         } │
CLEXEC1  00510          │       │ {line-number } │
CLEXEC1  00511          │       │ {label       } │
CLEXEC1  00512          └──────────────────────┘

CLEXEC1  00516          Transfer control to the top of the EXEC file, to the
CLEXEC1  00518          given line, or to the line starting with 'label'.

CLEXEC1  00520          The first character of a label must be a hyphen (minus
CLEXEC1  00520          sign).  A label may be attached to any executable
CLEXEC1  00521          statement, as the first token of the line. Scanning
CLEXEC1  00522          for a label starts on the line following the &GOTO
CLEXEC1  00523          statement, and (if unsuccessful) ends on the line above
CLEXEC1  00524          it.

CLEXEC1  00533          ┌──────────────────────────────┐
CLEXEC1  00534          │       │                ┌   ┐ │
CLEXEC1  00535          │ &LOOP │ (n     ) │(m         )│ │
CLEXEC1  00536          │       │ (label)  │(condition)│ │
CLEXEC1  00537          │       │                └   ┘ │
CLEXEC1  00538          └──────────────────────────────┘

CLEXEC1  00542          Loop through the following n lines, or down to (and
CLEXEC1  00543          including) the line starting with 'label', for m times,
CLEXEC1  00544          or until 'condition' is satisfied.

CLEXEC1  00545          The values of n and m  (if given) must be numeric; also
CLEXEC1  00547          n must be positive, and m must not be negative.

CLEXEC1  00548          The first character  of the label (if given)  must be a
```

minus sign, and the label must be attached, as the first token of the line, to an executable statement which lies below the &LOOP statement.

The form of the condition (if given) is the same as in the &IF statement, that is:

```
     {tok1}    {EQ}    {tok2}
     {&$  }    {NE}    {&$  }
     {&*  }    {LT}    {&*  }
               {LE}
               {GT}
               {GE}
```

```
| &PUNCH | tok1  [tok2 ... ]  |
```

Punch a card containing 'tok1', 'tok2', ... The card will be padded with blanks, or truncated, as necessary to fill an eighty-column card. The tokens, as punched, will be separated from each other by a single blank (see &BEGPUNCH).

```
|        | r                          |
| &READ  | n                          |
|        | 1                          |
|        | ARGS                       |
|        | VARS  [var1 [var2...]]     |
|        | L                          |
```

Read the next n lines from the terminal and treat them as if they had been in the EXEC file; or read a single line, assign the tokens in it to the arguments &1, &2, ..., and reset &INDEX to the number of arguments thus set; or read a single line and assign the tokens in it to the variables 'var1', 'var2', ...

If n is given, reading from the terminal stops when n lines have been read, or when a &LOOP statement or a statement which transfers control is encountered. If an &READ statement is encountered, the number of lines to be read by it is added to the number outstanding.

The variables 'var1', 'var2', ..., if given, are scanned in the same way as if they appeared on the left-hand side of an assignment statement (see &ARGS).

```
┌──────────────────┐
│        │ ┌   ┐   │
│ &SKIP  │ │ n │   │
│        │ │ 1 │   │
│        │ └   ┘   │
└──────────────────┘
```

If n > 0, skip the next n lines of the EXEC file. If n < 0, transfer control to the line which is -n lines above the current line. If n = 0, transfer control to the next line. If during the SKIP operation, end-of-file is reached, the EXEC file will exit with return to CMS. If n specifies a position before the beginning of the EXEC file, an &SKIP error will result.

```
┌───────────────────┐
│        │ ┌   ┐    │
│ &SPACE │ │ n │    │
│        │ │ 1 │    │
│        │ └   ┘    │
└───────────────────┘
```

Type n blank lines at the terminal.

```
┌─────────────────────────────────────────┐
│        │ ┌    ┐                         │
│ &STACK │ │FIFO│  [tok1  [tok2 ... ]]    │
│        │ │LIFO│                         │
│        │ └    ┘                         │
└─────────────────────────────────────────┘
```

Stack a line in the terminal input buffer containing 'tok1', 'tok2', ..., or stack a null line if the tokens are absent. The line is normally stacked 'FIFO' (first in, first out), but this can be changed by specifying the option 'LIFO' (last in, first out). The tokens, as stacked, will be separated from each other by a single blank.

Note: The Logical Line End character is a hexadecimal '15'. CMS routines will not process the symbolic character '#' as a logical line end character, that is, will not translate it to X'15' unless the user follows the following procedures:

1. changes or turns off the logical line end character via the CP command, TERMINAL.

2. uses EDIT's ALTER subcommand to cause the EDITOR to automatically convert the character "#" or some other character to X'15'.

```
┌───────────────────────────────────────┐
│        │ ┌     ┐   ┌       ┐          │
│ &TIME  │ │ON   │   │RESET  │          │
│        │ │OFF  │   │TYPE   │          │
│        │ └     ┘   └       ┘          │
└───────────────────────────────────────┘
```

This statement can be used to type timing information in the form:

T=x.xx/y.yy hh:mm:ss

where

x.xx	is the virtual CPU time used since it was last reset in the current EXEC file,
y.yy	is the total CPU time used since it was last reset in the current EXEC file, and
hh:mm:ss	is the time of day in hours:minutes:seconds.

The CPU times are set to zero before the execution of the first statement in the EXEC file, and are reset whenever the timing information is printed.

ON	Reset the CPU times before every CMS command, and print the timing information on return. If &CONTROL is set to CMS or ALL, the printing of the timing information is followed by a blank line.
OFF	Do not automatically reset the CPU times before every CMS command, or print the timing information on return. This is the initial setting.
RESET	Perform an immediate reset of the CPU times.
TYPE	Type the current timing information (and reset the CPU times).

```
┌──────────────────────────────────┐
│ &TYPE │ tok1  [tok2  ... ]       │
└──────────────────────────────────┘
```

Print at the terminal a line containing 'tok1', 'tok2', ... The tokens, as typed, are separated from each other by a single blank (see &BEGTYPE).

CLEXEC2 00003 **Built-in Functions**

CLEXEC2 00005 An EXEC built-in function consists of the name of the
CLEXEC2 00006 function and, usually, a list of arguments. Built-in
CLEXEC2 00007 function names are EXEC keywords, and start with an
CLEXEC2 00007 ampersand. With the exception of &LITBAL, they are
CLEXEC2 00008 recognized only if they appear as the token following the
CLEXEC2 00009 equal sign of an assignment statement. The details are as
CLEXEC2 00010 follows.

CLEXEC2 00020 ┌──────────────────────────────┐
CLEXEC2 00021 | &CONCAT | tok1 [tok2 ...] |
CLEXEC2 00022 └──────────────────────────────┘

CLEXEC2 00027 Concatenate 'tok1', 'tok2', ..., into a single token, with a
CLEXEC2 00027 maximum length of eight. This function is recognized only
CLEXEC2 00028 on the right-hand side of an assignment statement. For
CLEXEC2 00029 example:

```
CLEXEC2 00032          &A = **
CLEXEC2 00035          ...
CLEXEC2 00037          &B = &CONCAT XX &A 45
CLEXEC2 00039          &TYPE &B
```

CLEXEC2 00042 This results in the printed line:

```
CLEXEC2 00044          XX**45
```

CLEXEC2 00053 ┌──────────────────┐
CLEXEC2 00054 | &DATATYPE | tok |
CLEXEC2 00055 └──────────────────┘

CLEXEC2 00058 Has the value NUM or CHAR, depending on the data type of
CLEXEC2 00059 'tok'. This function is recognized only on the right-hand
CLEXEC2 00060 side of an assignment statement.

CLEXEC2 00063 ┌─────────────────┐
CLEXEC2 00064 | &LENGTH | tok |
CLEXEC2 00065 └─────────────────┘

CLEXEC2 00068 Gives the number of non-blank characters in 'tok'. This
CLEXEC2 00069 function is recognized only on the right-hand side of an
CLEXEC2 00070 assignment statement.

```
CLEXEC2  00073     ┌──────────────────┐
CLEXEC2  00074     | &LITERAL | tok   |
CLEXEC2  00075     └──────────────────┘

CLEXEC2  00079   Use the literal value of 'tok', without substitution for any
CLEXEC2  00080   EXEC variable which may appear in it.  This function is
CLEXEC2  00080   recognized anywhere in an executable statement.  For
CLEXEC2  00081   example:

CLEXEC2  00084        &X = **
CLEXEC2  00086        &TYPE &LITERAL &X EQUALS &X

CLEXEC2  00089   This results in the printed line:

CLEXEC2  00091        &X EQUALS **


CLEXEC2  00097     ┌──────────────────────┐
CLEXEC2  00098     | &SUBSTR | tok  i  [j] |
CLEXEC2  00099     └──────────────────────┘

CLEXEC2  00105   Extract that part of 'tok' which starts at character i, with
CLEXEC2  00105   length j; or which starts at character i and runs to the end
CLEXEC2  00106   of the token.  The value of i (and j if given) must be
CLEXEC2  00107   numeric; also i must be positive, and j must not be
CLEXEC2  00108   negative.  This function is recognized only on the
CLEXEC2  00109   right-hand side of an assignment statement.

CLEXEC2  00112        &A = &SUBSTR ABCDE 2 3
CLEXEC2  00114        &TYPE &A

CLEXEC2  00117   This results in the printed line:

CLEXEC2  00119        BCD
```

CLEXEC2 00122 **Special Variables**

CLEXEC2 00125 Certain EXEC variables are reserved for special purposes.

CLEXEC2 00128 &0 Initialized to the filename of the EXEC file
CLEXEC2 00129 being executed.

CLEXEC2 00131 &1, &2, ... Initialized to the arguments 'arg1', 'arg2',
CLEXEC2 00133 ..., which are passed to the EXEC file.

CLEXEC2 00135 &INDEX Initialized to the number of arguments passed to
CLEXEC2 00136 the EXEC file. Can take only integral numeric
CLEXEC2 00137 values.

CLEXEC2 00139 &RETCODE Set after every CMS command to the return
CLEXEC2 00140 (completion) code of the CMS command. Can take
CLEXEC2 00141 only integral numeric values.

CLEXEC2 00143 &LINENUM Contains the current line number of the EXEC
CLEXEC2 00145 file. Cannot be set explicitly.

CLEXEC2 00147 &GLOBAL Contains the recursion level of the EXEC
CLEXEC2 00148 interpreter (e.g. 1, 2, ...). Cannot be set
CLEXEC2 00149 explicitly.

CLEXEC2 00152 &GLOBAL0, &GLOBAL1, ... &GLOBAL9

CLEXEC2 00154 Can hold only integral numeric values. The are
CLEXEC2 00155 initially set to one. Unlike other EXEC
CLEXEC2 00156 variables, these can be used to communicate
CLEXEC2 00157 between different levels of the EXEC
CLEXEC2 00158 interpreter.

CLEXEC2 00160 &READFLAG Contains the value CONSOLE or STACK, depending
CLEXEC2 00161 on whether an attempt to read from the console
CLEXEC2 00161 would obtain a physical line from the real
CLEXEC2 00162 console, or a logical line from the console
CLEXEC2 00163 input stack. Cannot be set explicitly.

CLEXEC2 00165 &TYPEFLAG Has the setting RT or HT, depending on the value
CLEXEC2 00166 of the console output flag. Cannot be set
CLEXEC2 00167 explicitly.

CLEXEC2 00171 **Miscellaneous Notes**

CLEXEC2 00173 Columns of truncation: Except where otherwise stated, data
CLEXEC2 00174 lines read from an EXEC file are truncated at column 72, and
CLEXEC2 00177 lines read from the console are truncated at column 130.

CLEXEC2 00179 Setting an argument to blanks: An argument can be set to
CLEXEC2 00180 blanks by assigning it a percent sign (%) when invoking the
CLEXEC2 00181 EXEC file, in the &ARGS statement, or in the &READ ARGS
CLEXEC2 00182 statement. A variable can also be set to blanks by
CLEXEC2 00183 assigning it to the double-quotation mark in the &READ VARS
CLEXEC2 00184 statement. For example:

```
CLEXEC2 00186          &ARGS A % B
CLEXEC2 00187          ...
CLEXEC2 00188          &TYPE &1 &2 ** &3
```

CLEXEC2 00190 This results in the printed line:

```
CLEXEC2 00192          A ** B
```

CLEXEC2 00197 Missing end-token: A final token of blanks is added to any
CLEXEC2 00198 EXEC statement which is syntactically invalid if in doing so
CLEXEC2 00200 the statement becomes syntactically valid. For example:

```
CLEXEC2 00203      &BLANK =
CLEXEC2 00205      &TYPE
CLEXEC2 00207      &LOOP 3 &X NE
```

CLEXEC2 00210 Syntax note: An EXEC statement is interpreted anew each time
CLEXEC2 00211 it is executed. It follows that the same line may be
CLEXEC2 00212 executed as an entirely different statement on different
CLEXEC2 00213 occasions. For example:

```
CLEXEC2 00216      &ARGS -XX &GOTO DUMMY &LITERAL &LITERAL EQ " XX &IF
CLEXEC2 00217      &SKP = 0
CLEXEC2 00218      -L &8 &7 &6 &5 &4 &3 &2 &1
CLEXEC2 00219      &SKIP &SKP
CLEXEC2 00220      &ARGS SYSPRINT FILE TYPE
CLEXEC2 00221      &SKP = 3
CLEXEC2 00222      &GOTO -L
```

CLEXEC2 00227 The statement labelled -L will be executed the first time
CLEXEC2 00228 as:

```
CLEXEC2 00230      &IF XX EQ &LITERAL &GOTO -XX
```

CLEXEC2 00232 and the second time as:

```
CLEXEC2 00234      TYPE FILE SYSPRINT
```

CLEXEC2 00235 Note, however, the exceptions given below under 'Assignment
CLEXEC2 00237 statement' and '&LOOP statement'.

CLEXEC2 00240 Assignment statement: The token immediately following the
CLEXEC2 00241 target of an assignment statement must be a literal equal
CLEXEC2 00242 sign, and not an EXEC variable which has the value of an

CLEXEC2 00242 equal sign. Conversely, if an equal sign is to be the first
CLEXEC2 00243 token following a control word, then it must be given as an
CLEXEC2 00244 EXEC variable which has the value of an equal sign, and not
CLEXEC2 00245 as a literal equal sign; otherwise the statement will be
CLEXEC2 00246 interpreted as an assignment, and (if it is valid as such)
CLEXEC2 00247 the control word will thereafter be treated as a variable.

CLEXEC2 00249 &LOOP statement: The decision on which form of the &LOOP
CLEXEC2 00250 statement is intended is based on the number of tokens in
CLEXEC2 00250 the statement after scanning and substitution for any EXEC
CLEXEC2 00251 variables. If the conditional form is intended, and the
CLEXEC2 00252 first comparand or the comparator is given in the form of an
CLEXEC2 00253 EXEC variable, then the value of the variable must not be
CLEXEC2 00253 blank at the time the statement is interpreted. In the
CLEXEC2 00254 conditional form, the tokens forming the conditional phrase
CLEXEC2 00255 are saved in an unscanned format, so that substitution for
CLEXEC2 00256 any EXEC variables can be performed dynamically before each
CLEXEC2 00257 execution of the loop. For example:

CLEXEC2 00260 a. &X = 0
CLEXEC2 00261 &LOOP 2 &X EQ 2
CLEXEC2 00262 &X = &X + 1
CLEXEC2 00263 &TYPE &X

CLEXEC2 00265 This results in the printed lines:

CLEXEC2 00267 1
CLEXEC2 00268 2

CLEXEC2 00270 b. &X = XYZ
CLEXEC2 00271 &Y = &LITERAL A&B
CLEXEC2 00272 &LOOP 2 &X EQ &LITERAL &
CLEXEC2 00273 &X = &SUBSTR &Y 2 1
CLEXEC2 00274 &TYPE &X

CLEXEC2 00276 This will result in the single printed line:

CLEXEC2 00278 &

CLEXEC2 00283 &ERROR statement: The words following the &ERROR control
CLEXEC2 00284 word are saved in an unscanned format, and substitution for
CLEXEC2 00285 any variables among them is performed dynamically (if the
CLEXEC2 00285 occasion arises) after obtaining a non zero (error)
CLEXEC2 00287 completion code from a subsequent CMS command.

CLEXEC2 00289 Search for labels: The search for a label involves
CLEXEC2 00290 examination of only the first word on each line. It follows
CLEXEC2 00291 that care must be taken to avoid the use of any label names
CLEXEC2 00291 which may appear as the first word of a line within the
CLEXEC2 00293 scope of a &BEGTYPE, &BEGPUNCH or &BEGSTACK statement.

CLEXEC2 00295 Removal of leading zeros: Leading zeros can be removed from a numeric quantity by performing any arithmetic operation on
CLEXEC2 00296 it. For example:
CLEXEC2 00298

```
CLEXEC2 00301        &X = 000012
CLEXEC2 00303        &TYPE &X
CLEXEC2 00305        &X = &X + 0
CLEXEC2 00307        &TYPE &X
```

CLEXEC2 00310 This results in the printed lines:

```
CLEXEC2 00312        000012
CLEXEC2 00314        12
```

CLEXEC2 00317 Implementation Restrictions:

CLEXEC2 00321 1. No more than 30 arguments can be passed to an EXEC
CLEXEC2 00322 file.

CLEXEC2 00325 2. No more than 19 tokens can appear in an EXEC statement.

CLEXEC2 00328 3. No more than 4 nested loops may be specified.

CLEXEC2 00331 4. No more than 19 levels of recursion can be handled by
CLEXEC2 00332 the EXEC interpreter.

CLEXEC2 00336 Responses

CLEXEC2 00338 As each CMS command in the EXEC file is processed, it is
CLEXEC2 00339 typed at the terminal along with any non-zero return code.
CLEXEC2 00340 The &CONTROL command can be used to augment or reduce the
CLEXEC2 00341 amount of typing done during execution.

CLEXEC2 00342 If the EXEC interpreter finds an error, it will type the
CLEXEC2 00343 message:

CLEXEC2 00345 ERROR IN EXEC FILE fn, LINE n - description of error

CLEXEC2 00346 where "description of error" will be one of the following
CLEXEC2 00348 conditions with its appropriate return code:

```
CLEXEC2 00352        (802)  &SKIP OR &GOTO ERROR
CLEXEC2 00354        (804)  TOO MANY ARGUMENTS
CLEXEC2 00356        (805)  MAX DEPTH OF LOOP NESTING EXCEEDED
CLEXEC2 00358        (806)  DISK OR TERMINAL READ ERROR
CLEXEC2 00360        (807)  INVALID SYNTAX
CLEXEC2 00362        (808)  INVALID FORM OF CONDITION
```

```
CLEXEC2  00364          (809)  INVALID ASSIGNMENT
CLEXEC2  00366          (810)  MISUSE OF SPECIAL VARIABLE
CLEXEC2  00368          (811)  ERROR IN &ERROR ACTION
CLEXEC2  00370          (812)  CONVERSION ERROR
CLEXEC2  00372          (813)  TOO MANY TOKENS IN STATEMENT
CLEXEC2  00374          (814)  MISUSE OF BUILT-IN FUNCTION
CLEXEC2  00376          (815)  EOF FOUND IN LOOP
CLEXEC2  00378          (816)  INVALID CONTROL WORD
```

CLFILEDE 00005 **FILEDEF**

CLFILEDE 00008 The FILEDEF command is used to allow the OS simulation
CLFILEDE 00009 routines to simulate the functions of the JCL Data
CLFILEDE 00010 Definition (DD) card. It provides device independence by
CLFILEDE 00011 allowing the user to specify the I/O device as well as file
CLFILEDE 00012 characteristics to be used by a program at execution time.
CLFILEDE 00012 FILEDEF may be used only with programs using OS macros and
CLFILEDE 00013 functions.

CLFILEDE 00014 If a FILEDEF is not issued for a file within an CS program,
CLFILEDE 00016 a default FILEDEF is issued by CMS. Its format is

CLFILEDE 00018 FILEDEF ddname DISK FILE ddname A1

CLFILEDE 00020 where ddname is the ddname in the DCB in the program.

```
┌──────────┬────────────┬─────────────────────────────────────────────┬──────────────┐
│ FILEDEF  │ ┌ddname┐   │ ┌    TERMINAL [(optionA*]                 ┐ │              │
│ FI       │ <nn     >  │ │                                         │ │              │
│          │ │*      │  │ │                                         │ │              │
│          │ └       ┘  │ │                                         │ │              │
│          │            │ │    PRINTER                              │ │              │
│          │            │ │    PUNCH                                │ │              │
│          │            │ │    READER                               │ │              │
│          │            │ <                                         > [(optionD*]│
│          │            │ │                                         │ │              │
│          │            │ │ {  DISK [fn ft [fm] [(optionB*]]        │ │              │
│          │            │ │ {  DUMMY                                }│ │              │
│          │            │ │    TAPi  [(optionC*]                    │ │              │
│          │            │ │    CLEAR                                │ │              │
│          │            │ └                                         ┘ │              │
│          │            │                                              │              │
│          │   *optionA: [UPCASE|LOWCASE]                                             │
│          │            │                                                             │
│          │   *optionB: [KEYLEN n]                                                   │
│          │             [XTENT  {n|50}]                                              │
│          │             [LIMCT  n]                                                   │
│          │             [OPTCD  {A|E|F|R}]                                           │
│          │             [DISP MOD]                                                   │
│          │                                                                          │
│          │   *optionC: [7TRACK|9TRACK]                                              │
│          │             [TRTCH{O|OC|OT|E|ET}]                                        │
│          │             [DEN    {200|556|800|1600}]                                  │
│          │                                                                          │
│          │   *optionD: [PERM]                                                       │
│          │             [CHANGE|NOCHANGE]                                            │
│          │             [RECFM {F|FB|V|VB|U|VS|VBS|FS|FBS|A|M}]                      │
│          │             [LRECL n]                                                    │
│          │             [BLOCK n]                                                    │
└──────────┴──────────────────────────────────────────────────────────────────────────┘
```

If no parameters are entered with the command, a list of current filetypes is typed at the user's terminal including the ddname; device type and; if device type is DISK, the filename and filetype.

ddname
nn
: specifies the name by which the file is referred to in the user's program. If a number nn is specified, it is translated to a FORTRAN* data definition name of FTnnF001. If the "CLEAR" option is in effect, ddname may be specified as "*" to indicate that all file definitions not entered with the "PERM" option are to be removed.

----

* The FORTRAN processors are IBM Program Products.

```
CLFILEDE 00096   Devices

CLFILEDE 00100   TERMINAL  user terminal (terminal I/O may not be blocked)

CLFILEDE 00103   DISK      disk

CLFILEDE 00105   PRINTER   spooled printer

CLFILEDE 00107   PUNCH     spooled punch

CLFILEDE 00109   READER    spooled card reader (card reader I/O may not be
CLFILEDE 00110             blocked)

CLFILEDE 00113   TAPi      magnetic tape (i is the symbolic number of the
CLFILEDE 00114             tape. It is either 1, 2, 3, or 4 representing
CLFILEDE 00115             virtual units 187, 182, 183 and 184 respectively.)

CLFILEDE 00118   DUMMY     indicates that no real I/O is to take place for a
CLFILEDE 00119             disk data set.

CLFILEDE 00122   CLEAR     removes any existing definition for the specified
CLFILEDE 00123             ddname.

CLFILEDE 00126   If the device is DISK, the CMS fileid may be specified. If
CLFILEDE 00126   none is entered, the default file identification is "FILE
CLFILEDE 00127   ddname A1."

CLFILEDE 00132   fn        CMS filename for the file referred to by ddname

CLFILEDE 00135   ft        CMS filetype for the file referred to by ddname

CLFILEDE 00138   fm        CMS filemode for the file referred to by ddname


CLFILEDE 00141   Options

CLFILEDE 00142   If an option is specified that is invalid for the device
CLFILEDE 00144   type, an error message is issued. Figure 17 shows valid
CLFILEDE 00145   options for each device type.

CLFILEDE 00149   UPCASE          indicates that all terminal input data is to
CLFILEDE 00150                  be translated to uppercase.

CLFILEDE 00153   LOWCASE         indicates that all terminal input data is to
CLFILEDE 00153                  remain as typed in, and not translated on
CLFILEDE 00154                  output.

CLFILEDE 00156   KEYLEN nn       specifies the size of the key (nn bytes).
CLFILEDE 00158                  This option is used only for BDAM files.

CLFILEDE 00160   XTENT nn        specifies the number of records in the extent
CLFILEDE 00161                  for the file (nn bytes). This option is used
CLFILEDE 00162                  only for BDAM files.
```

LIMCT nn — specifies the number of extra tracks or blocks to be searched (nn bytes). This option is used for BDAM files.

OPTCD a — specifies the direct access search processing desired. a may be any combination of up to three of the following: (A and R are mutually exclusive.)

- A Actual device addressing
- E Extended search
- F Feedback addressing
- R Relative block addressing

DISP MOD — indicates that the read/write pointer is to be positioned after the last record in the disk file.

7TRACK / 9TRACK — Indicates tape setting.

TRTCH i — specifies Tape Recording Technique. i can be determined by referring to the following chart:

i	PARITY	CONVERTER	TRANSLATOR
OC	odd	on	off
OT	odd	off	on
O	odd	off	off
ET	even	off	on
E	even	off	off

DEN den — specifies tape density: den can be 200, 556, 800, or 1600 bytes per inch. If 200 or 556 is specified, 7TRACK is assumed. If 800 or 1600 is specified 9TRACK is assumed.

PERM — specifies that this definition is to remain in effect until it is either explicitly cleared or has been changed with a new FILEDEF with the CHANGE option. If PERM is not specified, the definition will be cleared when a FILEDEF * CLEAR is executed.

CHANGE — specifies that if a file definition exists

```
CLFILEDE 00238                  for the specified ddname, the new file
CLFILEDE 00239                  definition is to replace the old one.

CLFILEDE 00242    NOCHANGE      specifies that if a file definition exists
CLFILEDE 00242                  for the specified ddname, it is not to be
CLFILEDE 00243                  replaced.

CLFILEDE 00245    RECFM a       the record format of the file is specified as
CLFILEDE 00247                  a, where a is one of the following:

CLFILEDE 00249                  F       fixed
CLFILEDE 00251                  FB      fixed blocked
CLFILEDE 00253                  V       variable
CLFILEDE 00255                  VB      variable blocked
CLFILEDE 00257                  U       undefined
CLFILEDE 00259                  FS,FBS  fixed length, standard blocks
CLFILEDE 00261                  VS,VBS  variable length, spanned records
CLFILEDE 00263                  *A      ASA print control characters
CLFILEDE 00265                  *M      machine print control codes

CLFILEDE 00269    * A and M may be used in conjunction with any of the valid
CLFILEDE 00270    RECFM settings, (for example, FA, FBA, VA, VBA, etc.)

CLFILEDE 00273    LRECL nn      logical record length of the file (nn bytes).
CLFILEDE 00274                  LRECL should not exceed 32,767 bytes because
CLFILEDE 00275                  of OS restrictions.

CLFILEDE 00278    BLOCK nn      logical block of the file (nn bytes).
CLFILEDE 00279                  BLOCK should not exceed 32,767 bytes because
CLFILEDE 00280                  of OS restrictions. BLOCK should be not used
CLFILEDE 00281                  with TERMINAL or READER devices.
```

OPTION	UNIT RECORD PRINTER/PUNCH READER	TERMINAL	DUMMY	TAPE	DISK
UPCASE		X	No		
LOWCASE		X	Options		
LRECL	X	X	Nec-	X	X
BLOCK	X	X	ssary	X	X
RECFM	X		But	X	X
7TRACK/9TRACK			All	X	
TRTCH			Disk	X 7TRK	
			Options	ONLY	
DEN			Are	X	
KEYLEN			Acc-		X (BDAM)
XTENT			epted		X (BDAM)
LIMCT					X (BDAM)
OPTCD					X (BDAM)
DISP MOD					X
PERM	X	X		X	X
NOCHANGE/CHANGE	X	X		X	X

Figure 17. Valid File Characteristics for each Device Type for FILEDEF command

Example 1

```
FILEDEF | SYSPRINT PRINTER (PERM RECFM F LRECL 133)
```

Action: When SYSPRINT is referred to in a user's program, the output written to it is spooled to the virtual printer. The file is fixed-format with a logical record length of 133. The file definition may not be removed by the command FILEDEF * CLEAR.

Example 2

```
FILEDEF | 05   READER  (PERM RECFM F LRECL 80)
```

Action: When FT05F001 is referenced in a program, reading is done from the virtual spooled card reader. The file is fixed-length card image. The file definition may not be removed by the command FILEDEF * CLEAR.

CLFILEDE 00348 Example 3

```
CLFILEDE 00350 ┌──────────────────────────────────────────────────┐
CLFILEDE 00351 | FILEDEF | PRINTOUT TERMINAL (UPCASE NOCHANGE     |
CLFILEDE 00352 └──────────────────────────────────────────────────┘
```

CLFILEDE 00355 Action: If a file definition with the ddname PRINTOUT does
CLFILEDE 00356 not exist, one is established. Output written to PRINTOUT
CLFILEDE 00358 is typed at the terminal in uppercase.

CLFILEDE 00361 Example 4

```
CLFILEDE 00363 ┌──────────────────────────────────────────────────┐
CLFILEDE 00364 | FILEDEF | DISK DISK NAME OLDFILE                 |
CLFILEDE 00365 └──────────────────────────────────────────────────┘
```

CLFILEDE 00368 Action: When I/O is done to a file with a ddname of DISK,
CLFILEDE 00369 the I/O is done on a disk file with a file identifier of
CLFILEDE 00370 NAME OLDFILE.

CLFILEDE 00373 Example 5

```
CLFILEDE 00375 ┌──────────────────────────────────────────────────┐
CLFILEDE 00376 | FILEDEF | 05 CLEAR                              |
CLFILEDE 00377 └──────────────────────────────────────────────────┘
```

CLFILEDE 00380 Action: Any existing file definition for FIO5F001 is
CLFILEDE 00381 removed.

CLFILEDE 00384 Example 6

```
CLFILEDE 00386 ┌──────────────────────────────────────────────────┐
CLFILEDE 00387 | FILEDEF | SYSPUNCH DUMMY                        |
CLFILEDE 00388 └──────────────────────────────────────────────────┘
```

CLFILEDE 00391 Action: When any I/O command is issued to a disk file with a
CLFILEDE 00393 ddname of DISKFILE, the real I/O operation is not performed.

CLFILEDE 00396 Example 7

```
CLFILEDE 00398 ┌──────────────────────────────────────────────────────────────────────────
CLFILEDE 00399 | FILEDEF | NEWMAST TAP2 (9TRACK DEN 1600 RECFM FB LRECL 050 BLOCK 3
CLFILEDE 00400 └──────────────────────────────────────────────────────────────────────────
```

CLFILEDE 00403 Action: I/O commands issued to a file with ddname of NEWMAST
CLFILEDE 00404 are directed to the tape located at logical unit TAP2. The
CLFILEDE 00405 tape is nine-track, and recording is done at 1600 bpi. The
CLFILEDE 00406 tape is in fixed-block format with a logical record length
CLFILEDE 00409 of 50 and a physical blocksize of 3000.

CLFILEDE 00411 **Responses**

```
CLFILEDE 00416   ddname1   device1   [filename1  filetype1]
CLFILEDE 00417    •                                  •
CLFILEDE 00418    •                                  •
CLFILEDE 00419    •                                  •
CLFILEDE 00420   ddnameN   deviceN   [filenameN  filetypeN]

CLFILEDE 00424   A list of current definitions is typed if the FILEDEF
CLFILEDE 00425   command is entered with no operands.


CLFILED3 00428   DMKFLD704I  INVALID CLEAR REQUEST

CLFILEDE 00430   The FILE definition for which a CLEAR request was entered
CLFORMAT 00001   does not exist; no action is taken.
```

CLFORMAT 00004 **FORMAT**

CLFORMAT 00008 The FORMAT command is used to:

CLFORMAT 00011 1. Initialize a minidisk area in the CMS format.
CLFORMAT 00013 2. Count the number of cylinders on a minidisk.
CLFORMAT 00015 3. Write a label on a minidisk.
CLFORMAT 00017 4. Reset the number of cylinders on the minidisk.

CLFORMAT 00020 This command may be used with a virtual 3330, 2314, or 2319
CLFORMAT 00029 direct access storage device.

CLFORMAT 00031 ┌──┐
CLFORMAT 00032 | FORMAT | ccu mode [n'cyl] [(RECOMP)|(LABEL)] |
CLFORMAT 00033 └──┘

CLFORMAT 00040 ccu specifies the virtual device address of the minidisk
CLFORMAT 00041 to be formatted.

CLFORMAT 00043 mode specifies the mode letter to be related to the
CLFORMAT 00044 specified device address. Valid modes are A, B, C,
CLFORMAT 00046 D, E, F, G, Y and Z. This field must be specified.

CLFORMAT 00049 n'cyl specifies the number of cylinders which are to be
CLFORMAT 00049 available for use. If not specified, all cylinders
CLFORMAT 00050 are available.

CLFORMAT 00054 LABEL specifies that a label is to be written on the
CLFORMAT 00059 L disk and no formatting is to take place. This causes
CLFORMAT 00060 a six-character label to be written on cylinder 0,
CLFORMAT 00060 track 0, record 3 of the virtual disk or minidisk.
CLFORMAT 00061 The user will be asked for a six-character disk
CLFORMAT 00061 label (less than six is left-justified, blank
CLFORMAT 00062 padded).

CLFORMAT 00065 RECOMP specifies that the number of cylinders on the disk
CLFORMAT 00069 R which are available to the user is to be changed.
CLFORMAT 00069 N'cyl indicates the new size of available space. If
CLFORMAT 00070 n'cyl is not specified, all cylinders are to be
CLFORMAT 00071 used.

CLFORMAT 00073 Note: If neither RECOMP nor LABEL is specified, the user's
CLFORMAT 00074 disk area is initialized by writing a device-dependent
CLFORMAT 00075 number of records containing binary zeros on each track.
CLFORMAT 00076 Any previous data on the disk is erased. A read after write
CLFORMAT 00077 check is made as the disk is formatted.

Example 1

```
┌──────────────────────────────┐
| FORMAT | 191 A  25           |
└──────────────────────────────┘
```

Action: 25 cylinders of the disk located at virtual address 191 are to be initialized in CMS format.

Example 2

```
┌──────────────────────────────┐
| FORMAT | 192 B  25 (RECOMP)  |
└──────────────────────────────┘
```

Action: The amount of disk space available to the user on the disk located at virtual address 192 is to be changed to 25 cylinders.

Example 3

```
┌──────────────────────────────┐
| FORMAT | 193 C (LABEL)       |
└──────────────────────────────┘
```

Action: A label is to be written on the disk at virtual address 193. The user will be requested to enter a six-character label.

Responses

DMSFMT603R FORMAT WILL ERASE ALL FILES ON DISK 'mode (ccu)'
DO YOU WISH TO CONTINUE?

The user has indicated that a disk area is to be initialized which implies that any existing files will be erased. This message gives him the option of cancelling the execution of the FORMAT command. He should reply YES or NO.

DMSFMT705I DISK REMAINS UNCHANGED

The user replied 'NO' to the above message.

DMSFMT605R ENTER DISK LABEL:

The user has requested that a label is to be written on the disk. He should enter a 1-6 character label.

```
CLFORMAT 00140    DMSFMT732I  'nnn' CYL FORMATTED ON DISK 'mode (ccu)'

CLFORMAT 00142    A formatting operation ahs been done on nnn cylinders of the
CLFORMAT 00143    disk at virtual address ccu.

CLFORMAT 00145    DMSFMT733I  FORMATTING DISK 'mode'

CLFORMAT 00146    The disk represented by mode letter 'mcde' is being
CLFORMAT 00147    formatted.

CLFORMAT 00151    DISK 'mode (ccu)': 'n' FILES, 'n' BLOCKS, 'n' LEFT (OF 'n'),
CLFORMAT 00152    nn% FULL ('n' CYL)

CLFORMAT 00154    This message gives the extent  and other infcrmaticn about a
CLGENDIR 00001    disk when a RECOMP operation has been done.
```

GENDIRT

The GENDIRT command is for use in creating CMS commands in which the number of modules involved would increase the size of the resident directory significantly, and thus increase search time and storage requirements. By using GENDIRT to create an auxiliary directory, the file entries for the given command will be loaded only when the command is invoked.

```
┌──────────────────────────────────┐
| GENDIRT | directory-name         |
└──────────────────────────────────┘
```

directory-name is the entry point of the auxiliary directory.

Creation of Auxilliary Directory

The auxiliary directory TEXT deck is obtained by assembling a set of DMSFST macros, one for each module name. The format of the DMSFST macro is as follows:

```
        DMSFST   module-name [,alias-name]
```

module-name indicates the name of the module whose File Status Table (FST) information is to be copied.

alias-name indicates another name by which the module is to be known.

An example is shown below for the VM/370 System Assembler.

```
ASMDIRT  START  0
         DC     F '40'   Length of FST entry
         DC     A (DIRTEND-DIRTBEG)  Size of directory
DIRTBEG  EQU    *
         DMSFST IEUASM
         DMSFST IEUMAC
         DMSFST IEUFI
         DMSFST IEUF2
         DMSFST IEUF3
         DMSFST IEURTA
         DMSFST IEUF7
         DMSFST IEUFI
         DMSFST IEUF8
         DMSFST IEUFPP
```

```
CLGENDIR 00069                      DHSPST IEUERR
CLGENDIR 00070                      DC     2A(0)   Pointer to next PST block
CLGENDIR 00071           DIRTEND    EQU    *
CLGENDIR 00072                      END
```

CLGENMOD 00003 **GENMOD**

CLGENMOD 00005 The GENMOD command is used to generate absclute core-image
CLGENMOD 00013 files.

CLGENMOD 00015
CLGENMOD 00016 | GENMOD | [fn [ft [fm]]] [(options...)] |
CLGENMOD 00017 | G | |
CLGENMOD 00018 | | options: [MAP\|NOMAP] |
CLGENMOD 00019 | | [FROM entry1] |
CLGENMOD 00020 | | [TO entry2] |
CLGENMOD 00021 | | [STR\|NOSTR] |
CLGENMOD 00022 | | [SYSTEM] |
CLGENMOD 00023

CLGENMOD 00028 fn specifies the filename of the MODULE file being
CLGENMOD 00029 created. If fn is not specified, the file
CLGENMOD 00030 created will have a filename equal tc that cf
CLGENMOD 00031 the first entry point in the LOAD MAP.

CLGENMOD 00034 ft specifies the filetype of the MCDULE file being
CLGENMOD 00036 created. Must be MODULE if specified.

CLGENMOD 00039 fm specifies the filemode of the MODULE file being
CLGENMOD 00040 created. If not specified, the file is written
CLGENMOD 00041 on the user's primary read-write disk.

CLGENMOD 00044 **Options**

CLGENMOD 00045 If conflicting options are specified, the last cne entered
CLGENMOD 00046 will be used.

CLGENMOD 00048 FROM entry1 specifies an entry point or a control section
CLGENMOD 00049 name indicating the starting main storage
CLGENMOD 00049 location from which the core-image copy is to
CLGENMOD 00050 be generated.

CLGENMOD 00053 TO entry2 specifies an entry point or a control section
CLGENMOD 00054 name indicating the ending main storage lccatien
CLGENMOD 00054 from which the core-image copy is to be
CLGENMOD 00055 generated.

CLGENMOD 00057 **Note**: If 'FROM' is not specified the module consists of
CLGENMOD 00058 storage starting at the point of fn (or first loaded
CLGENMOD 00059 external name). If 'TO' is not specified the module
CLGENMOD 00060 consists of storage to the end of loaded code.

CLGENMOD 00062 **MAP** specifies that a load map is tc be ccntained in
CLGENMOD 00063 the MODULE file.

CLGENMOD 00066 **NOMAP** specifies that a load map is not to be contained
CLGENMOD 00067 in the MODULE file.

CLGENMOD 00070 **STR** indicates that when the MODULE is subsequently
CLGENMOD 00071 loaded (see the CMS LOADMOD command) the CMS
CLGENMOD 00072 storage initialization routine is to be invoked.
CLGENMOD 00073 This routine frees any storage remaining from a
CLGENMOD 00074 previous program. STR is the default setting if
CLGENMOD 00075 the MODULE is to be loaded at the beginning of
CLGENMOD 00076 user available storage.

CLGENMOD 00078 **NOSTR** indicates that, when the MODULE is loaded, free
CLGENMOD 00079 storage does not reset any storage currently in
CLGENMOD 00080 use. NOSTR is the default setting if the MODULE
CLGENMOD 00081 is to be loaded at a location other than the
CLGENMOD 00082 default load address.

CLGENMOD 00084 **SYSTEM** indicates that when the MODULE is subsequently
CLGENMOD 00085 loaded, it is to have a storage protect key of
CLGENMOD 00086 zero.

CLGENMOD 00089 Note: Before the file is written, undefined symbols are set
CLGENMOD 00090 to location zero and the common reference control section is
CLGENMOD 00091 initialized. The undefined symbols are not retained in the
CLGENMOD 00092 MODULE file as being unresolved; therefore, once the MODULE
CLGENMOD 00093 is generated, those references cannot be resolved and may
CLGLOBAL 00001 cause unpredictable results during execution.

CLGLOBAL 00012 GLOBAL

CLGLOBAL 00014 The GLOBAL command is used to specify CMS libraries to be
CLGLOBAL 00015 searched when processing subsequent CMS commands. The
CLGLOBAL 00016 GLOBAL command remains in effect for an entire CMS session
CLGLOBAL 00016 unless it is explicitly cancelled or until another GLOBAL
CLGLOBAL 00017 command is entered. There are no default libraries, so the
CLGLOBAL 00018 command must be entered if any libraries are to be used.
CLGLOBAL 00018 The GLOBAL command will verify the existence of the
CLGLOBAL 00019 libraries and issue an error message if a specified library
CLGLOBAL 00020 does not exist.

CLGLOBAL 00022
CLGLOBAL 00023 | GLOBAL | [MACLIB] [libname...] |
CLGLOBAL 00024 | GL | [TXTLIB] |
CLGLOBAL 00025

CLGLOBAL 00032 MACLIB the MACLIB form of the GLOBAL command allows
CLGLOBAL 00033 the user to specify the macro libraries that
CLGLOBAL 00033 are to be used during the execution of
CLGLOBAL 00035 language processor commands.

CLGLOBAL 00038 TXTLIB The TXTLIB form of the GLOBAL command allows
CLGLOBAL 00039 the user to specify text libraries to be
CLGLOBAL 00040 searched for missing subroutines when the
CLGLOBAL 00040 LOAD or INCLUDE command is issued or when a
CLGLOBAL 00040 dynamic load occurs (that is, OS SVC 8 is
CLGLOBAL 00041 issued).

CLGLOBAL 00044 libname specifies the filenames of up to eight
CLGLOBAL 00045 libraries. If the MACLIB form of the GLOBAL
CLGLOBAL 00046 command is being used, the filetypes of all
CLGLOBAL 00046 files specified must be MACLIB. If the
CLGLOBAL 00047 TXTLIB form of the command is being used, the
CLGLOBAL 00047 filetypes of all files specified must be
CLGLOBAL 00048 TXTLIB. The libraries are searched in the
CLGLOBAL 00048 order in which they are named. If no library
CLGLOBAL 00049 names are specified, the command is used to
CLGLOBAL 00049 cancel the effect of any previous GLOBAL
CLGLOBAL 00049 command of the specified form (that is,
CLGLOBAL 00050 MACLIB or TXTLIB).

CLGLOBAL 00053 Example 1

CLGLOBAL 00055
CLGLOBAL 00056 | GLOBAL | MACLIB ACCESS SYSMAC |
CLGLOBAL 00057

CLGLOBAL 00061 Action: The system searches files ACCESS MACLIB and SYSMAC
CLGLOBAL 00062 MACLIB for missing macros during compilations.

CLGLOBAL 00065 Example 2

CLGLOBAL 00067
CLGLOBAL 00068 | GLOBAL | TXTLIB CONVERT FLOAT |
CLGLOBAL 00069

CLGLOBAL 00072 Action: The system searches files CONVERT TXTLIB and FLOAT
CLGLOBAL 00073 TXTLIB for missing subroutines during subsequent LOAD and
CLGLOBAL 00074 INCLUDE commands.

CLGLOBAL 00076 Example 3

CLGLOBAL 00078
CLGLOBAL 00079 | GLOBAL | MACLIB |
CLGLOBAL 00080

CLGLOBAL 00084 Action: Cancels the effect of any previous GLOBAL MACLIB
CLINCLUD 00002 libname command.

CLINCLUD 00006 **INCLUDE**

CLINCLUD 00008 The INCLUDE command reads from disk one or more TEXT files
CLINCLUD 00009 containing relocatable object code and loads them into
CLINCLUD 00010 storage, establishing the proper linkages between the files.
CLINCLUD 00010 INCLUDE is normally used to resolve references left
CLINCLUD 00011 unresolved by a previous LOAD or INCLUDE command. A LOAD
CLINCLUD 00011 command must have been previously issued for the INCLUDE
CLINCLUD 00013 command to produce desirable results.

```
┌──────────────────────────────────────────────────────────────┐
│ INCLUDE│  fn...  [(options...)]                              │
│ IN     │                                                     │
│        │         options: ┌         ┐                        │
│        │                  │ CLEAR   │                        │
│        │                  │ NOCLEAR │                        │
│        │                  └         ┘                        │
│        │                  ┌                ┐                 │
│        │                  │        {entry} │                 │
│        │                  │ RESET  {     } │                 │
│        │                  │        {  *  } │                 │
│        │                  └                ┘                 │
│        │                  [ ORIGIN hexloc  ]                 │
│        │                  ┌         ┐                        │
│        │                  │ MAP     │                        │
│        │                  │ NOMAP   │                        │
│        │                  └         ┘                        │
│        │                  ┌         ┐                        │
│        │                  │ TYPE    │                        │
│        │                  │ NOTYPE  │                        │
│        │                  └         ┘                        │
│        │                  ┌         ┐                        │
│        │                  │ INV     │                        │
│        │                  │ NOINV   │                        │
│        │                  └         ┘                        │
│        │                  ┌         ┐                        │
│        │                  │ REP     │                        │
│        │                  │ NOREP   │                        │
│        │                  └         ┘                        │
│        │                  ┌         ┐                        │
│        │                  │ AUTO    │                        │
│        │                  │ NOAUTO  │                        │
│        │                  └         ┘                        │
│        │                  ┌         ┐                        │
│        │                  │ LIBE    │                        │
│        │                  │ NOLIBE  │                        │
│        │                  └         ┘                        │
│        │                  [ START    ]                       │
│        │                                                     │
│        │                  [ SAME     ]                       │
└──────────────────────────────────────────────────────────────┘
```

fn specifies the names of the files to be loaded into storage. Files must have a filetype of TEXT and consist of relocatable object code such as that produced by the language processor commands.

<u>Options</u>

If options have been specified with a previous LOAD or

INCLUDE command, these options (with the exception of CLEAR and ORIGIN) remain set if SAME is specified when INCLUDE is issued. Otherwise, the options assume their default settings. If conflicting options are specified, the last one entered will be in effect.

CLEAR — specifies that the load area in storage is to be cleared to binary zeros before the files are loaded.

NOCLEAR — does not clear the load area before loading.

RESET(entry) { * } — resets the current execution starting point that had been set by a previous LOAD or INCLUDE command. If 'entry' is specified, the starting execution address is reset to the specified location. If an asterisk is specified, the starting point is reset to the location of the first file or to the address specified as an operand of an END card, LDT card, or ENTRY card.

ORIGIN hexloc — specifies that the loading of the program is to begin at hexadecimal location "hexloc". If this option is not specified, loading will begin at the next available storage location. INCLUDE does not overlay any previously loaded files unless this option is specified and the address given indicates a location within a previously loaded object module.

MAP — specifies that information is to be added to the LOAD MAP file.

NOMAP — does not add any information to the LOAD MAP file.

TYPE — specifies that the LOAD MAP of the files loaded is to be typed at the terminal, in addition to being written onto the user's primary disk. This option is valid only if MAP is specified or implied.

NOTYPE — does not type the LOAD MAP file at the terminal.

INV — prints invalid card images in the LOAD MAP.

NOINV — suppresses the inclusion of invalid card images in the LOAD MAP.

REP — prints Replace card images in the LOAD MAP. See the explanation of the LOAD command for a

CLINCLUD 00135 description of the REP card.

CLINCLUD 00137 NOREP suppresses the printing of Replace card images
CLINCLUD 00138 in the LOAD MAP.

CLINCLUD 00140 AUTO specifies that user disks are to be searched
CLINCLUD 00140 for TEXT files to resolve undefined
CLINCLUD 00141 references.

CLINCLUD 00144 NOAUTO suppresses automatic searching for TEXT files.

CLINCLUD 00146 LIBE specifies that TXTLIB files which have been
CLINCLUD 00147 activated by the GLOBAL command are to be
CLINCLUD 00148 searched for missing subroutines.

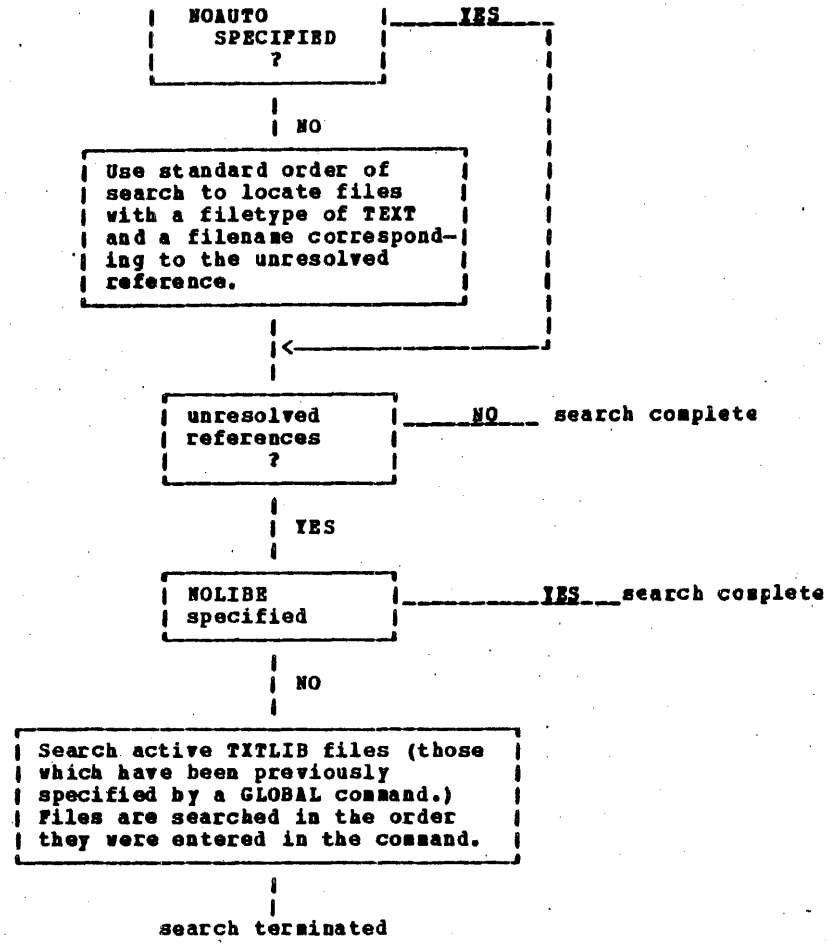
CLINCLUD 00150 NOLIBE does not search any TXTLIB files for
CLINCLUD 00151 unresolved references.

CLINCLUD 00153 START specifies that execution is to begin after
CLINCLUD 00154 loading has completed.

CLINCLUD 00156 SAME indicates that the same options (except ORIGIN
CLINCLUD 00157 and CLEAR) that were used by a previous
CLINCLUD 00157 INCLUDE or LOAD command are to remain in
CLINCLUD 00157 effect. Otherwise, the default setting of
CLINCLUD 00159 unspecified options is assumed. If other
CLINCLUD 00159 options are specified with SAME, they will
CLINCLUD 00160 override previously specified options. See
CLINCLUD 00161 Example 2.

CLINCLUD 00166 **Note: Resolution of References**

CLINCLUD 00176
CLINCLUD 00177 ┌──────────────────────────────────┐
CLINCLUD 00178 │ Use standard order of search to │
CLINCLUD 00179 │ locate the TEXT files specified by │
CLINCLUD 00180 │ fn ...   │
CLINCLUD 00181 └──────────────────────────────────┘
CLINCLUD 00182 │
CLINCLUD 00183 │ NOTE:
CLINCLUD 00184 │ IN FINAL PUBLICATION THIS
CLINCLUD 00185 ┌──────────────┐ WILL BE REDRAWN AS REQUIR
CLINCLUD 00186 │ │
CLINCLUD 00187 │ unresolved │
CLINCLUD 00188 │ references │──────NO──────search complete
CLINCLUD 00189 │ ? │
CLINCLUD 00190 └──────────────┘
CLINCLUD 00191 │
CLINCLUD 00192 │ yes
CLINCLUD 00193 │
CLINCLUD 00194 ┌──────────────┐



Example 1

```
INCLUDE | MAIN SUBI DATA (RESET MAIN MAP START)
```

Action: The files named MAIN TEXT, SUBI TEXT, and DATA TEXT are brought into main storage and appended to files which were previously loaded. Information about loaded files is added to the LOAD MAP file. Execution will begin at Entry

point MAIN.

Example 2

```
┌──────────────────────────────────────────────────┐
| LOAD | MAIN  (NOMAP NOLIBE NOREP)                |
└──────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────┐
| INCLUDE | SUBI (MAP SAME)                        |
└──────────────────────────────────────────────────┘
```

Action: During execution of the LOAD command, the file named MAIN TEXT is brought into main storage. The following options are in effect: NOMAP, NOLIBE, NOREP, NOTYPE, INV, AUTO. During execution of the INCLUDE command, the file named SUBI TEXT is appended to MAIN TEXT. The following options are in effect:

MAP, NOLIBE, NOREP, NOTYPE, INV, AUTO

Note: After IPLing CMS, at least one LOAD command must have been issued before INCLUDE can be used with predictable results.

Responses

EXECUTION BEGINS...

START has been specified with INCLUDE and the loaded program has begun execution. Any further responses are from the program.

INVALID CARD - xxx...xxx

INV has been specified with LOAD and an invalid card has been found. The message and the contents of the invalid card (xxx...xxx) are only listed in the file LOAD MAP. The invalid card is ignored and loading continues.

CONTROL CARD - ...

A loader or library-search control card has been encountered, (that is, ENTRY or LIBRARY). See the description of the LOAD command for the use of ENTRY and LIBRARY control cards. This response is placed in the LOAD MAP only.

CLLISTF 00004 **LISTFILE**

CLLISTF 00012 The LISTFILE command is used to obtain specified information
CLLISTF 00012 about the user's CMS files residing on accessible, user
CLLISTF 00013 disks. The information may either be typed at the terminal
CLLISTF 00014 or used to create a special EXEC file on disk. All operands
CLLISTF 00014 are optional; if no operands are specified, a list of
CLLISTF 00015 default information about each file on the user's disks is
CLLISTF 00016 typed at the terminal.

CLLISTF 00026
CLLISTF 00027 | LISTFILE | [[fn [ft [fm]]] [(options)]]
CLLISTF 00028 | L |
CLLISTF 00029 | | options: [HEADER|NOHEADER]
CLLISTF 00030 | | [EXEC|APPEND]
CLLISTF 00031 | | [FNAME|FTYPE|FMODE|FORMAT|ALLOC|DATE|LABEL]
CLLISTF 00032

CLLISTF 00037 fn specifies the filename of the files for which
CLLISTF 00038 information is to be collected. If an asterisk is
CLLISTF 00040 coded in this field, all filenames will be used.

CLLISTF 00042 ft specifies the filetype of the files for which
CLLISTF 00043 information is to be collected. If an asterisk is
CLLISTF 00045 coded in this field, all filetypes will be used.

CLLISTF 00047 fm specifies the filemode of the files for which
CLLISTF 00048 information is to be collected. If this field is
CLLISTF 00049 omitted, the primary disk will be searched. If an
CLLISTF 00050 asterisk is coded all disks will be searched.

CLLISTF 00052 Note: An asterisk (*), immediately preceded by any number of
CLLISTF 00053 characters for fn or ft, searches for the specified
CLLISTF 00054 characters as the leading characters for that identifier.
CLLISTF 00054 For example, LISTFILE ABC* ASSEMBLE prints the identifiers
CLLISTF 00056 for all ASSEMBLE files with filenames beginning with ABC.

CLLISTF 00059 Output Format Options

CLLISTF 00061 HEADER specifies that column headings are to be included
CLLISTF 00064 H in the listing. HEADER is the default. The format
CLLISTF 00065 of the heading is:

CLLISTF 00067 FNAME FTYPE FMODE FORMAT RECS BLKS DATE TIME LABEL

CLLISTF 00070 NOHEADER specifies that column headings are not to be
CLLISTF 00072 NOH included in the list.

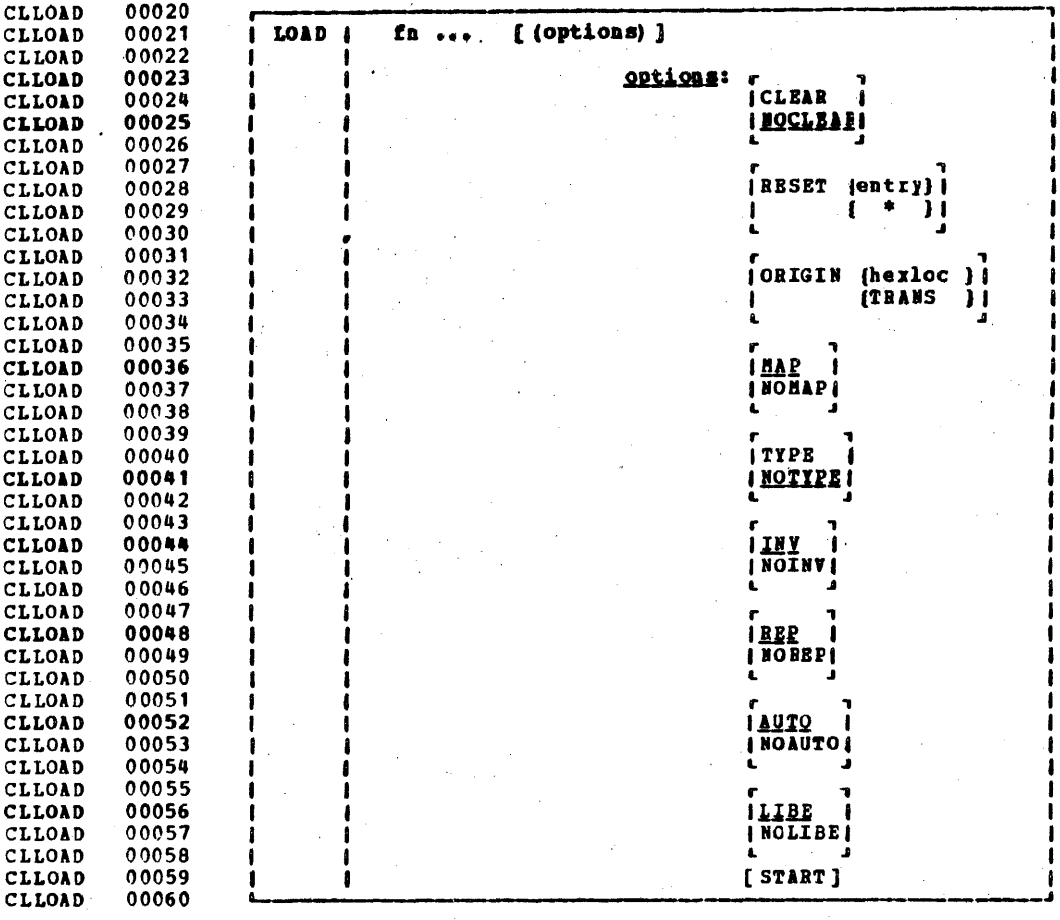
```
CLLISTF  00076    Output Disposition Options

CLLISTF  00078    EXEC       indicates that a file of 80-character records (one
CLLISTF  00082    E          record for each of the files which satisfies the
CLLISTF  00082               given fileid) is to be created on the user's
CLLISTF  00083               primary disk. The file name is CMS EXEC A1. If a
CLLISTF  00084               file with this name already exists, the existing
CLLISTF  00085               one is erased and a new one is created (unless the
CLLISTF  00086               APPEND option is specified, in which case the
CLLISTF  00087               existing file is retained and the new entries are
CLLISTF  00088               appended to it). The EXEC procedure thus created
CLLISTF  00089               contains two symbolic variables, &1 and &2. The
CLLISTF  00090               main purpose of the CMS EXEC file created via this
CLLISTF  00091               option is to be used with the EXEC command, but it
CLLISTF  00092               can be processed as any other file (that is,
CLLISTF  00092               printed, typed, edited, added to, changed, and so
CLLISTF  00093               forth). "Appendix E: Sample Terminal Session"
CLLISTF  00093               shows an example of the use of a CMS EXEC file
CLLISTF  00095               created in this manner.

CLLISTF  00098    APPEND     The EXEC list created is to be appended to the
CLLISTF  00101    AP         existing CMS EXEC A1 file. If the EXEC option is
CLLISTF  00102               specified instead of this one, any existing CMS
CLLISTF  00103               EXEC file is erased and replaced by the one
CLLISTF  00104               created by this LISTFILE command. If this option
CLLISTF  00105               is specified and no CMS EXEC file exists, one will
CLLISTF  00106               be created.

CLLISTF  00110    Information Request Options

CLLISTF  00112    Note: Only one of these options need be specified. If one
CLLISTF  00113    is specified, any options with a higher priority will also
CLLISTF  00114    be in effect. If none of the following options are
CLLISTF  00115    specified, the default information request options noted
CLLISTF  00116    below are in effect.

CLLISTF  00121    Default Information Request Options

CLLISTF  00124    FNAME      specifies that the list is to contain only
CLLISTF  00127    FN         filenames. Option priority is 7.

CLLISTF  00130    FTYPE      specifies that the list created is to contain only
CLLISTF  00133    FT         filenames and filetypes. Option priority is 6.

CLLISTF  00136    FMODE      specifies that the list will contain filenames,
CLLISTF  00139    FM         filetypes, and filemodes. Option priority is 5.

CLLISTF  00142    Supplemental Information Options

CLLISTF  00144    FORMAT     specifies that the list is to contain the record
CLLISTF  00146    FO         format and logical record length of the largest
CLLISTF  00148               record of each file. Option priority is 4.

CLLISTF  00151    ALLOC      specifies that the list is to include the amount
```

```
CLLISTF  00153    AL        of disk space that CMS has allocated to the
CLLISTF  00154              specified file. The quantities given are the
CLLISTF  00155              number of 800-byte blocks and the number of
CLLISTF  00157              logical records in the file. Option pricrity is 3.

CLLISTF  00160    DATE      specifies that the date the file was last written
CLLISTF  00163    D         is to be included in the list. The form of the
CLLISTF  00164              date is:

CLLISTF  00168                   month/day/year  hour:minute

CLLISTF  00172              Option priority is 2.

CLLISTF  00175    LABEL     specifies that the label of the disk on which the
CLLISTF  00178    L         file resides is to be included in the list. Option
CLLISTF  00179              priority is 1.


CLLISTF  00182    Example

CLLISTF  00189    ┌──────────────────────────────────────────┐
CLLISTF  00190    | LISTFILE  *   ASSEMBLE *   (LABEL)       |
CLLISTF  00191    |                                          |
CLLISTF  00192    └──────────────────────────────────────────┘

CLLISTF  00197    Action: LABEL is the lowest priority option; therefore, all
CLLISTF  00199    other options are also in effect. This command requests that
CLLISTF  00201    all information about the files with a filetype of ASSEMBLE
CLLISTF  00201    is to be typed at the terminal. The header will be typed
CLLISTF  00203    since LABEL is a supplemental information option.

CLLISTF  00205    The following will type at the terminal:

CLLISTF  00207    FNAME   FTYPE   FMODE   FORMAT   RECS   BLKS   DATE    TIME    LABEL
CLLISTF  00208                            ┌ ┐
CLLISTF  00209                            |F|
CLLISTF  00210     fn   ASSEMBLE   fm     |V| lrecl norecs noblks mm/dd/yy hh:mm  volid
CLLISTF  00212     •                      └ ┘                                      •
CLLISTF  00213     •                                                               •
CLLISTF  00214     •                                                               •

CLLISTF  00217    One entry will type for each file with a filetype of
CLLISTF  00218    ASSEMBLE.
CLLISTF  00220    where:

CLLISTF  00224    fn        is the filename of the file.

CLLISTF  00227    ASSEMBLE  is the filetype specified in the command.

CLLISTF  00230    fm        is the filemode of the file
```

```
CLLISTF  00234    ┌─┐
CLLISTF  00236    |F|
CLLISTF  00238    |V|      is the file format: F - (fixed length)
CLLISTF  00240    └─┘                          V - (variable length)

CLLISTF  00242    lrecl    is the logical record length of the largest record
CLLISTF  00243             in the file.

CLLISTF  00246    norecs   is the number of logical records in the file.

CLLISTF  00248    noblks   is the number of physical blocks that the file
CLLISTF  00249             occupies on disk.

CLLISTF  00251    mm/dd/yy is the date (month/day/year) that the file was
CLLISTF  00252             created.

CLLISTF  00254    hh:mm    is the time (hours:minutes) that the file was
CLLISTF  00255             created.

CLLISTF  00257    volid    is the volume serial number of the minidisk on
CLLISTF  00258             which the file resides.


CLLISTF  00263    Responses


CLLISTF  00266    If the EXEC option is not specified, the requested
CLLOAD   00001    information is typed at the terminal.
```

CLLOAD 00004 **LOAD**

CLLOAD 00005 The LOAD command reads from disk one or more TEXT files
CLLOAD 00006 containing relocatable object code and loads them into main
CLLOAD 00008 storage, establishing the proper linkages between the files.



CLLOAD 00067 **fn** specifies the names of the files to be loaded into
CLLOAD 00068 storage. Files must have a filetype of TEXT and
CLLOAD 00069 consist of relocatable object code such as that
CLLOAD 00070 produced by the language processors.

CLLOAD 00075 **Options**

CLLOAD 00076 If conflicting options are specified, the last one entered
CLLOAD 00077 will be in effect.

CLLOAD 00079 **CLEAR** specifies that the load area in storage is to be
CLLOAD 00080 cleared to binary zeroes before the object files are
CLLOAD 00081 loaded.

CLLOAD 00083 **NOCLEAR** specifies that the load area is not to be cleared
CLLOAD 00084 before loading.

CLLOAD 00087 **RESET** entry
CLLOAD 00089 *

CLLOAD 00090 sets the starting location for the programs
CLLOAD 00091 currently loaded. The parameter 'entry' must be an
CLLOAD 00092 external name (that is, CSECT, ENTRY, etc.) in the
CLLOAD 00093 loaded programs. If RESET is not specified, the
CLLOAD 00093 default entry point is either (1) the entry point of
CLLOAD 00094 the first file loaded, (2) the address on an END
CLLOAD 00095 card, (3) the location of a name occurring on an LDT
CLLOAD 00096 card, or (4) the location of the operand on an ENTRY
CLLOAD 00097 card. If * is entered the results are the same as if
CLLOAD 00098 the RESET option were omitted.

CLLOAD 00101 **ORIGIN** hexloc

CLLOAD 00102 specifies that loading of the program is to begin at
CLLOAD 00103 hexadecimal location "hexloc". If TRANS is
CLLOAD 00104 specified, the file is loaded into the CMS nucleus
CLLOAD 00104 transient storage locations. If this option is not
CLLOAD 00105 specified, loading commences at the first available
CLLOAD 00106 user location.

CLLOAD 00108 **MAP** specifies that the file LOAD MAP is to be created on
CLLOAD 00109 the primary disk.

CLLOAD 00111 **NOMAP** specifies that the file LOAD MAP is not to be
CLLOAD 00112 created.

CLLOAD 00115 **TYPE** types the LOAD MAP at the terminal. This option is
CLLOAD 00116 valid only if the MAP option is in effect.

CLLOAD 00119 **NOTYPE** does not type the LOAD MAP at the terminal.

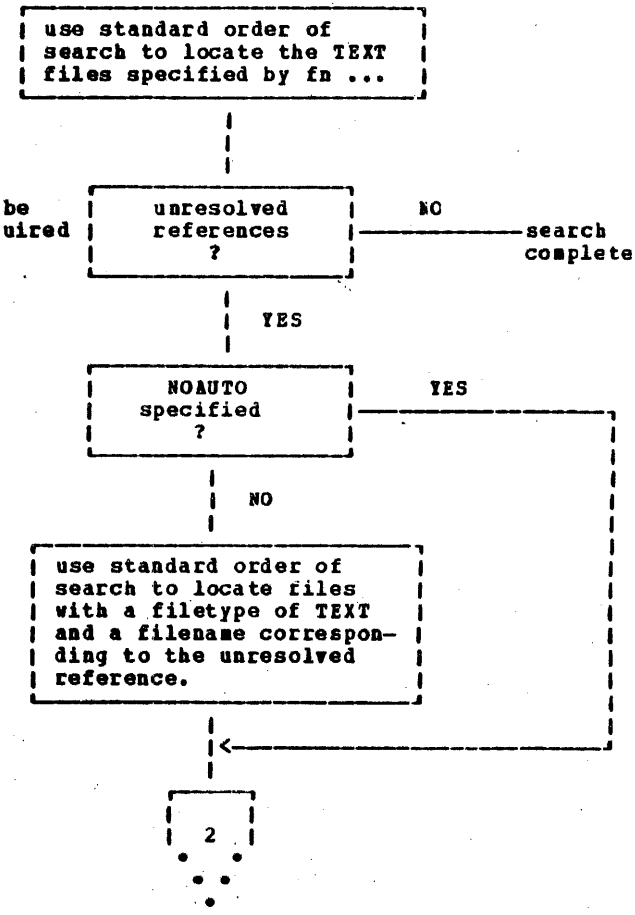
CLLOAD 00121 **INV** specifies that invalid card images are to be
CLLOAD 00122 included in the LOAD MAP file.

CLLOAD 00124 **NOINV** specifies that invalid card images are not to be
CLLOAD 00125 included in the LOAD MAP.

CLLOAD 00127 **REP** specifies that Replace card images are to be

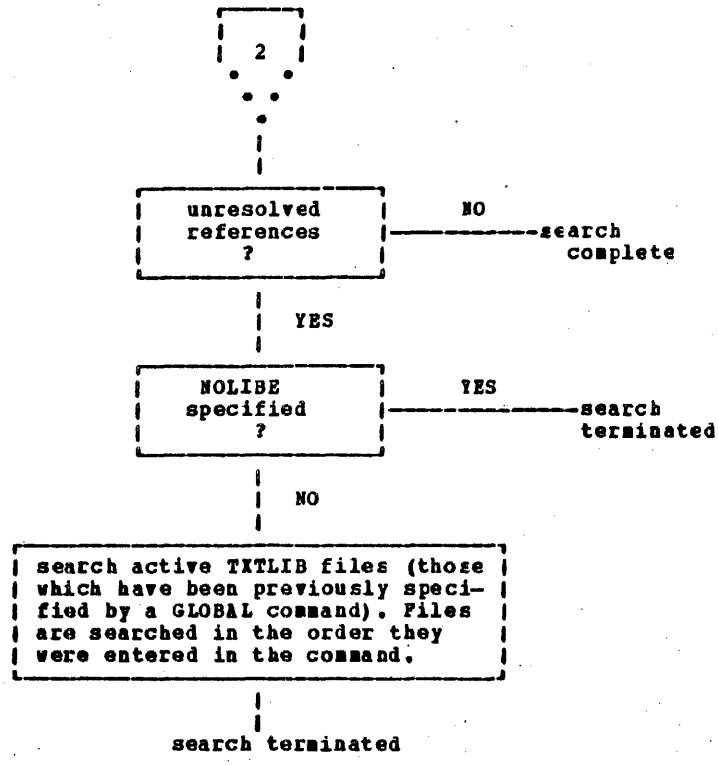
```
CLLOAD  00128           included in the LOAD MAP file.

CLLOAD  00130   NOREP   specifies that Replace card images are not to be
CLLOAD  00131           included in the LOAD MAP file.

CLLOAD  00133   AUTO    specifies that user disks are to be searched for
CLLOAD  00135           TEXT files to resolve undefined references.

CLLOAD  00138   NOAUTO  suppresses automatic searching for TEXT files.

CLLOAD  00140   LIBE    specifies that TXTLIB files are to be searched for
CLLOAD  00141           missing subroutines. If TXTLIB files are to be
CLLOAD  00141           searched, they must previously have been made active
CLLOAD  00142           by a GLOBAL command.

CLLOAD  00144   NOLIBE  specifies that TXTLIB files are not to be searched
CLLOAD  00145           for unresolved references.

CLLOAD  00147   START   specifies that the program being loaded is to be
CLLOAD  00148           executed when loading has completed. LOAD does not
CLLOAD  00148           normally begin execution of the loaded files. To
CLLOAD  00149           begin execution immediately upon successful
CLLOAD  00150           completion of loading, START can be specified. LOAD
CLLOAD  00151           then transfers control to an entry point in the
CLLOAD  00151           program. The default entry point is either the
CLLOAD  00152           address specified in the operand field of the first
CLLOAD  00153           END card containing a non-blank operand field, the
CLLOAD  00154           address of a name on an LDT card, the beginning of
CLLOAD  00155           the first file loaded if all END card images in the
CLLOAD  00155           TEXT files contain blank operand fields, or the
CLLOAD  00156           ENTRY specified by the user.
```

CLLOAD 00160 **Note: Resolution of Unresolved References**



CLLOAD 00167
CLLOAD 00168
CLLOAD 00169
CLLOAD 00170
CLLOAD 00171
CLLOAD 00172
CLLOAD 00173
CLLOAD 00174 **NOTE:**
CLLOAD 00175 In final PUB
CLLOAD 00176 this FIG will be
CLLOAD 00177 redrawn as required
CLLOAD 00178
CLLOAD 00179
CLLOAD 00180
CLLOAD 00181
CLLOAD 00182
CLLOAD 00183
CLLOAD 00184
CLLOAD 00185
CLLOAD 00186
CLLOAD 00187
CLLOAD 00188
CLLOAD 00189
CLLOAD 00190
CLLOAD 00191
CLLOAD 00192
CLLOAD 00193
CLLOAD 00194
CLLOAD 00195
CLLOAD 00196
CLLOAD 00197
CLLOAD 00198
CLLOAD 00199
CLLOAD 00200
CLLOAD 00201
CLLOAD 00202
CLLOAD 00203
CLLOAD 00204
CLLOAD 00206
CLLOAD 00207
CLLOAD 00208

```
CLLOAD   00210
CLLOAD   00211
CLLOAD   00212
CLLOAD   00213
CLLOAD   00214
CLLOAD   00215
CLLOAD   00216
CLLOAD   00217
CLLOAD   00218
CLLOAD   00219
CLLOAD   00220
CLLOAD   00221
CLLOAD   00222
CLLOAD   00223
CLLOAD   00224
CLLOAD   00225
CLLOAD   00226
CLLOAD   00227
CLLOAD   00228
CLLOAD   00229
CLLOAD   00230
CLLOAD   00231
CLLOAD   00232
CLLOAD   00233
CLLOAD   00234
CLLOAD   00235
CLLOAD   00236
CLLOAD   00237
CLLOAD   00238
CLLOAD   00239
CLLOAD   00240
CLLOAD   00241
CLLOAD   00242
CLLOAD   00243
```



CLLOAD 00248 LOAD MAP FILE: Unless the NOMAP option is specified, a load
CLLOAD 00249 map is created on the primary disk each time the LOAD
CLLOAD 00249 command is issued. A load map is a file that contains the
CLLOAD 00250 location of control sections and entry points of files
CLLOAD 00251 loaded into main storage. It may also contain certain
CLLOAD 00252 messages and card images of any invalid cards or replace
CLLOAD 00252 cards that exist in the loaded files. This load map is
CLLOAD 00253 normally created as a file with the identifier LOAD MAP.
CLLOAD 00253 Only one such file may exist on the primary disk. Each time
CLLOAD 00254 LOAD is issued, a new LOAD MAP replaces any previous LOAD
CLLOAD 00255 MAP file.

CLLOAD 00256 If invalid card images exist in the file or files that are
CLLOAD 00257 being loaded, they are listed with the message INVALID CARD
CLLOAD 00257 in the LOAD MAP file. To suppress this listing in the load
CLLOAD 00259 map, the NOINV option must be specified.

CLLOAD 00260 If Replace (REP) card images exist in the file being loaded,
CLLOAD 00261 they are included in LOAD MAP. To suppress this listing of
CLLOAD 00262 REP card images, the NOREP option must be specified.

CLLOAD 00264 DUPLICATE CSECTS: Duplicate CSECTS (control sections) are
CLLOAD 00265 bypassed by the loader. Only the first CSECT encountered is
CLLOAD 00265 physically loaded. The duplicates are not loaded and a
CLLOAD 00267 warning message is included in the LOAD MAP.

CLLOAD 00269 LOADER CONTROL CARDS: Five types of cards can be added to a
CLLOAD 00270 TEXT file. These are the Set Location Counter (SLC), the
CLLOAD 00271 Include Control Section (ICS), the Replace (REP), the ENTRY,
CLLOAD 00272 and the LIBRARY cards. These are used to set the main
CLLOAD 00273 storage location where LOAD begins placing the file, to make
CLLOAD 00274 corrections and additions to the relocatable object code in
CLLOAD 00274 main storage once the file is loaded, to specify entry
CLLOAD 00275 points, and to specify references that are not to be
CLLOAD 00276 resolved. These cards can be added to the TEXT files which
CLLOAD 00277 have been punched and can then be read back in, or they can
CLLOAD 00279 be added using the EDIT command.

CLLOAD 00281 Set Location Counter (SLC) Card: The SLC card sets the
CLLOAD 00282 location counter used with the loader. The file loaded in
CLLOAD 00282 after the SLC card is placed in main storage beginning at
CLLOAD 00283 the address set by this SLC card. The SLC card has the
CLLOAD 00284 format shown in Figure 18. It sets the location counter in
CLLOAD 00285 one of three ways:

CLLOAD 00288 1. With the absolute virtual address specified as a
CLLOAD 00290 hexadecimal number in columns 7-12.

CLLOAD 00292 2. With the symbolic address already defined as a program
CLLOAD 00293 name or entry point. This is specified by a symbolic
CLLOAD 00294 name punched in columns 17-22.

CLLOAD 00296 3. If both a hexadecimal address and a symbolic name are
CLLOAD 00297 specified, the absolute virtual address is converted to

CLLOAD 00298 binary and added to the address assigned to the
CLLOAD 00298 symbolic name; the resulting sum is the address to
CLLOAD 00299 which the loader's location counter is set. For
CLLOAD 00300 example, if 0000F8 was specified in columns 7-12 of the
CLLOAD 00301 SLC card image and GAMMA was specified in columns
CLLOAD 00301 17-21, where GAMMA has an assigned address of 006100
CLLOAD 00302 (hexadecimal), the absolute address in columns 7-12 is
CLLOAD 00303 added to the address assigned to GAMMA giving a total
CLLOAD 00303 of 0061F8. Thus, the location counter would be set to
CLLOAD 00304 0061F8.

CLLOAD 00305 If there are blanks in both columns 7-12 and 17-22, or
CLLOAD 00306 the symbolic name has not yet been defined, the
CLLOAD 00307 response INVALID CARD xxx...xxx is typed or, depending
CLLOAD 00308 on whether the option NOINV or INV was specified, is
CLLOAD 00308 written in the LOAD MAP file. If only the symbolic
CLLOAD 00309 address is to be used, columns 7-12 must be left blank
CLLOAD 00310 or be all zeros. If only the absolute address is to be
CLLOAD 00311 used, columns 17-22 must be left blank.

COLUMN	CONTENTS
1	Load card identification (12-2-9). Identifies this as a card acceptable to the loader.
2-4	SLC Identifies the type of load card
5-6	Blank
7-12	Hexadecimal address to be added to the value of the symbol, if any, in columns 17-22. Must be right-justified in these columns, with unused leading columns filled in with zeros.
13-16	Blank
17-22	Symbolic name whose assigned location is used by the loader. Must be left-justified in these columns. If blank, the address in the absolute field is used.
23	BLANK
24-72	May be used for comments or left blank.
73-80	Not used by the loader. The user may leave these blank or insert program identification for his own convenience.

Figure 18. SLC Card Format

CLLOAD 00357 Loader Terminate (LDT) Card: The LDT card is used in a
CLLOAD 00358 TXTLIB as the last record of a member. It indicates to the
CLLOAD 00359 loader that all records for that member have been processed.
CLLOAD 00359 The LDT card may contain a name which will be used as the
CLLOAD 00361 entry point for the loaded member.

CLLOAD 00368
CLLOAD 00369
CLLOAD 00370
CLLOAD 00371
CLLOAD 00372
CLLOAD 00373
CLLOAD 00374
CLLOAD 00375
CLLOAD 00376
CLLOAD 00377
CLLOAD 00378
CLLOAD 00379
CLLOAD 00380
CLLOAD 00381

COLUMN	CONTENTS
1	12-2-9 Punch
2-4	LDT identifies type of card
5-16	Not used.
17-24	Blank or entry name (left justified padded with blanks to 8 characters)
25-80	Not used.

CLLOAD 00383 Figure 19. LDT Card Format

CLLOAD 00388 Include Control Section (ICS) Card: The ICS card changes
CLLOAD 00389 the length of a specified control section or defines a new
CLLOAD 00390 control section. It should be used only when REPLACE cards
CLLOAD 00391 cause a control section to be increased in length. The
CLLOAD 00392 format of an ICS card is shown in Figure 20. An ICS card
CLLOAD 00394 must be placed at the front of the card deck or TEXT file.

COLUMN	CONTENTS
1	Load card identification (12-2-9). Identifies this as a card acceptable to the loader.
2-4	ICS. Identifies the type of load card.
5-16	Blank
17-22	Control section name--left-justified in these columns.
23	Blank
24	, (comma)
25-28	Hexadecimal length in bytes of the control section. This must not be less than the actual length of the previously specified control section. Must be right-justified in these columns with unused leading columns filled with zeros.
29	Blank
30-72	May be used for comments or left blank.
73-80	Not used by the loader. The user may leave these blank or insert program identification for his own convenience.

CLLOAD 00429 Figure 20. ICS Card Format

CLLOAD 00434 Replace (REP) Card: A REP card allows instructions and
CLLOAD 00435 constants to be changed and additions made. The REP card
CLLOAD 00436 must be punched in hexadecimal code. The format of a REP
CLLOAD 00437 card is shown in Figure 21. The data in columns 17-70
CLLOAD 00438 (excluding the commas) replaces what has already been loaded
CLLOAD 00439 into main storage, beginning at the address specified in
CLLOAD 00440 columns 7-12. REP cards are placed in the card deck either
CLLOAD 00441 (1) immediately preceding the last card (END card) if the
CLLOAD 00442 text deck does not contain relocatable data such as address
CLLOAD 00443 constants, or (2) immediately preceding the first RLD
CLLOAD 00444 (relocatable dictionary) card if there is relocatable data
CLLOAD 00445 in the text deck. If additions made by REP cards increase
CLLOAD 00446 the length of a control section, an ICS card, which defines
CLLOAD 00447 the total length of the control section, must be placed at
CLLOAD 00448 the front of the deck.

CLLOAD 00450
CLLOAD 00451
CLLOAD 00452
CLLOAD 00453
CLLOAD 00454
CLLOAD 00455
CLLOAD 00456
CLLOAD 00457
CLLOAD 00458
CLLOAD 00459
CLLOAD 00460
CLLOAD 00461
CLLOAD 00462
CLLOAD 00463
CLLOAD 00464
CLLOAD 00465
CLLOAD 00466
CLLOAD 00467
CLLOAD 00468
CLLOAD 00469
CLLOAD 00470
CLLOAD 00471
CLLOAD 00472
CLLOAD 00473
CLLOAD 00474
CLLOAD 00475
CLLOAD 00476
CLLOAD 00477
CLLOAD 00478
CLLOAD 00479
CLLOAD 00480
CLLOAD 00481
CLLOAD 00482
CLLOAD 00483

COLUMN	CONTENTS
1	Load card identification (12-2-9). Identifies this as a card acceptable to the loader.
2-4	REP Identifies the type of load card.
5-6	Blank
7-12	Hexadecimal starting address of the area to be replaced as assigned by the assembler. Must be right-justified in these columns with unused leading columns filled in with zeros.
13-14	Blank
15-16	ESID --External Symbol Identification-- the hexadecimal number assigned to the control section in which replacement is to be made. The LISTING file produced by the compiler indicates this number.
17-70	A maximum of 11 four-digit hexadecimal fields, separated by commas, each replacing one previously loaded halfword (two bytes). The last field must not be followed by a comma.
71-72	Blank
73-80	Not used by the loader. The user may leave these blank or insert program identification for his own convenience.

CLLOAD 00485 Figure 21. REP Card Format

CLLOAD 00491 Entry Card: The ENTRY statement specifies the first
CLLOAD 00492 instruction to be executed. It can be placed before,
CLLOAD 00493 between, or after object modules or other control
CLLOAD 00494 statements. The format of the ENTRY statement is shown in
CLLOAD 00495 Figure 22. The "external name" is the name of a control
CLLOAD 00496 section or an entry name in the input deck. It must be the
CLLOAD 00497 name of an instruction, not of data.

CLLOAD 00504 ┌──────────────────────────────┐
CLLOAD 00505 | ENTRY | external name |
CLLOAD 00506 └──────────────────────────────┘

CLLOAD 00508 Figure 22. ENTRY Card Format

CLLOAD 00513 The loader selects the entry point for the loaded program in
CLLOAD 00514 the following ways:

CLLOAD 00517 1. From the parameter list on the START command.

CLLOAD 00519 2. From the last RESET parameter in the LOAD or INCLUDE
CLLOAD 00520 command entered.

CLLOAD 00523 3. From the last ENTRY statement in the input.

CLLOAD 00526 4. From the last LDT card in the input.

CLLOAD 00529 5. From the first assembler- or compiler-produced END
CLLOAD 00530 statement that specifies an entry point if no ENTRY
CLLOAD 00531 statement is in the input.

CLLOAD 00534 6. From the first byte of the first control section of the
CLLOAD 00535 loaded program if no ENTRY statement and no assembler-
CLLOAD 00536 or compiler-produced END statement specifying an entry
CLLOAD 00538 point. For example:

CLLOAD 00540 ENTRY GO
CLLOAD 00542 Where GO is defined as the external name of the first
CLLOAD 00543 instruction to be executed when the program is loaded.
CLLOAD 00543 The address of the instruction, indicated by the
CLLOAD 00544 symbolic name GO, is specified by the loader as the
CLLOAD 00546 starting point of the program when it is executed.

CLLOAD 00550 Library Card: The LIBRARY statement can be used to specify
CLLOAD 00551 the "never-call" function. The never-call function
CLLOAD 00552 (indicated by an "*" as the first operand) specifies those
CLLOAD 00553 external references that are not to be resolved by the
CLLOAD 00554 automatic library call during any loader step. It is
CLLOAD 00555 negated when a deck containing the external name referred to
CLLOAD 00556 is included as part of the input to the loader. The format
CLLOAD 00558 of the LIBRARY card can be seen in Figure 23. The "external
CLLOAD 00559 reference" refers to an external reference that may be

CLLOAD 00560 unresolved after input processing. It is not to be
CLLOAD 00561 resolved. Multiple external references within the
CLLOAD 00562 parentheses must be separated by commas. The LIBRARY
CLLOAD 00563 statement can be placed before, between, or after object
CLLOAD 00564 decks or other control statements.

CLLOAD 00572 ┌──────────┬──────────────────────────┐
CLLOAD 00573 | LIBRARY | * (external reference) |
CLLOAD 00574 └──────────┴──────────────────────────┘

CLLOAD 00579 Figure 23. LIBRARY Card Format

CLLOAD 00581 Example 1

CLLOAD 00585 ┌──────────┬──────────────────────────┐
CLLOAD 00586 | LIBRARY | * SINE |
CLLOAD 00587 └──────────┴──────────────────────────┘

CLLOAD 00593 Action: * specifies the never-call function. SINE is an
CLLOAD 00594 external reference in the output. As a result, if SINE is
CLLOAD 00595 unresolved after input processing, no automatic library call
CLLOAD 00596 is made.

CLLOAD 00599 Responses

CLLOAD 00601 EXECUTION BEGINS...

CLLOAD 00602 START has been specified with LOAD and the loaded program
CLLOAD 00603 has begun execution. Any further responses are from the
CLLOAD 00604 program.

CLLOAD 00606 INVALID CARD - xxx...xxx

CLLOAD 00607 INV has been specified with LOAD and an invalid card has
CLLOAD 00608 been found. The message and the contents of the invalid
CLLOAD 00609 card (xxx...xxx) are listed in the file LOAD MAP. The
CLLOAD 00610 invalid card is ignored and loading continues.

CLLOAD 00612 CONTROL CARD - ...

CLLOAD 00613 A loader or library-search control card has been encountered
CLLOAD 00614 (that is, ENTRY or LIBRARY). This response is placed in the
CLLOADM 00001 LOAD MAP file.

CLLOADM 00003 **LOADMOD**

CLLOADM 00005 The LOADMOD command brings a disk file into storage. The
CLLOADM 00006 file must be in absolute core-image form as created by the
CLLOADM 00007 GENMOD command.

CLLOADM 00017
CLLOADM 00018 | LOADMOD | fn [ft fm] |
CLLOADM 00019 | LOADM | |
CLLOADM 00020

CLLOADM 00026 fn is the filename of the file to be loaded into
CLLOADM 00027 storage.

CLLOADM 00030 ft is the filetype of the file to be loaded. If
CLLOADM 00032 supplied, it must be MODULE.

CLLOADM 00036 fm is the filemode of the module to be loaded.
CLLOADM 00037 If ft and fm are not supplied, the standard
CLLOADM 00038 order of search is used to locate a file with
CLLOADM 00038 the specified filename and a filetype of
CLLOADM 00039 MODULE.

MACLIB

The MACLIB command is used to create and maintain macro libraries. A macro library consists of macrc definitions and a dictionary which contains the name of the macro, its size, and its location relative to the beginning of the library. The MACLIB command does not check the macrc definitions for errors except for the checking of MACRC and MEND statements.

```
┌──────────────────────────────────────────────────────────────────────────┐
| MACLIB | {(GEN)                                                        } |
| MAC    | {(ADD)   libname    fn1  [fn2...]                              } |
|        | {(REP)                                                        } |
|        | { DEL    libname    macname [macname2...macnameN]              } |
|        | { COMP   libname                                              } |
|        | {                 ┌          ┐                                 } |
|        | {                 |  (TERM)  |                                 } |
|        | { MAP    libname  |  (DISK)  |                                 } |
|        | {                 |  (PRINT) |                                 } |
|        | {                 └          ┘                                 } |
└──────────────────────────────────────────────────────────────────────────┘
```

Functions Performed by the MACLIB Command

GEN	generates a CMS macro library
ADD	adds macros to an existing macro library
REP	replaces existing macros in a macrc library
DEL	deletes members from a macro library
COMP	compacts a macro library
MAP	lists certain information about the macros in a macro library. Available informaticr includes macro name, size, and location relative to the beginning of the library
libname	specifies the filename of a macrc library. If the file already exists, it must have a filetype of MACLIB; if it is being created, it will have a filetype of MACLIB
fn	specifies the names of the macrc definition files to be used. A macro definition file must reside on a CMS disk and its filetype must be either MACRO or COPY. Each file may contain one or more macros and must contain fixed length, 80 character records.

In a MACRO file, the MACRO name is taken from a prototype card within the macro.

If the filetype is COPY and the file contains more than one macro, each macro must be preceded by a control card of the following format:

```
CLMACLIB 00078                  ┌──────────────────┐
CLMACLIB 00079                  | *COPY   macname  |
CLMACLIB 00080                  └──────────────────┘

CLMACLIB 00084                  The name on the control card will be the name of
CLMACLIB 00084                  the macro when it' is placed in the macro
CLMACLIB 00085                  library.  If there is only one macro in the COPY
CLMACLIB 00085                  file and it is not preceded by a COPY control
CLMACLIB 00086                  card, its name (in the macro library) will be
CLMACLIB 00086                  the same as fn.  If there are several macro
CLMACLIB 00087                  definitions in a COPY file and the first one is
CLMACLIB 00087                  not preceded by a COPY control card the entire
CLMACLIB 00088                  file is treated as one macro.

CLMACLIB 00090    macname       specifies the name of a macro which exists in a
CLMACLIB 00091                  macro library.

CLMACLIB 00095    TERM          specifies the device to which the output
CLMACLIB 00097    DISK          generated by the MAP function is to be written.
CLMACLIB 00100    PRINT         Valid device names are DISK (disk file), PRINT
CLMACLIB 00101                  (virtual spooled printer), and TERM (terminal
CLMACLIB 00102                  typewriter).  If no device is specified, DISK is
CLMACLIB 00103                  assumed.  Since this is the only option allowed
CLMACLIB 00104                  in the MACLIB command, only the first word after
CLMACLIB 00105                  the left parenthesis will be examined.  If the
CLMACLIB 00105                  DISK option is specified, the information is
CLMACLIB 00106                  written to a file named 'libname MAP A1'.  If a
CLMACLIB 00107                  file with that name previously existed it is
CLMACLIB 00108                  replaced by the new file.

CLMACLIB 00112    Detailed Descriptions of Functions

CLMACLIB 00114    GEN Function

CLMACLIB 00115    The GEN (generate) function creates a new CMS macro library
CLMACLIB 00117    with a user-defined filename and a filetype of MACLIB.  If a
CLMACLIB 00117    MACLIB file with the same filename already exists, it is
CLMACLIB 00118    erased and replaced by the new MACLIB.  The new file is
CLMACLIB 00119    created from input files specified by fn1 - fnN.  For
CLMACLIB 00120    example:
```

```
CLMACLIB 00128    ┌──────────────────────────────────────┐
CLMACLIB 00129    │ MACLIB GEN OSMAC ACCESS TIME PUT REGEQU  │
CLMACLIB 00130    └──────────────────────────────────────┘

CLMACLIB 00135    Creates a  new file  with the  file identifier  OSMAC MACLIB
CLMACLIB 00137    from macros existing in the files with the file identifiers:

CLMACLIB 00139    ACCESS {MACRO}, TIME {MACRO}, PUT {MACRO} and REGEQU {MACRO}
CLMACLIB 00140           {COPY },      {COPY },     {COPY }            {COPY }

CLMACLIB 00143    If a  file named  OSMAC MACLIB existed  before issue  of the
CLMACLIB 00146    command, that file is erased.

CLMACLIB 00147    Assume that  the files ACCESS  MACRO, TIME COPY,  PUT MACRO,
CLMACLIB 00148    and REGEQU  COPY exist and  contain macros in  the following
CLMACLIB 00149    form.

CLMACLIB 00151    ACCESS MACRO      TIME COPY      PUT MACRO      REGEQU COPY
CLMACLIB 00152    ------------      ---------      ---------      -----------
CLMACLIB 00153       GET         *COPY TTIMER       PUT            XREG
CLMACLIB 00154                    TTIMER
CLMACLIB 00155       PUT         *COPY STIMER                      YREG
CLMACLIB 00156                    STIMER


CLMACLIB 00159    The resulting file  OSMAC MACLIB will contain  the following
CLMACLIB 00160    members:

CLMACLIB 00163         GET
CLMACLIB 00165         PUT
CLMACLIB 00167         TTIMER
CLMACLIB 00169         STIMER
CLMACLIB 00171         PUT
CLMACLIB 00173         REGEQU


CLMACLIB 00177    Note that the PUT macro, which appears twice in the input to
CLMACLIB 00178    the command  also appears twice  in the output.  The MACLIB
CLMACLIB 00179    command does not  check for duplicate macro names.  If at a
CLMACLIB 00179    later time the user were to request the macro PUT from OSMAC
CLMACLIB 00180    MACLIB, the first one encountered in the dictionary would be
CLMACLIB 00181    used.


CLMACLIB 00183    ADD Function

CLMACLIB 00184    The  ADD function  appends the  macro definitions  from  the
CLMACLIB 00185    specified  files  to  an existing  macro  library  with  the
CLMACLIB 00186    designated filename and a filetype of MACLIB. For example:

CLMACLIB 00193    ┌──────────────────────┐
CLMACLIB 00194    │ MACLIB ADD OSMAC DCB │
CLMACLIB 00195    └──────────────────────┘
```

Assume that OSMAC MACLIB has been created by the example in the explanation of the GEN function above and the file DCB COPY exists as follows.

```
*COPY DCB
 DCB macro definition
*COPY DCBD
 DCBD macro definition
```

The resulting OSMAC MACLIB will contain the following members:

```
GET
PUT
TTIMER
STIMER
PUT
REGEQU
DCB
DCBD
```

DEL Function

The DEL (delete) function removes the specified macro name from the macro library dictionary and compresses the dictionary so there are no unused entries. The macro definition still exists in the library, but since no dictionary entry exists it cannot be accessed or retrieved. If the user attempts to delete a macro for which two macro definitions exist in the macro library, only the first one encountered is deleted. For example:

```
MACLIB DEL OSMAC GET PUT TTIMER DCB
```

Deletes macro names GET, OPEN, TIME, and DCB from the dictionary of the macro library named OSMAC MACLIB, assuming that OSMAC exists as in the ADD function example. After the above command, it will contain the following members:

```
STIMER
PUT
REGEQU
DCBD
```

REP Function

REP (replace) is effectively a delete function followed by an ADD function. REP will delete the dictionary entry for the macro definition in the files specified by fn1-fnN. It

CLMACLIB 00261 then appends the new macro definitions to the macro library
CLMACLIB 00263 and creates new dictionary entries. For example, assume that
CLMACLIB 00264 a macro library MYMAC MACLIB contains the following members
CLMACLIB 00266 A, B, and C, and that the following command is entered:

```
MACLIB REP MYMAC A C
```

CLMACLIB 00273 The files represented by file identifiers A MACRO and C
CLMACLIB 00274 MACRO each have one macro definition. After execution of
CLMACLIB 00275 the command, MYMAC MACLIB will contain members with the same
CLMACLIB 00275 names as before, but the contents of A and C will be
CLMACLIB 00276 different.

CLMACLIB 00279 COMP Function

CLMACLIB 00281 Execution of a MACLIB command with the DEL or REP functions
CLMACLIB 00281 can result in unused space within a macro library. The COMP
CLMACLIB 00282 (compress) function is used to compress a macro library
CLMACLIB 00282 (that is, remove any macros for which there is no dictionary
CLMACLIB 00283 entry). This function uses a temporary data set named
CLMACLIB 00285 MACLIB CMSUT1. For example, the command:

```
MACLIB COMP MYMAC
```

CLMACLIB 00293 Compresses the library MYMAC MACLIB.

CLMACLIB 00295 MAP Function

CLMACLIB 00296 The MAP function creates a list containing the name of each
CLMACLIB 00297 macro in the dictionary, the size of the macro and its
CLMACLIB 00297 position within the macro library. The user may specify the
CLMACLIB 00298 device to which the list is to be written. Acceptable
CLMACLIB 00299 devices are:

CLMACLIB 00302 DISK — the list is placed in a file with the file
CLMACLIB 00303 identifier 'libname MAP A1'.

CLMACLIB 00306 PRINT — the list is spooled to the printer.

CLMACLIB 00309 TERM — the list is typed at the terminal.

CLMODMAP 00003 **MODMAP**

CLMODMAP 00007 The MODMAP command types the load map associated with the
CLMODMAP 00008 specified MODULE file. Two types of modules contain no load
CLMODMAP 00008 map and therefore produce an error message when they are
CLMODMAP 00009 specified in a MODMAP command. They are (1) CMS transient
CLMODMAP 00011 area modules and (2) MODULE files created with the GENMOD
CLMODMAP 00012 command using the NOMAP option.

CLMODMAP 00019
CLMODMAP 00020 | MODMAP | fn |
CLMODMAP 00021 | MOD | |
CLMODMAP 00022

CLMODMAP 00028 fn is the filename of the MODULE file whose load
CLMODMAP 00029 map is to be typed. The filetype of the file
CLMODMAP 00030 must be MODULE.

CLMODMAP 00033 **Responses**

CLMODMAP 00034 The LOAD MAP associated with the file will be typed at the
CLMOVEFI 00001 terminal.

CLMOVEFI 00003 **MOVEFILE**

CLMOVEFI 00006 The MOVEFILE command allows the user to move data from any
CLMOVEFI 00006 device supported by VM/370 to any other device supported by
CLMOVEFI 00007 VM/370. The command accepts two ddnames as arguments. The
CLMOVEFI 00008 FILEDEF command must have been used to specify devices or
CLMOVEFI 00009 disk files which these ddnames represent. The command moves
CLMOVEFI 00010 data records from the device or file specified by the first
CLMOVEFI 00011 ddname to the device or file specified by the second ddname.

CLMOVEFI 00022
CLMOVEFI 00023
CLMOVEFI 00024
CLMOVEFI 00025
CLMOVEFI 00026
CLMOVEFI 00027

CLMOVEFI 00032 input-ddname specifies the ddname representing a file
CLMOVEFI 00033 definition which describes the device or disk
CLMOVEFI 00033 file on which the data to be moved currently
CLMOVEFI 00034 resides. If ddname is not specified, the
CLMOVEFI 00035 default name INMOVE is used.

CLMOVEFI 00038 output-ddname specifies the ddname representing a file
CLMOVEFI 00039 definition which describes the device or disk
CLMOVEFI 00039 file to which the data is to be written. If
CLMOVEFI 00040 ddname is not specified, the default name
CLMOVEFI 00041 OUTMOVE is used.

CLMOVEFI 00043 **Note:** Normally the user should have used the FILEDEF command
CLMOVEFI 00044 to establish device characteristics for the ddname specified
CLMOVEFI 00045 with the command. If the user does not do this, the OS Macro
CLMOVEFI 00046 Simulation routines will supply default characteristics.
CLMOVEFI 00047 For example, if the ddname is X a default FILEDEF X DISK
CLMOVEFI 00048 FILE X A1 command will be performed.

CLMOVEFI 00051 **Default Device Attributes:** If the user has not specified
CLMOVEFI 00051 certain file attributes in his file definition (such as
CLMOVEFI 00052 record format and blocksize), the MOVEFILE command assumes
CLMOVEFI 00053 the values shown in Figure 24.

Device	INPUT DDNAME		OUTPUT DDNAME	
	RECFM	BLOCKSIZE	RECFM	BLOCKSIZE
Card Reader	F	80	N/A	N/A
Card Punch	N/A	N/A	F	80
Printer	N/A	N/A	U	133
Terminal	U	130	U	130
Tape *	U	3600	RECFM of Input ddname	BLOCK of Inp ddname
Existing Disk file	RECFM of file	BLOCK of file	RECFM of file	BLOCK of file
New Disk file	N/A	N/A	RECFM of Input ddname	BLOCK of Inp ddname
Dummy	N/A	N/A	RECFM of Input ddname	BLOCK of Inp ddname

* If the default record format and blocksize are used in a tape-to-tape move operation and an input record is greater than 3600 bytes, it will be truncated to 3600 bytes on the output tape.

Figure 24. Default Device Attributes for MOVEFILE command

Example

```
MOVEFILE | NEWMAST  OLDMAST
```

Action: The existing file referred to by the FILEDEF ddname NEWMAST is moved to the file whose definition is referred to by a FILEDEF ddname of OLDMAST.

Responses

DMSMVE706I CONSOLE INPUT -- TYPE NULL LINE FOR END OF DATA

The input-ddname in the MOVEFILE command refers to a device

```
CLMOVEFI 00118     type of  TERMINAL.  This message  prompts the user  to begin
CLMOVEFI 00119     entering the  input data and instructs  him to enter  a null
CLPRINT  00001     line to terminate input.
```

CLPRINT 00012 **PRINT**

CLPRINT 00014 The PRINT command prints the specified file on the spooled
CLPRINT 00017 virtual printer. The file may contain carriage control
CLPRINT 00017 characters and may have either fixed or variable length
CLPRINT 00018 records.

CLPRINT 00020
CLPRINT 00021 | PRINT | fn ft [fm] [(options...)] |
CLPRINT 00022 | PR | |
CLPRINT 00023 | | options: [CC|NOCC] |
CLPRINT 00024 | | [UPCASE] |
CLPRINT 00025 | | ┌ ┐ |
CLPRINT 00026 | | |MEMBER(*)| |
CLPRINT 00027 | | | (name)| |
CLPRINT 00028 | | └ ┘ |
CLPRINT 00029 | | [HEX] |
CLPRINT 00030

CLPRINT 00035 **fn** specifies the filename of the file to be
CLPRINT 00037 printed. This field must be specified.

CLPRINT 00040 **ft** specifies the filetype of the file to be
CLPRINT 00042 printed. This field must be specified.

CLPRINT 00045 **fm** specifies the filemode of the file to be
CLPRINT 00046 printed. If this field is specified as an
CLPRINT 00046 asterisk(*), the standard order of search is
CLPRINT 00047 followed and the first file encountered with
CLPRINT 00048 the given filename and filetype is printed.
CLPRINT 00048 If fm is not specified, the primary disk and
CLPRINT 00049 its extensions are searched.

CLPRINT 00052 **Options**

CLPRINT 00054 CC specifies that the first character of each
CLPRINT 00055 record is to be interpreted as a carriage
CLPRINT 00056 control character. If the filetype is
CLPRINT 00057 LISTING, this is the default.

CLPRINT 00060 NOCC specifies that the first character of each
CLPRINT 00061 record is not to be interpreted as a carriage
CLPRINT 00062 control character.

CLPRINT 00067 UPCASE specifies that lowercase characters in the
CLPRINT 00069 none file are to be translated to uppercase in the
CLPRINT 00072 UP printout.

CLPRINT 00077 MEMBER(*) if ft is MACLIB, this option may be specified.
CLPRINT 00079 none If * is entered, all individual members of

CLPRINT 00084 MEM (name) that MACLIB are printed. If a name is
CLPRINT 00085 specified, only the named member is printed.

CLPRINT 00088 HEX specifies that the file is to be printed in
CLPRINT 00089 hexadecimal format.

CLPRINT 00095 Example 1

CLPRINT 00097
CLPRINT 00098 | PRINT | ACCESS MACLIB (MEMBER GET) |
CLPRINT 00099

CLPRINT 00103 Action: The contents of member GET in file ACCESS MACLIB is
CLPRINT 00104 spooled to the printer. The first character of each record
CLPRINT 00105 in the file is not treated as carriage control.

CLPRINT 00107 Example 2

CLPRINT 00109
CLPRINT 00110 | PRINT | OLDMAST NAME (CC) |
CLPRINT 00111

CLPRINT 00114 Action: The contents of the file OLDMAST NAME are spooled to
CLPRINT 00115 the printer and the first character of each record in the
CLPRINT 00118 file is used for carriage control.

CLPRINT 00120 Responses

CLPRINT 00121 The READY message indicates the command has completed
CLPRINT 00122 without error. (that is, that the file has been written to
CLPRINT 00123 the spooled printer). The file is now under the control of
CLPUNCH 00001 CP's spooling functions.

PUNCH

The PUNCH command causes a specified disk file to be spooled to the punch. PUNCH accepts fixed or variable length records so long as no record exceeds 80 characters. Records with less than 80 characters are right padded with blanks to 80 characters. Records larger than 80 characters are rejected.

PUNCH PU	fn ft [fm]	((HEADER)) ((NOHEADER))

fn specifies the filename of the file to be punched. This field must be specified.

ft specifies the filetype of the file to be punched. This file must be specified.

fm specifies the filemode of the file to be punched. If this field is specified as an asterisk(*), the standard order of search is followed and the first acceptable file encountered is punched. If fm is not specified, the primary disk and its extensions are searched.

Options

HEADER (none, **H**) indicates that a control card is to be inserted in the punched output preceding the specified file. This control card is intended for use in identifying the file for a subsequent READCARD command to restore the file to a disk. The control card format is shown in Figure 25.

Column	Number of Characters	Contents	Meaning
1	1	:	identifies card as a control card
2–5	4	READ	identifies card as a READ control card
6–7	2	blank	
8–15	8	fname	filename of the file punched
16	1	blank	
17–24	8	ftype	filetype of the file punched
25	1	blank	
26–27	2	fmode	filemode of the file punched
28	1	blank	
29–34	6	volid	label of the disk from which the file was read
35	1	blank	
36–43	8	mm.dd.yy	the date that the file was last written
44	1	blank	
45–52	8	hh.mm.ss	the time of day that the file was written to disk
53–80	28	blank	

Figure 25. Header Card Format

NOHEADER none
NOH

no header control card is to be inserted in the punched deck.

Example

```
| PUNCH | NEWMAST TRANS (NOH)
```

Action: The file NEWMAST TRANS is spooled to the punch. No header card is punched preceding the output deck.

CLPUNCH 00124 **Responses**

CLPUNCH 00127 If the command has completed without error (the file has
CLPUNCH 00129 been successfully spooled), the READY message appears. The
CLQUERY 00001 file is now under control of CP spooling functions.

CLQUERY 00005 **QUERY**

CLQUERY 00008 The QUERY command provides the user with certain information
CLQUERY 00009 about his virtual machine environment. Information which
CLQUERY 00010 may be obtained includes:

CLQUERY 00013 1. the state of any of the virtual machine characteristics
CLQUERY 00015 which are maintained by the CMS SET command.

CLQUERY 00018 2. file definitions (FILEDEFs) which are in effect.

CLQUERY 00021 3. the status of the disks attached to the user's virtual
CLQUERY 00022 machine.

```
CLQUERY 00032 ┌──────────────────────────────────────────┐
CLQUERY 00033 │ QUERY  │ { BLIP                  }       │
CLQUERY 00034 │ Q      │ { RDYMSG                }       │
CLQUERY 00035 │        │ { LDRTBLS               }       │
CLQUERY 00036 │        │ { RELPAGE               }       │
CLQUERY 00037 │        │ { IMPCP                 }       │
CLQUERY 00038 │        │ { IMPEX                 }       │
CLQUERY 00039 │        │ { ABBREV                }       │
CLQUERY 00040 │        │ { REDTYPE               }       │
CLQUERY 00041 │        │ { PROTECT               }       │
CLQUERY 00042 │        │ { SEARCH                }       │
CLQUERY 00043 │        │ {          ┌    ┐       }       │
CLQUERY 00044 │        │ {          |mode|       }       │
CLQUERY 00045 │        │ { DISK     | *  |       }       │
CLQUERY 00046 │        │ {          └    ┘       }       │
CLQUERY 00047 │        │ {          {SYSTEM}     }       │
CLQUERY 00048 │        │ { SYNONYM  {USER  }     }       │
CLQUERY 00049 │        │ {          {ALL   }     }       │
CLQUERY 00050 │        │ {                       }       │
CLQUERY 00051 │        │ { FILEDEF               }       │
CLQUERY 00052 │        │ { MACLIB                }       │
CLQUERY 00053 │        │ { TXTLIB                }       │
CLQUERY 00054 │        │ { LIBRARY               }       │
CLQUERY 00055 │        │ { INPUT                 }       │
CLQUERY 00056 │        │ { OUTPUT                }       │
CLQUERY 00057 └──────────────────────────────────────────┘
```

CLQUERY 00062 **SET Command Functions**: The SET Command description presents
CLQUERY 00069 detailed descriptions of the following functions:

CLQUERY 00076 BLIP displays the BLIP character(s).

CLQUERY 00078 response: BLIP = {xxxxxxxx}
CLQUERY 00080 {NONE }

```
CLQUERY  00084    RDYMSG    displays the RDYMSG format.

CLQUERY  00088
CLQUERY  00090              response: RDYMSG = ┌     ┐
CLQUERY  00091                                 |LMSG|
CLQUERY  00092                                 |SMSG|
CLQUERY  00093                                 └     ┘
CLQUERY  00095                        LMSG--standard    CMS   READY    message
CLQUERY  00097                              provided. For example:
CLQUERY  00099                              R; T = 0.12/0.33 17:06:20
CLQUERY  00099                        SMSG--shortened   CMS   READY   message
CLQUERY  00099                              excluding   date   and   time   is
CLQUERY  00100                              provided. For example:
CLQUERY  00102                              R;

CLQUERY  00107    LDRTBLS   displays the number of loader tables the user has.

CLQUERY  00109              response:  LDRTBLS = nn

CLQUERY  00111    RELPAGE   indicates whether pages of storage in which
CLQUERY  00112              certain commands have completed execution are to
CLQUERY  00113              be released or retained.

CLQUERY  00115                                 ┌     ┐
CLQUERY  00117              response: RELPAGE = |ON   |
CLQUERY  00118                                 |OFF  |
CLQUERY  00119                                 └     ┘
CLQUERY  00120                        ON---pages will be released.
CLQUERY  00122                        OFF--pages will be retained.

CLQUERY  00126    IMPCP     displays status of implied CP command indicator.

CLQUERY  00128                                 ┌     ┐
CLQUERY  00130              response: IMPCP  = |ON   |
CLQUERY  00131                                 |OFF  |
CLQUERY  00132                                 └     ┘
CLQUERY  00136                        ON---commands not recognized by CMS will
CLQUERY  00137                             be passed to CP.
CLQUERY  00138                        OFF--commands not recognized by CMS will
CLQUERY  00140                             be flagged as unknown to CMS.

CLQUERY  00146    IMPEX     displays status of implied EXEC indicator.

CLQUERY  00148                                 ┌     ┐
CLQUERY  00150              response: IMPEX  = |ON   |
CLQUERY  00151                                 |OFF  |
CLQUERY  00152                                 └     ┘
CLQUERY  00155                        ON---EXEC files may be executed by
CLQUERY  00156                             entering filename of file.
CLQUERY  00157                        OFF--the EXEC command must be explicitly
CLQUERY  00158                             entered to execute EXEC files.

CLQUERY  00161    ABBREV    displays the status of the minimum abbreviation
CLQUERY  00162              indicator.
```

```
CLQUERY  00165
CLQUERY  00167              response: ABBREV = ┌ ┐
CLQUERY  00168                                 │ON │
CLQUERY  00169                                 │OFF│
CLQUERY  00170                                 └ ┘
CLQUERY  00171                        ON---minimum  CMS command  abbreviations
CLQUERY  00172                             (truncations) will be accepted.
CLQUERY  00173                        OFF--minimum    command    abbreviations
                                           (truncations) will not be accepted.

CLQUERY  00177     REDTYPE    displays the status of the REDTYPE indicator.

CLQUERY  00180
CLQUERY  00182              response: REDTYPE = ┌ ┐
CLQUERY  00183                                  │ON │
CLQUERY  00184                                  │OFF│
CLQUERY  00187                                  └ ┘
CLQUERY  00188                        ON---CMS error messages will be typed in
                                           red, providing the  terminal  is
CLQUERY  00188                             equipped with the  two-color ribbon
CLQUERY  00189                             feature.
CLQUERY  00190                        OFF--CMS  error  messages  will  not  be
CLQUERY  00191                             typed in red.


CLQUERY  00197     PROTECT    displays the status of CMS nucleus protection.

CLQUERY  00199
CLQUERY  00201              response: PROTECT = ┌ ┐
CLQUERY  00202                                  │ON │
CLQUERY  00203                                  │OFF│
                                                =   ┘

CLQUERY  00205                        ON---CMS   nucleus   protection   is   in
CLQUERY  00206              effect.
CLQUERY  00207                        OFF--CMS  nucleus protection  is not  in
CLQUERY  00208              effect.


CLQUERY  00212     CMS Disk Status Functions


CLQUERY  00214     SEARCH     displays  the  search  order  of  all  CMS  disks
CLQUERY  00215              currently accessible.

CLQUERY  00219
CLQUERY  00221              response: volid vaddr mode ┌ ┐
CLQUERY  00222                        .                │R/O│
CLQUERY  00223                        .                │R/W│
CLQUERY  00224                        .                └ ┘
```

```
CLQUERY  00230    DISK      mode  displays  the  status  of  the  single  disk
CLQUERY  00231              represented by 'mode'.

CLQUERY  00236              response: mode (vaddr): nn FILES, nnnn REC IN USE, nnnn LEFT (CF
CLQUERY  00237
CLQUERY  00238                        nn% FULL (n CYL), type|R/O|
CLQUERY  00239                                             |R/W|
CLQUERY  00240                                             L   J

CLQUERY  00243              If the disk  with the specified mode  has not been
CLQUERY  00244              accessed, the response is

CLQUERY  00246                   DISK 'mode' NOT ACCESSED

CLQUERY  00250    DISK      *     displays the status of all CMS disks.

CLQUERY  00255              response: same as for QUERY DISK mode; one line is
CLQUERY  00257                        typed for each accessible disk.


CLQUERY  00262    Other Functions


CLQUERY  00265    SYNONYM   SYSTEM  displays  CMS system  synonyms in  effect.
CLQUERY  00265              "Appendix H:  Minimum Truncations for  CMS Command
CLQUERY  00266              Names" presents a summary of system synonyms.

CLQUERY  00270                              SYSTEM      SHORTEST
CLQUERY  00271              response:       COMMAND       FORM
CLQUERY  00272                              command  minimum truncation
CLQUERY  00273                                 .           .
CLQUERY  00274                                 .           .
CLQUERY  00275                                 .           .

CLQUERY  00278              If no system synonyms are in effect, the following
CLQUERY  00280              message is typed out:

CLQUERY  00282                   NO SYSTEM SYNONYMS IN EFFECT

CLQUERY  00285    SYNONYM   USER    displays user synonyms in effect.

CLQUERY  00288              response:     SYSTEM   USER    SHORTEST
CLQUERY  00289                            COMMAND  SYNONYM  FORM (IF ANY)
CLQUERY  00290                            command  synonym  minimum truncation
CLQUERY  00291                               .        .          .
CLQUERY  00292                               .        .          .
CLQUERY  00293                               .        .          .

CLQUERY  00296              If no user  synonyms are in effect,  the following
CLQUERY  00297              message is typed out:

CLQUERY  00299                   NO USER SYNONYMS IN EFFECT
```

```
CLQUERY  00302    SYNONYM   ALL     displays all synonyms in effect.

CLQUERY  00306              response: Same as SYNONYM SYSTEM and SYNONYM USER.


CLQUERY  00313    FILEDEF   displays all file definitions in effect.

CLQUERY  00315              response: ddname device [fn [ft]]

CLQUERY  00316              If there are  no user file definitions  in effect,
CLQUERY  00318              the following message is typed out:

CLQUERY  00320              NO USER DEFINED FILEDEF'S IN EFFECT


CLQUERY  00323    MACLIB    displays the names of all files with a filetype of
CLQUERY  00323              MACLIB (libraries  to  be  searched  for  macro
CLQUERY  00324              definitions).

CLQUERY  00326              response: MACLIB = libname...

CLQUERY  00327              If no MACLIBS are  searched for macro definitions,
CLQUERY  00328              the response is

CLQUERY  00330                   MACLIB = NONE


CLQUERY  00333    TXTLIB    displays the names of all files with a filetype of
CLQUERY  00333              TXTLIB (libraries  to  be  searched for  unresolved
CLQUERY  00334              references).

CLQUERY  00336              response: TXTLIB = libname...

CLQUERY  00337              If  no   TXTLIBs  are   searched  for   unresolved
CLQUERY  00339              references, the following message is typed out:

CLQUERY  00341                   TXTLIB = NONE


CLQUERY  00343    LIBRARY   displays the names of all  MACLIB and TXTLIB files
CLQUERY  00345              which are to be searched.

CLQUERY  00347              response: MACLIB = libname...|NONE
CLQUERY  00349                        TXTLIB = libname...|NONE


CLQUERY  00351    INPUT     displays the contents of  the user input translate
CLQUERY  00353              table if he has specified one.

CLQUERY  00357              response:  INPUT  a1  xx1
CLQUERY  00358                             .   .
CLQUERY  00359                             .   .
CLQUERY  00360                             .   .
CLQUERY  00361                            an  xxn
```

```
CLQUERY  00362            If no user translate table is in effect, the
CLQUERY  00363            response is

CLQUERY  00365            NO USER DEFINED INPUT TRANSLATE TABLE IN EFFECT


CLQUERY  00367   OUTPUT   displays the contents of the user output translate
CLQUERY  00369            table if he has specified one.

CLQUERY  00371            response:  OUTPUT  xx1  a1
CLQUERY  00372                               •    •
CLQUERY  00373                               •    •
CLQUERY  00374                               •    •
CLQUERY  00375                               xxn  an

CLQUERY  00376            If no user output translate table has been
CLQUERY  00377            defined, the response is

CLQUERY  00379            NO USER DEFINED OUTPUT TRANSLATE TABLE IN EFFECT


CLQUERY  00383   Responses

CLQUERY  00384   The status of the specified function is typed on the
CLREADCD 00001   terminal in the format shown following the command.
```

READCARD

The READCARD command reads data records from the spooled card unit input device and creates files on disk containing the data records. The data records must be fixed length and are normally 80 characters although they may consist of up to 132 characters. Records of less than 80 characters cannot be read. If a file exists with the same identifiers as the one to be created it is erased.

The deck entered through the card reader may contain any number of files, each immediately preceded by a READ control card specifying the filename, filetype, and optionally, filemode. These control cards are typed at the terminal as they are encountered, and are interpreted by the system just as if fn ft fm had been entered from the terminal. Any existing file with the same identifiers as those specified on any of the READ cards is erased. Each READ control card ends the file preceding it, and the last file is ended by the end-of-file indication by the reader.

User card decks must have been spooled into the user's virtual reader before a READCARD command can be issued. The user need not be logged in at the time the decks are transferred to his virtual reader. Decks may be entered separately, and each must be preceded by an identification card (the characters ID punched in columns 1 and 2 and the user's userid starting in column 10). CP saves the deck until the user logs in and requests it with a READCARD command. If more than one deck has been read resulting in more than one file in the virtual reader, the decks are processed by successive READCARD commands in the order in which they were entered.

```
┌──────────────────────────────────────────────┐
| READCARD | {          ┌    ┐ }              |
| READ     | { fn ft    | fm | }              |
|          | {          | A1 | }              |
|          | {          └    ┘ }              |
|          | { *               }              |
|          | {          ┌    ┐ }              |
|          | { *    *   | fm | }              |
|          | {          | A1 | }              |
|          | {          └    ┘                |
└──────────────────────────────────────────────┘
```

fn specifies the filename of the file to be read. An asterisk may be coded in this field.

ft specifies the filetype of the file to be read. An asterisk may be coded in this field if fn was coded as an asterisk.

CLREADCD 00067 fm specifies the filemode of the file to be read. If
CLREADCD 00069 this field is omitted, A1 is assumed.

CLREADCD 00074 **Forms of the READCARD Parameters**

CLREADCD 00077 1. If filename and filetype are specified with the
CLREADCD 00079 READCARD command, only one file is read in.

CLREADCD 00082 2. If the file designations are to be entered in the card
CLREADCD 00083 stream, a single asterisk must be specified with the
CLREADCD 00084 READCARD command. If this form of the command is
CLREADCD 00085 specified, and the first card in the input stream is
CLREADCD 00086 not a valid READ control card, a file named READ CMSUT1
CLREADCD 00087 A1 is set up to contain all data read until a valid
CLREADCD 00088 READ control card is encountered.

CLREADCD 00091 3. Two asterisks accept the filename and filetype from the
CLREADCD 00092 card deck, but use a filemode of A1.

CLREADCD 00095 4. Two asterisks with a mode specified accept the
CLREADCD 00096 identifiers from the card deck, but use the filemode
CLREADCD 00097 specified.

Column	Number of Characters	Contents	Meaning
1	1	:	identifies card as a control card
2–5	4	READ	identifies card as a READ control card
6–7	2	blank	
8–15	8	fname	filename of the file punched
16	1	blank	
17–24	8	ftype	filetype of the file punched
25	1	blank	
26–27	2	fmode	filemode of the file punched
28	1	blank	
29–34	6	volid	label of the disk from which the file was read
35	1	blank	
36–43	8	mm.dd.yy	the date that the file was last written
44	1	blank	
45–52	8	hh.mm.ss	the time of day that the file was written to disk
53–80	28	anything	

Figure 26. Format of the READ Control Card

Responses

After the command READCARD *, control cards encountered in the input card stream are typed at the terminal.

DMSRDC701I NULL FILE

The spooled card reader contains no files for the user issuing the READCARD command

DMSRDC702I READ CONTROL CARD IS MISSING. FOLLOWING ASSUMED:

```
CLREADCD 00166                READ READCARD CMSUT1 A1

CLREADCD 00168    The first card  in the deck is not a  READCARD control card.
CLREADCD 00170    Therefore the file READ CMSUT1 A1 is created.


CLREADCD 00172    DMSRDC738I RECORD LENGTH = 'nnn' BYTES

CLREADCD 00174    The records being  read are not 80 bytes  long; this message
CLREADCD 00175    gives the length.

CLREADCD 00177    When a  READCARD control card  is encountered, the  first 43
CLRELEAS 00001    columns of the card are typed to the terminal.
```

CLRELEAS 00008 **RELEASE**

CLRELEAS 00010 The RELEASE command frees an active disk when the user no
CLRELEAS 00011 longer needs it. An ACCESS command must have been previously
CLRELEAS 00012 entered for the specified disk.

CLRELEAS 00014
CLRELEAS 00015 | RELEASE | |
CLRELEAS 00018 | REL | { ccu } |
CLRELEAS 00019 | | {mode } |
CLRELEAS 00020

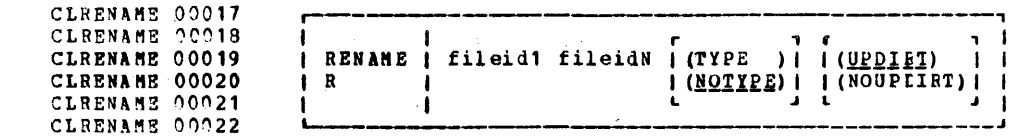
CLRELEAS 00025 ccu specifies the virtual device address of the disk
CLRELEAS 00026 that is to be released.

CLRELEAS 00030 mode specifies the mode of the disk to be released.

CLRELEAS 00032 Note: If a disk is accessed more than once, the RELEASE ccu
CLRELEAS 00033 form of the command will cause all instances of ccu to be
CLRENAME 00001 released. The system disk cannot be released.

RENAME

The RENAME command is used to change the file identifier of one or more files. The RENAME command may not be used on a file which is located on a read/only disk.



fileid1 specifies the original file identification of the file whose name is to be changed. All components of the fileid (filename, filetype, and filemode) must be coded, either with a name or an asterisk. If an asterisk is coded in any field it indicates that any fileid which satisfies the other qualifications will be renamed. For example, if fileid is coded as A * A1, all files on the A disk with a filename of A will be renamed.

fileidN specifies the new file identification of the file. All components of the fileid (filename, filetype, and filemode) must be coded, either with a name or an equal sign. If an equal sign is coded, the corresponding fileid field is left unchanged.

Options

TYPE
none
T
specifies that the new identifiers of all files renamed are to be typed at the terminal. Typing takes place only when an * appears in the fileid.

NOTYPE
none
NOT
specifies that the new file identifiers of all files renamed are not to be typed at the terminal.

UPDIRT
none
UP
specifies that the user's file directory is to updated upon completion of this command.

NOUPDIRT
none
NOUP
specifies that the user's file directory is not to be updated upon completion of this command. Normally, user file directories are updated at the completion of each CMS command that affects disk files.

CLRENAME 00084 **Responses**

CLRENAME 00086 xxxxxxxx xxxxxxxx xx

CLRENAME 00089 The TYPE option has been specified and an asterisk appears
CLRENAME 00090 in the fileid. The new filename, filetype, and filemode of
CLRENAME 00091 each file altered is typed.

CLRENAME 00092 If fileidN is the name of an existing file, an error message
CLRUN 00001 is generated.

CLRUN 00011 **RUN**

CLRUN 00014 RUN is used to initiate an automated series of functions on
CLRUN 00015 a file. RUN is capable of compiling, loading, and starting
CLRUN 00016 execution of the specified file, depending upon the
CLRUN 00018 filetype. The acceptable filetypes are EXEC, MODULE, TEXT,
CLRUN 00019 and those required by the language processors. The RUN
CLRUN 00020 command is actually an EXEC procedure and if it is executed
CLRUN 00020 from within an EXEC file, it must be preceded by the EXEC
CLRUN 00021 command name.

CLRUN 00023 ┌──────────────────────────────┐
CLRUN 00024 | RUN | fn [ft [fm]] [(args...)] |
CLRUN 00025 └──────────────────────────────┘

CLRUN 00031 fn specifies the filename of the file to be
CLRUN 00033 manipulated. This field must be entered.

CLRUN 00035 ft specifies the filetype of the file to be
CLRUN 00036 manipulated. If filetype is not specified, a
CLRUN 00037 search is made for a file with the specified
CLRUN 00038 filename and the filetype of (search to be
CLRUN 00039 performed in this order): EXEC, MODULE, TEXT. If a
CLRUN 00039 filetype is specified which is one required as
CLRUN 00040 input to a language processor, the respective
CLRUN 00041 processor is invoked to compile the source code
CLRUN 00042 included in the file and to produce a TEXT file.
CLRUN 00043 LOAD and START are then called to initiate program
CLRUN 00044 execution. The filetype must be specified if
CLRUN 00045 filemode is specified. Valid filetypes and actions
CLRUN 00046 performed on the specific filetype for this
CLRUN 00047 command are:

CLRUN 00051 EXEC The file is assumed to contain one
CLRUN 00052 or more CMS commands and the EXEC
CLRUN 00054 processor is called to process the file.

CLRUN 00057 MODULE The file is assumed to be one created by
CLRUN 00057 the GENMOD command. It must have an
CLRUN 00058 entry point with the same name as fn.
CLRUN 00059 LOADMOD is called to load the program
CLRUN 00059 into main storage and START is called to
CLRUN 00060 begin execution of the program at the
CLRUN 00061 entry point equal to fn.

CLRUN 00064 TEXT The file is assumed to contain
CLRUN 00064 relocatable object code as produced by
CLRUN 00065 the language processors. It must have an
CLRUN 00066 entry point with the same name as fn.
CLRUN 00066 LOAD is called to bring the file into
CLRUN 00067 main storage in an executable format and
CLRUN 00068 START is called to execute the program
CLRUN 00069 beginning at the entry point named by

```
CLRUN   00070                   fn.

CLRUN   00075   fm      specifies the filemode of the file to be
CLRUN   00076           manipulated. If this field is specified, a
CLRUN   00077           filetype must be specified. If fm is not
CLRUN   00078           specified, the default search order is used to
CLRUN   00080           search the user's disks for the file.

CLRUN   00083   args    are one or more user arguments to be used during
CLRUN   00084           execution. If compiling or loading is to be
CLRUN   00085           performed, the default options for these functions
CLRUN   00086           are assumed. The user may specify up to thirty
CLRUN   00087           arguments in the RUN command, provided they fit on
CLRUN   00088           a single input line. These arguments are used
CLRUN   00089           during operation of an EXEC file, or during
CLRUN   00090           execution of a MODULE or TEXT file. The arguments
CLRUN   00091           are set up as a string of doublewords, one
CLRUN   00091           argument per doubleword. The address of this
CLRUN   00092           string is passed to the specified file at
CLRUN   00093           execution time. Each argument is left-justified,
CLRUN   00094           and any argument more than eight characters long
CLRUN   00095           is truncated on the right. With an EXEC file, any
CLRUN   00096           arguments specified in the RUN command replace the
CLRUN   00096           corresponding &n operands in the individual
CLRUN   00098           commands of the EXEC file.

CLRUN   00099           With a file whose filetype is other than EXEC, the
CLRUN   00100           arguments are placed in a string as described
CLRUN   00101           above. The address of the string may be obtained
CLRUN   00102           by adding 8 to the address contained in general
CLRUN   00103           register 1 at the time execution of the specified
CLRUN   00104           program begins. Additional arguments may be
CLRUN   00105           obtained by using displacements of 16, 24, 32, and
CLRUN   00105           so forth, from the address thus obtained until a
CLRUN   00106           hexadecimal 'FF' is encountered as the first byte
CLRUN   00107           of an argument field. This signals the end of the
CLRUN   00108           argument list.
```

CLSET 00004 **SET**

CLSET 00006 The SET command is used to establish, turn off, or reset a
CLSET 00007 particular function of the user's virtual machine. Only one
CLSET 00008 function may be specified per SET command.

CLSET 00012 **Note**: If the function is one of those performed by the CP
CLSET 00012 SET command, CMS will pass the command input to the control
CLSET 00013 program for processing.

```
CLSET 00024 ┌──────────────────────────────────────────────────────┐
CLSET 00025 │ SET │  function                                      │
CLSET 00026 │     │                                                │
CLSET 00027 │     │  functions: [BLIP {string[ (count)] |ON|OFF}]  │
CLSET 00028 │     │             [RDYMSG {SMSG|LMSG}]               │
CLSET 00029 │     │             [LDRTBLS [nn]]                     │
CLSET 00030 │     │             [RELPAGE {ON|OFF}]                 │
CLSET 00031 │     │             [INPUT [a xx]]                     │
CLSET 00032 │     │             [OUTPUT [xx a]]                    │
CLSET 00033 │     │             [ABBREV {ON|OFF}]                  │
CLSET 00034 │     │             [REDTYPE {ON|OFF}]                 │
CLSET 00035 │     │             [IMPEX {ON|OFF}]                   │
CLSET 00036 │     │             [IMPCP {ON|OFF}]                   │
CLSET 00037 │     │             [PROTECT {ON|OFF}]                 │
CLSET 00038 └──────────────────────────────────────────────────────┘
```

CLSET 00043 **Functions**

CLSET 00048 **BLIP string[(count)]** defines the characters which CMS uses
CLSET 00048 to notify the CMS user of every two CPU
CLSET 00049 seconds of real execution time. Up to
CLSET 00049 eight characters may be defined, and if
CLSET 00049 trailing blanks are desired, the count
CLSET 00050 field must be used. ON and OFF may not
CLSET 00051 be used as BLIP characters.

CLSET 00054 ON returns the default BLIP character
CLSET 00054 string which is a string of unprintable
CLSET 00055 characters.

CLSET 00058 OFF turns off BLIP

CLSET 00060 **RDYMSG** LMSG provides the user with the standard CMS
CLSET 00061 READY message including date and time.

CLSET 00063 SMSG provides the user with a shortened form
CLSET 00064 of the CMS READY message not including
CLSET 00065 date and time.

CLSET 00068 **LDRTBLS** nn The user may change the number (nn) of
CLSET 00071 02 pages of storage used for loader tables

```
CLSET    00071                          Normally, a virtual machine having from
CLSET    00071                          256K to 512K of addressable main
CLSET    00072                          storage has two pages of loader tables;
CLSET    00072                          a larger virtual machine has three.
CLSET    00073                          This number may be changed with SET
CLSET    00073                          LDRTBLS nn provided that (1) nn is a
CLSET    00073                          decimal number less than 128, and (2)
CLSET    00074                          the user has enough storage available
CLSET    00074                          to allow nn pages to be used for loader
CLSET    00074                          tables.  If both of these conditions
CLSET    00075                          are met, nn pages are set aside for
CLSET    00076                          loader tables.

CLSET    00079    RELPAGE   ON          When CMS is running in a virtual
CLSET    00081              OFF         machine, pages of storage are released
CLSET    00082                          and set to binary zeroes after the
CLSET    00082                          following commands complete execution:
CLSET    00082                          ASSEMBLE, COBOL*, COPYFILE, COMPARE,
CLSET    00083                          EDIT, MACLIB, PLIOPT*, SORT, TAPE,
CLSET    00083                          TXTLIB, and UPDATE.  This is the normal
CLSET    00083                          mode of operation -- that obtained by
CLSET    00084                          default or when SET RELPAGE ON is
CLSET    00084                          specified.  However, if a user wishes
CLSET    00084                          to examine storage after any of these
CLSET    00085                          commands has finished -- for example,
CLSET    00085                          for debugging or analyzing a problem --
CLSET    00085                          the command SET RELPAGE OFF inhibits
CLSET    00087                          this releasing feature.

CLSET    00089    INPUT     a  xx       translate the specified character (a)
CLSET    00089                          to the specified hexadecimal code (xx)
CLSET    00090                          upon input from the terminal.

CLSET    00092              (blank)     return all characters to their default
CLSET    00092                          translation upon input from the
CLSET    00093                          terminal.

CLSET    00095    OUTPUT    xx a        translate the specified hexadecimal
CLSET    00096                          representation (xx) to the specified
CLSET    00096                          character (a) upon output to the
CLSET    00097                          terminal.

CLSET    00099              (blank)     return all characters to their default
CLSET    00099                          translation upon input from the
CLSET    00100                          terminal.


CLSET    00103    ABBREV    ON          accept CMS minimum command
CLSET    00104                          abbreviations (truncations).

CLSET    00107    ------
CLSET    00109    * COBOL and PLIOPT invoke IBM Program Products which may
CLSET    00111       be obtained from IBM for a license fee.
```

```
CLSET    00113              OFF      do not accept CMS minimum command
CLSET    00114                       abbreviations (truncticns).

CLSET    00116    REDTYPE   ON       CMS error messages will be typed in
CLSET    00117                       red, if the terminal is equipped with
CLSET    00117                       the appropriate terminal feature and a
CLSET    00118                       two-color ribbon.

CLSET    00120              OFF      suppress red printing of error
CLSET    00121                       messages.

CLSET    00123    IMPEX     ON       EXEC files will be considered commands,
CLSET    00123                       and can be invoked simply by entering
CLSET    00124                       the filename of the EXEC file. This is
CLSET    00125                       the initial setting.

CLSET    00127              OFF      EXEC files are not considered commands.
CLSET    00128                       To execute an EXEC file, the "EXEC"
CLSET    00129                       command name must be issued.

CLSET    00131    IMPCP     ON       Command names that CMS does not
CLSET    00131                       recognize will be passed to CP; that
CLSET    00132                       is, unknown commands will be assumed to
CLSET    00133                       be CP commands.

CLSET    00135              OFF      commands that CMS does not recognize
CLSET    00136                       generate an error message at the user's
CLSET    00137                       terminal.

CLSET    00141    PROTECT   ON       The CMS nucleus is protected against
CLSET    00141                       writing being done within its storage
CLSET    00142                       area.

CLSET    00144              OFF      The storage area containing the CMS
CLSET    00146                       nucleus may be written on.
```

CLSORT 00005 **SORT**

CLSORT 00007 The SORT command reads fixed length records from a CMS input
CLSORT 00008 file, arranges them in ascending EBCDIC order according to
CLSORT 00009 user-specified sort fields, and creates a new file
CLSORT 00010 containing the sorted records. The input and output files
CLSORT 00011 must not have the same file identifiers, since SORT cannot
CLSORT 00011 write the sorted output back into the space occupied by the
CLSORT 00012 file. If a file with the same name as the output file
CLSORT 00014 already exists, the old file is erased.

CLSORT 00022 ┌──────────────────────────────┐
CLSORT 00023 | SORT | fn1 ft1 fm1 fn2 ft2 fm2 |
CLSORT 00024 └──────────────────────────────┘

CLSORT 00030 fn1 specifies the filename of the file containing the
CLSORT 00031 records to be sorted.

CLSORT 00034 ft1 specifies the filetype of the file containing the
CLSORT 00035 records to be sorted.

CLSORT 00038 fm1 specifies the filemode of the file containing the
CLSORT 00039 records to be sorted.

CLSORT 00042 fn2 specifies the filename of the new output file which
CLSORT 00043 will contain the sorted records

CLSORT 00046 ft2 specifies the filetype of the new output file which
CLSORT 00047 will contain the sorted records.

CLSORT 00050 fm2 specifies the filemode of the new output file which
CLSORT 00051 will contain the sorted records.

CLSORT 00057 Entering Sort Control Fields: After the SORT command is
CLSORT 00058 entered, CMS responds with the following message on the
CLSORT 00059 terminal typewriter:

CLSORT 00062 DMSSRT604R ENTER SORT FIELDS:

CLSORT 00064 The user should respond by typing one or more pairs of
CLSORT 00065 numbers of the form "xx yy" separated by one or more blanks.
CLSORT 00066 Each xx is the starting character position of a sort field
CLSORT 00067 within each input record and yy is the ending character
CLSORT 00068 position. The leftmost pair of numbers denotes the major
CLSORT 00069 sort field. The number of sort fields is limited to the
CLSORT 00070 number of fields the user can key in on one line. The
CLSORT 00071 records can be sorted on up to a total of 253 positions.

CLSORT 00075 Main Storage Requirements for Sort: The sorting operation
CLSORT 00076 takes place with two passes of the input file. Pass one
CLSORT 00076 creates an ordered pointer table in the user's virtual
CLSORT 00077 storage. Pass two uses the pointer table to read the input
CLSORT 00078 file in a random manner and write the output file.
CLSORT 00079 Therefore, the size of storage and the size and number of
CLSORT 00080 sort fields are the limiting factors in determining the
CLSORT 00081 number of records that can be sorted at any one time. An
CLSORT 00082 estimate of the maximum number of records that can be sorted
CLSORT 00083 can be made by using the following formula:

CLSORT 00087
CLSORT 00088 $$NR = \frac{VMSIZE - 132K}{14 + NC}$$

CLSORT 00092 where NR is the estimated maximum number of input records;
CLSORT 00093 NC is the total number of characters in the defined sort
CLSORT 00094 fields; VMSIZE is the storage size of the user's virtual
CLSORT 00096 machine; and K represents 1024 bytes (132K is the size of
CLSORT 00098 the resident CMS nucleus). For example:

CLSORT 00101 ┌──┐
CLSORT 00102 | SORT | NAME ADDRESS A1 SORTEDNA ADDRESS B1 |
CLSORT 00103 └──┘

CLSORT 00105 DMSSRT604R ENTER SORT FIELDS:

CLSORT 00107 1 10 25 28

CLSORT 00113 The records in the file NAME ADDRESS are sorted on positions
CLSORT 00114 1-10 and 25-28. The sorted output is written into the newly
CLSORT 00115 created file SORTEDNA ADDRESS. If the user has a 320K
CLSORT 00117 virtual machine he can sort a maximum of 6875 records.

CLSORT 00121
CLSORT 00122 $$NR = \frac{VMSIZE-132K}{14 + NC} = \frac{320K-132K}{14 + 14} = \frac{188K}{28} = \frac{192,512}{28} = 6875$$

CLSORT 00127 Responses

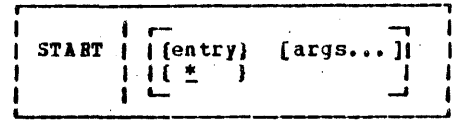
CLSORT 00130 DMSSRT604R ENTER SORT FIELDS:

CLSORT 00133 The user is requested to enter SORT control fields. He
CLSORT 00134 should enter them in the form described above.

CLSTART 00006 START

CLSTART 00008 START begins execution of programs that have been previously
CLSTART 00009 loaded, and passes the address of a string of user arguments
CLSTART 00010 to that program.

CLSTART 00021
CLSTART 00022
CLSTART 00023
CLSTART 00024
CLSTART 00025
CLSTART 00026



CLSTART 00031 **entry** specifies the name of a control section or entry
CLSTART 00032 point to which control is passed at execution
CLSTART 00033 time. Entry must be a control section name or an
CLSTART 00033 entry point name. It may be a filename only if
CLSTART 00034 the filename is identical to a control section
CLSTART 00035 name or an entry point name.

CLSTART 00037 * specifies that control is to be passed to the
CLSTART 00038 default entry point. If nothing is coded in this
CLSTART 00039 field this is the default. The default entry
CLSTART 00039 point is either the address specified in the
CLSTART 00040 operand field, of the first END card containing
CLSTART 00040 a non-blank operand field, or the beginning of
CLSTART 00041 the first file loaded if all END cards in the
CLSTART 00041 TEXT files contain blank operand fields. The
CLSTART 00042 default entry point can be changed by issuing
CLSTART 00042 the INCLUDE-RESET option when loading additional
CLSTART 00043 files.

CLSTART 00045 **args...** specify information to be passed to the started
CLSTART 00046 program. If user arguments are specified, entry
CLSTART 00046 or * must be specified; otherwise, the first
CLSTART 00046 argument is taken as the entry point. Arguments
CLSTART 00047 are passed to the program via general register
CLSTART 00047 1. The entry operand and any arguments are set
CLSTART 00048 up as a string of doublewords, one argument per
CLSTART 00048 doubleword, and the address of the list is
CLSTART 00049 placed in general register 1. The arguments are
CLSTART 00049 obtained by using displacements of 8, 16, 24,
CLSTART 00050 and so forth, from the address contained in
CLSTART 00050 register 1, when execution of the specified
CLSTART 00051 program begins.

CLSTART 00054 Note: Any undefined names or references specified in the
CLSTART 00055 files loaded into storage are defined to location zero.
CLSTART 00055 Thus, if there is a call or branch to a subroutine from a
CLSTART 00056 main program and the subroutine has never been loaded, the
CLSTART 00056 call, or branch transfers control to location zero of the
CLSTART 00057 users virtual machine at execution time.

```
CLSTART  00060    Responses
CLSTART  00063    "EXECUTION BEGINS..."
CLSTART  00065    is typed when the designated entry point is validated.
```

CLSTATE 00006 **STATE**

CLSTATE 00009 The STATE command verifies the existence of a CMS file.

CLSTATE 00018
CLSTATE 00019 | STATE | fn ft [fm] |
CLSTATE 00020

CLSTATE 00025 fn specifies the filename of the file whose existence is
CLSTATE 00026 to be verified. This field must be specified.

CLSTATE 00028 ft specifies the filetype of the file whose existence is
CLSTATE 00029 to be verified. This field must be specified.

CLSTATE 00031 fm specifies the filemode of the file whose existence is
CLSTATE 00032 to be verified. If this field is omitted, all user
CLSTATE 00033 disks will be searched.

CLSTATE 00035 Note: If * is specified for fn, ft and/or fm the first file
CLSTATE 00037 found satisfying the rest of the fileid is used.

CLSTATE 00039 **Responses**

CLSTATE 00040 When STATE is issued for an existing file, the command
CLSTATE 00041 returns with the READY message; otherwise, an error message
CLSVCTRA 00001 is returned.

CLSVCTRA 00004 **SVCTRACE**

CLSVCTRA 00006 The SVCTRACE command is used to trace and record information
CLSVCTRA 00007 about Supervisor Calls which occur in the system running in
CLSVCTRA 00008 the user's virtual machine.

CLSVCTRA 00009 The information recorded will include the main storage
CLSVCTRA 00011 location of the calling SVC instruction and the name of the
CLSVCTRA 00012 called program or routine; the normal and error return
CLSVCTRA 00013 addresses; the contents of the general and floating-point
CLSVCTRA 00014 registers before branching to the SVC-called program and
CLSVCTRA 00014 after returning from it, and 16 words of the parameter list
CLSVCTRA 00016 which existed when the SVC was issued.

CLSVCTRA 00017 To terminate tracing previously established by the SVCTRACE
CLSVCTRA 00018 command, the user may issue one of the CMS commands, HO or
CLSVCTRA 00019 SVCTRACE OFF. Both SVCTRACE OFF and HO cause all trace
CLSVCTRA 00020 information recorded up to the point they are issued to be
CLSVCTRA 00021 printed on the virtual spooled printer. SVCTRACE OFF can be
CLSVCTRA 00021 issued only when the keyboard is unlocked to accept input to
CLSVCTRA 00022 the CMS command environment. To terminate tracing at any
CLSVCTRA 00023 other point in system processing, HO must be issued. If a
CLSVCTRA 00024 user issues the CMS command HX or logs out from the control
CLSVCTRA 00024 program before termination of tracing set by SVCTRACE, the
CLSVCTRA 00025 switches are cleared automatically and all recorded trace
CLSVCTRA 00026 information is printed on the virtual spooled printer.

CLSVCTRA 00036
CLSVCTRA 00037 | SVCTRACE | {ON } |
CLSVCTRA 00038 | SVCT | {OFF} |
CLSVCTRA 00039

CLSVCTRA 00045 ON specifies that tracing of all SVC instructions
CLSVCTRA 00047 issued within CMS is to commence.

CLSVCTRA 00050 OFF specifies that tracing is to be discontinued.

CLSVCTRA 00052 **Note**: The printer trace output will consist of the
CLSVCTRA 00054 following:

CLSVCTRA 00058 • the contents of the general registers both before the
CLSVCTRA 00058 SVC-called program is given control and after a return
CLSVCTRA 00059 from that program.

CLSVCTRA 00061 • the contents of the general registers when the SVC
CLSVCTRA 00063 handling routine is finished with processing.

CLSVCTRA 00065 • the contents of the floating-point registers before the
CLSVCTRA 00066 SVC-called program is given control and after a return
CLSVCTRA 00067 from that program.

- the contents of the floating-point registers when the SVC handling routine is finished processing.

- the parameter list passed to the SVC.

Responses

The format of the first line of trace output is as follows:

```
┌   ┐
| - |
| + |  N/D = xxx/dd name FROM loc OLDPSW = psw1 GCPSW = psw2 RC = rc
| * |
└   ┘
```

–	information before processing of the SVC trace.
+	information after processing of the SVC.
*	complete before and after tracing.
xxx	SVC number.
dd	nesting level of SVC call.
name	name of the macro or routine being called.
loc	program location from which the SVC was issued.
psw1	PSW at time of SVC invocation.
psw2	before–PSW with which the SVC is invoked. after–PSW which returns control to the user.
rc	return code passed from the SVC handling routine via general register 15.

CLSYNO 00004 **SYNONYM**

CLSYNO 00006 The SYNONYM command allows the user to specify his own
CLSYNO 00007 command names to be used with, or in place of the standard
CLSYNO 00008 CMS command names.

CLSYNO 00009 User-defined names may be used either instead of, or in
CLSYNO 00010 conjunction with the standard CMS system abbreviations—that
CLSYNO 00010 is, the synonym command permits the user to modify the
CLSYNO 00011 command names to make them acceptable in his own
CLSYNO 00012 environment.

CLSYNO 00021
CLSYNO 00022 | SYNONYM | [[fn [ft [fm]]] [(options...)]] |
CLSYNO 00023 | SYN | |
CLSYNO 00024 | | options: [STD\|NOSTD] |
CLSYNO 00025 | | [CLEAR] |
CLSYNO 00026

CLSYNO 00031 fn specifies the filename of the user file which
CLSYNO 00033 contains the user-defined synonyms.

CLSYNO 00037 ft specifies the filetype of the user file which
CLSYNO 00038 contains the user-defined synonyms. The
CLSYNO 00039 filetype must be SYNONYM; if omitted, SYNONYM is
CLSYNO 00040 assumed.

CLSYNO 00043 fm specifies the filemode of the user file which
CLSYNO 00044 contains user-define synonyms. If omitted, all
CLSYNO 00045 user disks are searched for the file.

CLSYNO 00047 **Note:** If no file identifier is specified, no user-defined
CLSYNO 00048 synonyms are established and CMS abbreviations are used in
CLSYNO 00049 the manner defined by the options.

CLSYNO 00052 **Options**

CLSYNO 00054 STD specifies that standard CMS abbreviations are to
CLSYNO 00055 be used.

CLSYNO 00060 NOSTD standard CMS abbreviations are not to be used.

CLSYNO 00063 CLEAR removes any synonym table set by a previously
CLSYNO 00064 entered SYNONYM command.

CLSYNO 00068 Note: The user's SYNONYM file consists of 80-byte
CLSYNO 00069 fixed-length records in free-form format with columns 73-80
CLSYNO 00070 ignored. Its filetype must be SYNONYM. The format for each
CLSYNO 00071 record is

CLSYNO 00073 ┌──────────────────────────────────────┐
CLSYNO 00074 | system-command| user-synonym| count |
CLSYNO 00075 | | | |
CLSYNO 00076 └──────────────────────────────────────┘

CLSYNO 00081 where count is the minimum number of characters that must be
CLSYNO 00082 entered for the synonym to be accepted by CMS. If omitted,
CLSYNO 00082 the entire synonym must be entered (see the example below).
CLSYNO 00083 A table is built from the contents of this file to use for
CLSYNO 00084 command synonyms. The user may have several SYNONYM files
CLSYNO 00085 but only one may be active at any one time. For example, if
CLSYNO 00086 the synonym file contains:

CLSYNO 00088 MOVEFILE MVIT

CLSYNO 00089 The synonym MVIT may be entered as a command name to execute
CLSYNO 00090 the MOVEFILE command. It may not be abbreviated since no
CLSYNO 00091 count is specified.

CLSYNO 00094 ACCESS GETDISK 3

CLSYNO 00096 The synonyms GET, GETD, GETDI, GETDIS, or GETDISK may be
CLSYNO 00097 entered as the command name instead of ACCESS.

CLSYNO 00100 Responses

CLSYNO 00102 DMSSYN711I NO SYSTEM SYNONYMS IN EFFECT

CLSYNO 00104 No synonyms have been defined for the system.

CLSYNO 00106 DMSSYN712I NO SYNONYMS (DMSINA NOT IN NUCLEUS)

CLSYNO 00107 The system routine which handles SYNONYM processing is not
CLTAPE 00001 in the system.

TAPE

The TAPE command dumps CMS-formatted files from disk to tape, loads previously dumped files from tape to disk, and performs various control operations on a specified tape drive. TAPE is used solely for CMS files; therefore, the files on tape are of a format unique to CMS. The TAPE command will not process multi-volume files. Disk files to be dumped may contain either fixed or variable length records.

```
┌──────┬─────────────────────────────────────────────────────────────────────────────┐
|      |                              ┌  ┐                                            |
| TAPE | {   DUMP   {fn}  {ft}  |fm|                        *     }                   |
|      | {          {* }  {* }  |* |   [(optionA optionB)]        }                   |
|      | {                      └  ┘                              }                   |
|      | { {                                  }                   }                   |
|      | { { LOAD   |{fn}{ft}|fm||            }                   }                   |
|      | { {        |{* }{* }|A ||            }                   }                   |
|      | { {        └       └  ┘┘            }          *        }    *              |
|      | { {                                  } [(optionB optionC)]  }  (optionD)    |
|      | { {        ┌          ┐              }                   }                   |
|      | { { SCAN   |{fn}{ft}|               }                   }                   |
|      | { {        |{* }{* }|               }                   }                   |
|      | { {        └          ┘              }                   }                   |
|      | { { SKIP   {fn}  {ft}                }                   }                   |
|      | { {        {* }  {* }                }                   }                   |
|      | {                                                        }                   |
|      | {   MODESET                                              }                   |
|      | {             ┌ ┐                                        }                   |
|      | {   tapcmd    |n|                                        }                   |
|      | {             |1|                                        }                   |
|      | {             └ ┘                                        }                   |
|      |                                                                              |
|      | optionA:    [WTM|NOWTM]                                                      |
|      |                                                                              |
|      | optionB:    [NOPRINT|PRINT|TERM|DISK]                                        |
|      |                                                                              |
|      | optionC:    [EOF n|EOT|EOF 1]                                                |
|      |                                                                              |
|      | optionD:    [TAPi|ccu|TAP1|181]                                              |
|      |             [7TRACK | 9TRACK]                                                |
|      |             [DEN {200|556|800|1600}]                                         |
|      |             [TRTCH {O|OC|OT|E|ET}]                                           |
|      |                                                                              |
|      | tapcmd:     [BSF|BSR|ERG|FSF|FSR|REW|RUN|WTM]                                |
└──────┴─────────────────────────────────────────────────────────────────────────────┘
```

* These parentheses are not used if option D is selected.

fn specifies the filename of the files upon which the indicated operations are to be performed.

```
CLTAPE  00072   ft          specifies the  filetype of the files  upon which
CLTAPE  00074               the indicated operations are to be performed.

CLTAPE  00076   fm          specifies the  filemode of the files  upon which
CLTAPE  00078               the indicated operations are to be performed.


CLTAPE  00081   Functions

CLTAPE  00085   DUMP        The DUMP function  dumps one or more  disk files
CLTAPE  00087               to tape. The file  identifier must be specified.
CLTAPE  00088               If an asterisk is coded for  fn or ft, all files
CLTAPE  00089               which satisfy the resulting fileid are dumped.

CLTAPE  00091   LOAD        The LOAD function writes tape files to disk.  If
CLTAPE  00093               a fileid  is specified,  only that  one file  is
CLTAPE  00093               loaded. If the option EOF  n is specified and no
CLTAPE  00094               file identification is entered, n tape files are
CLTAPE  00095               written to disk. If an asterisk (*) is specified
CLTAPE  00096               for  fn or  ft,  all files  within  EOF n  which
CLTAPE  00096               satisfy the resulting file identifer are loaded.
CLTAPE  00097               The files are  written to the disk  indicated by
CLTAPE  00098               the mode letter.  The mode number if entered, is
CLTAPE  00099               ignored.

CLTAPE  00102   SCAN        The SCAN function types  to the terminal (unless
CLTAPE  00103               NOPRINT, PRINT  or DISK is specified)  the names
CLTAPE  00104               of the files on tape.  If DISK is specified the
CLTAPE  00105               list of  file identifiers is  written to  a file
CLTAPE  00106               named  TAPE MAP.  If a  fileid is  specified,
CLTAPE  00107               scanning stops upon encountering  that file.  If
CLTAPE  00107               not specified, scanning occurs over n tape marks
CLTAPE  00108               as specified by the option EOF n.

CLTAPE  00111   SKIP        The  SKIP  function  positions  the  tape  at  a
CLTAPE  00111               specified  point, depending  upon other  options
CLTAPE  00112               and parameters.  If fileid is entered,  the tape
CLTAPE  00113               is positioned after the specified file; if EOF n
CLTAPE  00113               is entered, the tape is  positioned after n tape
CLTAPE  00114               marks.

CLTAPE  00117   MODESET     specifies that the values set by the DEN, TRACK
CLTAPE  00120   MODE        and TRTCH options described  below are to remain
CLTAPE  00121               in effect for the specified  tape until they are
CLTAPE  00122               changed in a subsequent TAPE command.

CLTAPE  00125   tapcmd n    specifies a tape control function to be executed
CLTAPE  00128           1   n times (default is 1 if n is not specified):

CLTAPE  00130               BSF         backspace n tape marks
CLTAPE  00131               BSR         backspace n tape records
CLTAPE  00132               ERG         erase gap
```

```
CLTAPE  00133                FSF       forward space n tape marks
CLTAPE  00134                FSR       forward space n tape records
CLTAPE  00135                REW       rewind tape to load point
CLTAPE  00136                RUN       rewind tape and unload
CLTAPE  00137                WTM       write n tape marks


CLTAPE  00145  Options

CLTAPE  00147  WTM            specifies that a tape mark is to be written on
CLTAPE  00148                 the tape after each file dumped.

CLTAPE  00150  NOWTM          specifies that no tape mark is to be written on
CLTAPE  00152                 the tape upon completion of the DUMP function.

CLTAPE  00155  NOPRINT        speicifes that no spool print record is to be
CLTAPE  00157  NOPR           built summarizing the files operated on during
CLTAPE  00158                 the DUMP function.

CLTAPE  00161  TERM           specifies that a list of files operated on is to
CLTAPE  00164  T              be typed out at the user's terminal typewriter.

CLTAPE  00167  PRINT          specifies that the list of files operated on is
CLTAPE  00170  PR             to be spooled to the printer.

CLTAPE  00172  DISK           specifies that a disk file is to be created
CLTAPE  00173                 containing the list of files operated on by the
CLTAPE  00174                 DUMP operation. The disk file has a file
CLTAPE  00175                 identifier of TAPE MAP.

CLTAPE  00177  EOT            indicates that tape reading should continue
CLTAPE  00179                 until end-of-tape indication is received.

CLTAPE  00181  EOF n          indicates that tape reading should continue
CLTAPE  00182                 through a maximum of n tape marks. Default is
CLTAPE  00183                 EOF 1.

CLTAPE  00185  TAPi/ccu       specifies the symbolic tape identification or
CLTAPE  00186                 the actual device address of the tape to be read
CLTAPE  00187                 from or written to. The default is TAP1/181.
CLTAPE  00187                 The unit specified by ccu must have previously
CLTAPE  00188                 been attached to the user's system before any
CLTAPE  00189                 tape I/O operation can be attempted.

CLTAPE  00191  7TRACK         specifies a 7 track tape. Odd parity, data
CLTAPE  00192                 convert on and translate off are assumed unless
CLTAPE  00193                 TRTCH is specified.

CLTAPE  00196  9TRACK         specifies a 9 track tape.

CLTAPE  00199  DEN nnn        specifies the tape density - 200, 556, 800, or
CLTAPE  00199                 1600 bytes per inch. If 200 or 556 is
```

CLTAPE 00200 specified, 7TRACK is assumed. If 1600 is
CLTAPE 00201 specified, 9TRACK is assumed; if 800 is
CLTAPE 00201 specified, 9TRACK is assumed unless 7TRACK is
CLTAPE 00202 specified.

CLTAPE 00205 **TRTCH xx** specifies the tape recording technique for 7
CLTAPE 00205 track tape. If TRTCH is specified, 7TRACK is
CLTAPE 00207 assumed. One of the following must be specified
CLTAPE 00208 as xx:

CLTAPE 00214 0 - odd parity, data convert off, translate off
CLTAPE 00217 OC - odd parity, data convert on, translate off
CLTAPE 00222 OT - odd parity, data convert off, translate off
CLTAPE 00224 E - even parity, data convert off, translate
CLTAPE 00225 off
CLTAPE 00226 ET - even parity, data convert off, translate
CLTAPE 00227 off

CLTAPE 00231 Format of Tape Created by TAPE DUMP Command:

CLTAPE 00232 Tape records written by TAPE DUMP are 805 bytes long. The
CLTAPE 00233 first character is a binary 2 (X'02'), followed by the
CLTAPE 00234 characters CMS and an EBCDIC blank (X'40'), followed by 800
CLTAPE 00235 bytes of file data packed without regard for logical record
CLTAPE 00236 length. In the final record, the character N replaces the
CLTAPE 00236 blank after CMS, and the data area contains directory
CLTAPE 00237 information.

CLTAPE 00239 TAPCMD Restrictions:

CLTAPE 00240 The REW and RUN functions indicate completion before the
CLTAPE 00241 physical operation is completed. Thus a subsequent
CLTAPE 00242 operation to the same physical device may encounter a
CLTAPE 00243 "device busy" situation.

CLTAPE 00245 Responses

CLTAPE 00248 DMSTPE701I NULL FILE

CLTAPE 00250 A final record was encountered and no prior records had been
CLTAPE 00251 read in a TAPE LOAD operation. No file is created on disk.

CLTAPE 00252 If the TERM option is specified, the following is typed to
CLTAPE 00254 the terminal depending on the operation specified:

CLTAPE 00256 LOADING.....
CLTAPE 00258 fn ft fm
CLTAPE 00261 .

```
CLTAPE   00263     .
CLTAPE   00265     .

CLTAPE   00267     SKIPPING.....
CLTAPE   00269     fn ft fm

CLTAPE   00271     DUMPING.....
CLTAPE   00273     fn ft fm
CLTAPE   00275     .
CLTAPE   00277     .
CLTAPE   00279     .

CLTAPE   00281     SCANNING.....
CLTAPE   00283     fn ft fm
CLTAPE   00285     .
CLTAPE   00287     .
CLTAPE   00289     .

CLTAPE   00290     When a  tape mark is encountered  the following is  typed to
CLTAPE   00292     the terminal if the TERM option is specified:

CLTAPPDS 00001          END-OF-FILE OR END-OF-TAPE
```

CLTAPPDS 00004 **TAPPDS**

CLTAPPDS 00006 The TAPPDS command is used to create CMS disk files from
CLTAPPDS 00007 tapes in unblocked card-image format. The tape may be
CLTAPPDS 00008 unlabeled or contain OS standard labels.

CLTAPPDS 00018
CLTAPPDS 00019 TAPPDS [fn [ft [fm]]] [(options...)]
CLTAPPDS 00020 options:
CLTAPPDS 00021 PDS
CLTAPPDS 00022 NOPDS
CLTAPPDS 00023
CLTAPPDS 00024
CLTAPPDS 00025 COL1
CLTAPPDS 00026 NOCOL1
CLTAPPDS 00027
CLTAPPDS 00028 TAPn
CLTAPPDS 00029 TAP1
CLTAPPDS 00030
CLTAPPDS 00031
CLTAPPDS 00032 END
CLTAPPDS 00033 NOEND
CLTAPPDS 00034
CLTAPPDS 00035
CLTAPPDS 00036 MAXTEN
CLTAPPDS 00037 NOMAXTEN
CLTAPPDS 00038

CLTAPPDS 00045 **fn** specifies the filename of the disk file to be
CLTAPPDS 00046 created. This field has meaning only if the
CLTAPPDS 00047 NOPDS option is selected (that is, the tape
CLTAPPDS 00047 does not contain members of a partitioned
CLTAPPDS 00048 data set). If the tape does contain members
CLTAPPDS 00048 of a partitioned data set (PDS), an asterisk
CLTAPPDS 00049 must be specified; one file is created for
CLTAPPDS 00049 each member with filename the same as the
CLTAPPDS 00050 member name. If NOPDS is specified, the
CLTAPPDS 00051 default filename is TAPPDS. This is assumed
CLTAPPDS 00052 if the filename is omitted or coded as "*".

CLTAPPDS 00055 **ft** specifies the filetype of the newly created
CLTAPPDS 00056 files. The default filetype is CMSUT1, if ft
CLTAPPDS 00057 is omitted or specified as *.

CLTAPPDS 00059 **fm** specifies the disk on which to place the new
CLTAPPDS 00060 files. The default filemode is A1 if this
CLTAPPDS 00060 field is omitted or specified as an
CLTAPPDS 00061 asterisk(*).

CLTAPPDS 00065 **Options**

CLTAPPDS 00066 If conflicting options are specified, the last one entered
CLTAPPDS 00067 is used.

CLTAPPDS 00069 **PDS** Indicates that the tape contains members of
CLTAPPDS 00070 an OS partitioned data set, each preceded by
CLTAPPDS 00070 a "MEMBER NAME=name" card. The tape must
CLTAPPDS 00071 have been created by the OS IEBPTPCH utility
CLTAPPDS 00072 if this option is specified.

CLTAPPDS 00076 NOPDS Indicates that the tape contains one file.

CLTAPPDS 00079 COL1 Indicates that column 1 contains data. Data
CLTAPPDS 00081 is taken from columns 1-80

CLTAPPDS 00084 **NOCOL1** Indicates that column 1 contains control
CLTAPPDS 00085 character information. Data is taken from
CLTAPPDS 00085 columns 2-81. This is the format produced by
CLTAPPDS 00086 the OS IEBPTPCH utility.

CLTAPPDS 00090 TAPn n indicates the tape unit number. TAP1 is
CLTAPPDS 00091 **TAP1** the default tape unit number, which
CLTAPPDS 00091 corresponds to the virtual address 181.
CLTAPPDS 00092 There are four possible values of n: TAP1 to
CLTAPPDS 00092 TAP4, indicating virtual tape drives 181 to
CLTAPPDS 00093 184.

CLTAPPDS 00096 END specifies that an END card when encountered
CLTAPPDS 00097 means that the entire current member has been
CLTAPPDS 00098 read.

CLTAPPDS 00101 **NOEND** specifies that END cards are not to be
CLTAPPDS 00102 treated as member delimiters, but are to be
CLTAPPDS 00103 processed as text.

CLTAPPDS 00106 MAXTEN Indicates that up to ten members are to be
CLTAPPDS 00107 read. This is valid only if the PDS option
CLTAPPDS 00108 is selected.

CLTAPPDS 00111 **NOMAXTEN** Indicates that any number of members may be
CLTAPPDS 00112 read.

CLTAPPDS 00116 **Responses**

CLTAPPDS 00121 DMSTPD703I FILE 'fn ft [fm]' COPIED

CLTAPPDS 00123 The named file has been copied to disk

```
CLTAPPDS 00125     DMSTPD707I  TEN FILES COPIED

CLTAPPDS 00127     MAXTEN has been specified and ten members have been copied.

CLTAPPDS 00129     If  the tape  being read  contains standard  CS labels,  the
CLTXTLIB 00001     labels are typed at the terminal.
```

CLTXTLIB 00003 **TXTLIB**

CLTXTLIB 00005 The TXTLIB command is used to perform maintenance on CMS
CLTXTLIB 00006 text libraries. A text library is one which is to be
CLTXTLIB 00006 searched for missing subroutines in LOAD and INCLUDE
CLTXTLIB 00007 commands.

CLTXTLIB 00008 A text library is a file that has a filetype of TXTLIB and
CLTXTLIB 00009 contains a dictionary and one or more relocatable object
CLTXTLIB 00010 programs which were obtained from CMS files having a
CLTXTLIB 00011 filetype of TEXT.

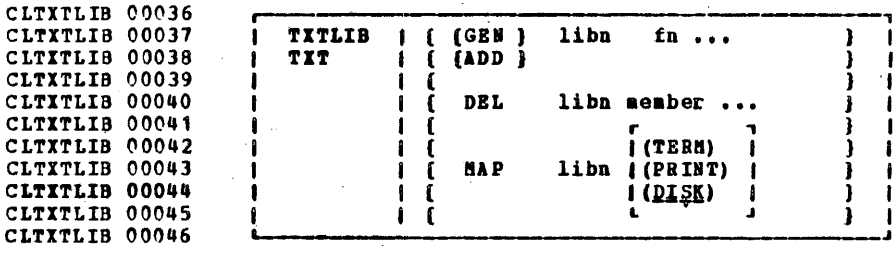
CLTXTLIB 00013 The TXTLIB command performs the following functions:

CLTXTLIB 00016 1. Generates a text library.

CLTXTLIB 00018 2. Adds to an existing text library.

CLTXTLIB 00020 3. Deletes from an existing text library.

CLTXTLIB 00023 4. Lists the names and aliases or entry points and control
CLTXTLIB 00024 section names and the location of the TEXT files
CLTXTLIB 00025 included in the text library.



CLTXTLIB 00051 **GEN** a text library with a user-specified filename of
CLTXTLIB 00052 libn and a filetype of TXTLIB is to be created
CLTXTLIB 00052 from the TEXT files specified by fn1 fn2... If
CLTXTLIB 00053 a file exists with the identifier "libn TXTLIB",
CLTXTLIB 00054 it is erased and a new one is created.

CLTXTLIB 00057 **ADD** the contents of the files specified by fn1
CLTXTLIB 00058 fn2... are to be appended to the end of the
CLTXTLIB 00058 existing library with the file identifier "libn
CLTXTLIB 00058 TXTLIB". No checking for duplicate names,
CLTXTLIB 00059 aliases, entry points or CSECT names is
CLTXTLIB 00060 performed.

CLTXTLIB 00063 **DEL** the TEXT decks with member names csectn1,
CLTXTLIB 00064 csectn2... are to be removed from the directory
CLTXTLIB 00064 of the file with the file identifier "libn

```
CLTXTLIB 00064                 TXTLIB".  If two members have the same name as
CLTXTLIB 00065                 one specified by member, only the first one
CLTXTLIB 00065                 encountered is deleted (unless the member name
CLTXTLIB 00067                 is given twice in the argument list).

CLTXTLIB 00068                 Deletes must be performed on the NAME or first
CLTXTLIB 00069                 Section Definition (SD) in the text deck.  A
CLTXTLIB 00069                 delete against an alias name or subsequent entry
CLTXTLIB 00070                 point results in a "not found" message with no
CLTXTLIB 00070                 change to the member.  Delete removes the member
CLTXTLIB 00071                 and all references to it.

CLTXTLIB 00072                 The file is automatically compressed so that
CLTXTLIB 00072                 space occupied by the deleted CSECTs may be
CLTXTLIB 00073                 reused.

CLTXTLIB 00076     MAP         the MAP form of the TXTLIB command generates the
CLTXTLIB 00077                 file "libn MAP" on the user's primary disk.  If
CLTXTLIB 00077                 a file already exists with the same identifier,
CLTXTLIB 00078                 it is erased and the new file created.  The
CLTXTLIB 00079                 "libn MAP" file contains the same information as
CLTXTLIB 00079                 that in the dictionary of the specified text
CLTXTLIB 00080                 library and is in the format of a list of entry
CLTXTLIB 00080                 points and control section names that reside in
CLTXTLIB 00081                 the text library, and their location or index in
CLTXTLIB 00082                 the file.  The option list on the command line
CLTXTLIB 00082                 is examined to determine if the MAP is to be
CLTXTLIB 00083                 directed to the terminal (TERM), or the printer
CLTXTLIB 00084                 (PRINT), or is to remain on disk (DISK).

CLTXTLIB 00085                 The MAP form of TXTLIB types out a statement
CLTXTLIB 00086                 indicating the total number of entry points and
CLTXTLIB 00086                 control section names that currently exist in
CLTXTLIB 00087                 the TXTLIB file.

CLTXTLIB 00090     libn        is the name of the text library to be generated,
CLTXTLIB 00091                 added to, printed, or listed.  The filetype of
CLTXTLIB 00092                 libn must be TXTLIB.

CLTXTLIB 00095     fn          specify the file(s) to be used in either
CLTXTLIB 00096                 generating or adding to a TXTLIB file.  Their
CLTXTLIB 00097                 filetype must be TEXT.

CLTXTLIB 00098                 Each TEXT file that is to be included in the
CLTXTLIB 00099                 TXTLIB file must consist of one or more control
CLTXTLIB 00099                 sections with an END card image following each
CLTXTLIB 00100                 control section.

CLTXTLIB 00104     member      specifies the names of the CSECTs to be deleted
CLTXTLIB 00105                 from a TXTLIB file.  Entries which are not
CLTXTLIB 00105                 control section definitions or names specified
CLTXTLIB 00106                 on a loader NAME control card result in a "not
CLTXTLIB 00107                 found" message and no change to the library.
```

CLTXTLIB 00112 **Notes**

CLTXTLIB 00114 1. The total number of members the TXTLIB file cannot
CLTXTLIB 00115 exceed 1000. When this number is reached, an error
CLTXTLIB 00115 message is typed out. The text library created
CLTXTLIB 00116 includes all the text files entered up to (but not
CLTXTLIB 00117 including) the one that caused the overflow.

CLTXTLIB 00120 2. OS Linkage Editor ENTRY, ALIAS, and NAME control cards
CLTXTLIB 00121 are accepted. If a NAME card is detected, only ALIAS
CLTXTLIB 00122 and NAME 'names' will be included in the dictionary for
CLTXTLIB 00122 that text deck. Deletes must be performed on the NAME
CLTXTLIB 00123 'name'. The total number of ALIAS names cannot exceed
CLTXTLIB 00124 sixteen names per text deck.

CLTXTLIB 00127 **Responses**

CLTXTLIB 00129 xxx ENTRIES IN LIBRARY

CLTXTLIB 00130 When TXTLIB is issued, the contents of the dictionary of the
CLTXTLIB 00131 specified text library are typed out. The number of entries
CLTXTLIB 00132 in the text library (xxx) is typed out when TXTLIB MAP is
CLTYPE 00001 issued.

CLTYPE 00005 **TYPE**

CLTYPE 00007 The TYPE command causes all or part of a specified file to
CLTYPE 00008 be typed at the terminal in either EBCDIC or the hexadecimal
CLTYPE 00009 representation of the EBCDIC code.

CLTYPE 00020
CLTYPE 00021
CLTYPE 00022 TYPE fn ft [fm] [(rec1) [recN]] [(options)]
CLTYPE 00023 [*] [*]
CLTYPE 00024
CLTYPE 00025 options:
CLTYPE 00026 COL {xxx} {yyy }
CLTYPE 00027 { } -{ }
CLTYPE 00028 { 1 } {lrecl}
CLTYPE 00029
CLTYPE 00030
CLTYPE 00031 HEX
CLTYPE 00032
CLTYPE 00033
CLTYPE 00034
CLTYPE 00035 { * }
CLTYPE 00036 MEMBER { }
CLTYPE 00037 {NAME }
CLTYPE 00038
CLTYPE 00039

CLTYPE 00045 **fn** specifies the filename of the file to be typed.
CLTYPE 00046 This field must be specified.

CLTYPE 00048 **ft** specifies the filetype of the file to be typed.
CLTYPE 00049 This field must be specified.

CLTYPE 00051 **fm** specifies the filemode of the file to be typed.
CLTYPE 00052 If this field is omitted, the A disk and its
CLTYPE 00053 extensions are searched to locate the file. In
CLTYPE 00054 the case of files with duplicate filename and
CLTYPE 00055 filetype, only the first file found is typed.

CLTYPE 00057 **recl** specifies the record number of the first record
CLTYPE 00058 to be typed. This field must contain no special
CLTYPE 00059 characters. If it is greater than the number of
CLTYPE 00060 records in the file, the file length is assumed.
CLTYPE 00061 If this field is omitted or entered as an
CLTYPE 00062 asterisk, a record number of 1 is assumed.

CLTYPE 00064 **recN** specifies the record number of the last record
CLTYPE 00064 to be typed. This value must be less than 9999
CLTYPE 00065 with no embedded commas. If this field is not
CLTYPE 00066 specified or is entered as an asterisk, typing
CLTYPE 00067 continues until end of file is reached.

CLTYPE 00071 **Options**

CLTYPE 00074 COL xxx-yyy specifies that only certain positions of each
CLTYPE 00074 record in the file are to be typed. xxx
CLTYPE 00075 specifies the beginning position and yyy the
CLTYPE 00075 ending position of the field within the record
CLTYPE 00076 which is to be typed. If a field is not
CLTYPE 00077 specified, the entire record is typed unless the
CLTYPE 00078 filetype is LISTING, in which case the first
CLTYPE 00078 position of each record is not typed, since it
CLTYPE 00080 is assumed to be a carriage control character.

CLTYPE 00083 HEX specifies that the file is to be typed in
CLTYPE 00084 hexadecimal format.

CLTYPE 00087 MEMBER if the file is a library, a MEMBER entry may be
CLTYPE 00090 none specified. If * is entered, all individual
CLTYPE 00093 MEM members of the library are typed. If a name is
CLTYPE 00094 specified, only that particular member is typed.

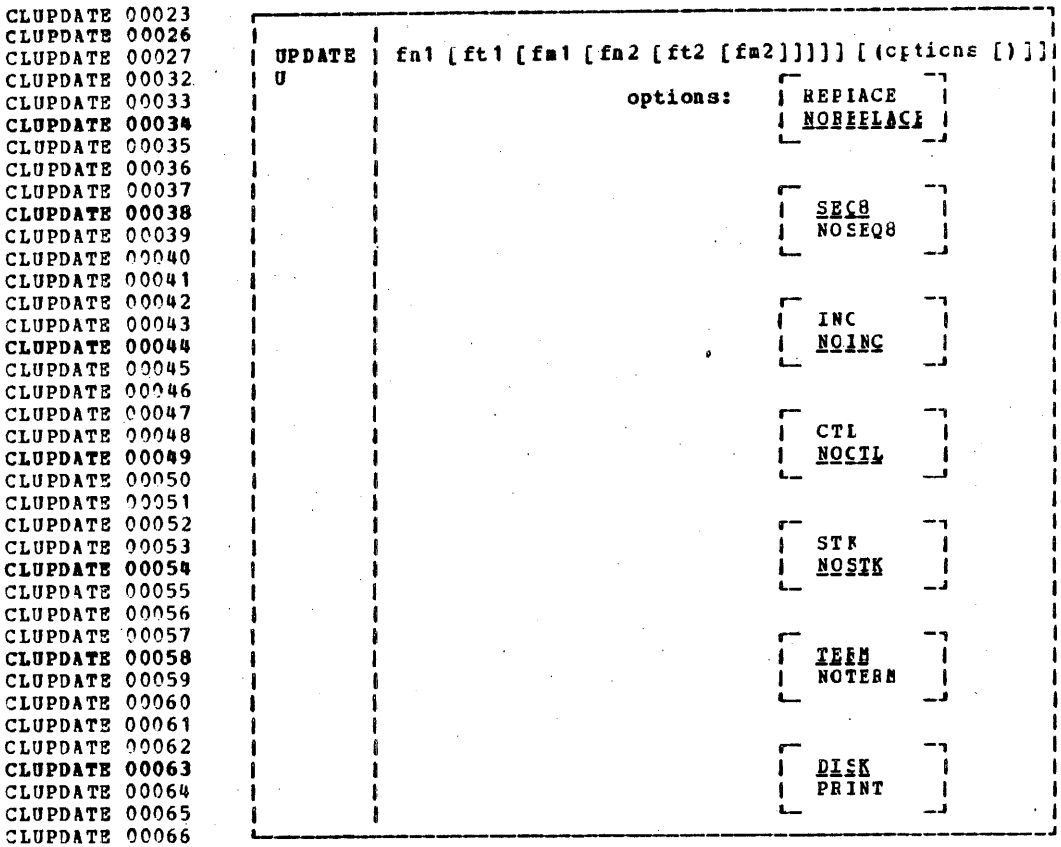
CLTYPE 00098 **Responses**

CLTYPE 00099 The file is typed at the terminal according to the given
CLUPDATE 00101 specifications.

CLUPDATE 00004 **UPDATE**

CLUPDATE 00006 The UPDATE command provides a flexible and generalized
CLUPDATE 00007 facility for maintenance and modification of program source
CLUPDATE 00008 files stored as 80-character card-image files with sequence
CLUPDATE 00009 fields in record positions 73 to 80; UPDATE accepts a source
CLUPDATE 00009 input file and one or more files containing update control
CLUPDATE 00010 cards and updated source records, and creates an updated
CLUPDATE 00011 source output file, an update log file indicating what
CLUPDATE 00011 changes, if any, were made, and an update record file if
CLUPDATE 00012 more than a single update file was applied to the input
CLUPDATE 00013 file.

CLUPDATE 00014 Updates may be applied either permanently (i.e. the updated
CLUPDATE 00015 output file replaces the source input file), or temporarily,
CLUPDATE 00016 in which case the updated output file has the name '$fn',
CLUPDATE 00017 where 'fn' is the file name of the input source file.



fn1 ft1 fm1

specifies the filename, filetype, and filemode of the source input file. If the filemode or filetype is omitted, 'A1' and 'ASSEMBLE' are assumed, respectively.

fn2 ft2 fm2

specifies the filename, filetype, and filemode of the file containing the update control cards and updated source records, or, if the 'CTL' option was specified, specifies the filename, filetype, and filemode of the update control file to be used for a multiple update. The defaults are 'fn1 UPDATE A1' if NOCTL, and 'fn1

CLUPDATE 00089 CNTRL A1' if CTL.

CLUPDATE 00092 **Options**

CLUPDATE 00096 REP specifies that the updated source file is tc
CLUPDATE 00097 replace the source input file.

CLUPDATE 00100 **NOREP** specifies that the old file is to be retained in
CLUPDATE 00101 its original form, and the new file receives a
CLUPDATE 00102 different filename, consisting of a dollar sign
CLUPDATE 00103 ($) plus the first seven characters of the input
CLUPDATE 00104 filename (fn1).

CLUPDATE 00106 **SEQ8** specifies that the entire sequence field
CLUPDATE 00107 (columns 73 through 80) contains an eight-digit
CLUPDATE 00109 sequence number on every record of source input.

CLUPDATE 00113 NOSEQ8 specifies that columns 73-75 contain a three
CLUPDATE 00113 character label field, and that the sequence
CLUPDATE 00115 number is a five-digit value in columns 76-80.

CLUPDATE 00118 INC specifies that the sequence numbers in columns
CLUPDATE 00119 73 to 80 of the updated records inserted from
CLUPDATE 00120 the update file are to be included in the
CLUPDATE 00122 updated source file.

CLUPDATE 00126 **NOINC** specifies that each updated record inserted from
CLUPDATE 00126 the update file is to be identified by asterisks
CLUPDATE 00127 ('********') in columns 73-80 in the updated
CLUPDATE 00128 source file.

CLUPDATE 00132 CTL specifies that fn2, ft2, fm2 describe an update
CLUPDATE 00133 control file for applying multiple update files
CLUPDATE 00133 to the source input file. (see format
CLUPDATE 00135 description in a later section.)
CLUPDATE 00137 Note: The CTL option implies the INC option.

CLUPDATE 00139 **NOCTL** specifies that a single update file is tc be
CLUPDATE 00141 applied to the source input file.

CLUPDATE 00146 STK for use with 'CTL' option only. specifies that
CLUPDATE 00146 the results of the multiple update are to be
CLUPDATE 00147 provided to an external EXEC procedure by way of
CLUPDATE 00149 the CMS console read stack.

CLUPDATE 00153 **NOSTK** specifies that no external communication of the
CLUPDATE 00154 multiple update results is desired.

CLUPDATE 00157 **TERM** specifies that warning messages are to be typed
CLUPDATE 00158 on the terminal whenever a sequence or update
CLUPDATE 00158 control card error is discovered. (Such warning
CLUPDATE 00158 messages will appear in the update log, whether

CLUPDATE 00160 they are typed on the terminal or not.)

CLUPDATE 00163 NOTERM specifies that warning messages are not to be
CLUPDATE 00163 typed on the terminal. However, error messages
CLUPDATE 00164 which abort the entire update procedure will be
CLUPDATE 00165 typed on the terminal anyway.

CLUPDATE 00168 DISK specifies that the update log file is to be
CLUPDATE 00169 placed on disk. This file will have fileid
CLUPDATE 00169 "fname UPDLOG", where "fname" is the filename of
CLUPDATE 00170 the file being updated.

CLUPDATE 00173 PRINT specifies that the update log file is to be
CLUPDATE 00174 printed directly on the virtual printer.

CLUPDATE 00177 Control Cards

CLUPDATE 00179 The UPDATE control cards allow specification of insertions,
CLUPDATE 00180 deletions, and replacement of source records, as well as
CLUPDATE 00181 resequencing of the output file. All UPDATE control cards
CLUPDATE 00181 are identified by the characters './' in columns 1 and 2 of
CLUPDATE 00182 the 80-byte record, followed by one or more blanks and a
CLUPDATE 00183 maximum of six additional, blank-delimited fields. Control
CLUPDATE 00184 card data may not extend beyond column 50. All references to
CLUPDATE 00184 the sequence field of an input record refer to the numeric
CLUPDATE 00185 data in columns 73-80 of the source record, or columns 76-80
CLUPDATE 00186 if NOSEQ8 was specified. Leading zeroes in sequence fields
CLUPDATE 00186 are not required. If no sequence numbers exist on an input
CLUPDATE 00187 file, a preliminary UPDATE with only the './ S' control card
CLUPDATE 00188 can be used to establish file sequencing.

CLUPDATE 00189 Any sequence fields on the update control cards are ignored;
CLUPDATE 00190 if the NOINC option is used, all sequence fields in the
CLUPDATE 00190 update file are ignored, including those on inserted
CLUPDATE 00191 records. If the INC option is used, sequence fields for the
CLUPDATE 00192 inserted records are either generated by UPDATE (if the
CLUPDATE 00192 dollar-sign '$' delimiter is used) or are included intact
CLUPDATE 00194 from the update file (if the dollar-sign '$' is not used).

CLUPDATE 00195 Changes are made sequentially in a single pass through the
CLUPDATE 00196 input and update files; an error condition results if any
CLUPDATE 00196 sequence errors occur in the update control cards, and
CLUPDATE 00197 warnings are issued if an error is detected in the
CLUPDATE 00198 sequencing of the input file. Any source input records with
CLUPDATE 00199 a sequence field of eight blanks will be skipped over by the
CLUPDATE 00200 UPDATE command, without any indication of a sequence error.
CLUPDATE 00200 Such records may be replaced or deleted only if they occur
CLUPDATE 00201 within a range of records, the limits of which are sequenced
CLUPDATE 00202 normally, which is being replaced or deleted in toto; there
CLUPDATE 00202 is no means provided for specifying a sequence target of
CLUPDATE 00203 blanks on an update control card.

CLUPDATE 00205 **Control Card Formats**

CLUPDATE 00207 **1. Resequence the updated source output file.**

CLUPDATE 00214
CLUPDATE 00217
CLUPDATE 00218

```
./  S  [seqstrt  [seqincr  [label]]]
```

CLUPDATE 00227 S — specifies that the updated output file is to be
CLUPDATE 00228 sequenced in columns 73-80 (if SEQ8 is
CLUPDATE 00228 specified), or in columns 76-80 with the 'label'
CLUPDATE 00229 placed in columns 73-75 (if NOSEQ8 is
CLUPDATE 00229 specified). If this card is included in the
CLUPDATE 00230 update file, it must be the first control card.

CLUPDATE 00233 seqstrt — a one- to eight-digit numeric field specifying
CLUPDATE 00234 the first decimal sequence number to be used.
CLUPDATE 00235 Defaults to 1000 for SEQ8, 10 for NOSEQ8.

CLUPDATE 00238 seqincr — a one- to eight-digit numeric field specifying
CLUPDATE 00239 the decimal increment for resequencing the
CLUPDATE 00240 output file. Defaults to 'seqstrt' value.

CLUPDATE 00243 label — a three-character field to be duplicated in
CLUPDATE 00244 columns 73-75 of each source record if NOSEQ8 is
CLUPDATE 00244 specified. Defaults to the first three
CLUPDATE 00245 characters of the input filename (fn1).

CLUPDATE 00247 An error will be indicated if any valid control card
CLUPDATE 00248 precedes the './ S' card in the update file, and the
CLUPDATE 00249 resequence operation will be suppressed.

CLUPDATE 00250 Each source record is resequenced in columns 73-80 as it is
CLUPDATE 00251 written to the output file. Both unchanged records from the
CLUPDATE 00251 input file and records inserted from the update file are
CLUPDATE 00252 resequenced.

CLUPDATE 00255 **2. Insert update records into the output file.**

CLUPDATE 00262
CLUPDATE 00265
CLUPDATE 00266

```
./  I  seqno  [$  [seqstrt  [seqincr]]]
```

CLUPDATE 00275 I — specifies that all cards following the './ I'
CLUPDATE 00276 card, up to the next control card, are to be
CLUPDATE 00277 inserted in the output file.

CLUPDATE 00280 seqno — the sequence field of the source input record
CLUPDATE 00281 following which the insertion is to be made.

CLUPDATE 00285 $ optional delimiter key indicating that the
CLUPDATE 00285 inserted records are to be sequenced
CLUPDATE 00286 incrementally.

CLUPDATE 00289 seqstrt a one- to eight-digit numeric field specifying
CLUPDATE 00290 the first decimal number to be used for
CLUPDATE 00291 sequencing the inserted records.

CLUPDATE 00294 seqincr a one- to eight-digit numeric field specifying
CLUPDATE 00295 the decimal increment for sequencing the
CLUPDATE 00296 inserted records.

CLUPDATE 00298 All records following the './ I' card, up to the next
CLUPDATE 00299 control card, are inserted in the output file following the
CLUPDATE 00300 record identified by the 'seqno' field. If the NOINC option
CLUPDATE 00300 was specified, each inserted record is identified with
CLUPDATE 00302 asterisks ('********') in columns 73-80. If either the INC
CLUPDATE 00302 or CTL option was specified, the records are inserted
CLUPDATE 00303 unchanged in the output file, or they are sequenced
CLUPDATE 00303 according to the 'seqstrt' and 'seqincr' fields if the
CLUPDATE 00304 dollar sign '$' key was included.

CLUPDATE 00305 The default sequence increment, if the dollar sign is
CLUPDATE 00306 included, is determined by using one tenth of the least
CLUPDATE 00307 significant, non-zero digit in the 'seqno' field, with a
CLUPDATE 00307 maximum of 100. The default 'seqstrt' is computed as 'seqno'
CLUPDATE 00309 plus the default 'seqincr'. For example, the card:

```
./  I  2600 $ 2610
```

CLUPDATE 00311 would cause the inserted records to be sequenced XXX02610,
CLUPDATE 00313 XXX02620, etc. (NOSEQ8 assumed here). For the card:

```
./  I  240000 $
```

CLUPDATE 00315 the defaulted 'seqincr' would be the maximum, 100, and the
CLUPDATE 00316 starting sequence would be 240100. SEQ8 was assumed, so the
CLUPDATE 00317 inserted records would be sequenced 00240100, 00240200, etc.

CLUPDATE 00318 If either INC or CTL was specified but the dollar sign was
CLUPDATE 00319 not included, whatever sequence appears on the inserted
CLUPDATE 00319 records in the update file will be included in the output
CLUPDATE 00320 file.

CLUPDATE 00322 3. Delete one or more records from the source file.

CLUPDATE 00329
CLUPDATE 00332
CLUPDATE 00333

```
./  D  seqno1  [seqno2]  [$]
```

CLUPDATE 00342 seqno1 specifies the sequence field identifying the
CLUPDATE 00343 first or only record to be deleted.

CLUPDATE 00346 seqno2 specifies the sequence field of the last record
CLUPDATE 00347 to be deleted.

CLUPDATE 00350 $ optional delimiter indicating the end of the
CLUPDATE 00351 control fields.

CLUPDATE 00353 All records of the input file, beginning at 'seqno1', are
CLUPDATE 00354 deleted from the output file, up to and including the
CLUPDATE 00354 'seqno2' record. If the 'seqno2' field is omitted, only a
CLUPDATE 00356 single record is deleted.

CLUPDATE 00357 4. Replace one or more input records with updated records
CLUPDATE 00358 from the update file.

CLUPDATE 00365
CLUPDATE 00368 | ./ R seqno1 [seqno2] [$ [seqstrt [seqincr]]] |
CLUPDATE 00369

CLUPDATE 00378 seqno1 specifies the sequence number of the first input
CLUPDATE 00379 record to be replaced.

CLUPDATE 00382 seqno2 specifies the sequence number of the last record
CLUPDATE 00383 to be replaced.

CLUPDATE 00386 $ optional delimiter key indicating that the
CLUPDATE 00386 substituted records are to be sequenced
CLUPDATE 00387 incrementally.

CLUPDATE 00390 seqstrt a one- to eight-digit numeric field specifying
CLUPDATE 00391 the first decimal number to be used for
CLUPDATE 00392 sequencing the substituted records.

CLUPDATE 00395 seqincr a one- to eight-digit numeric field specifying
CLUPDATE 00396 the decimal increment for sequencing the
CLUPDATE 00397 substituted records.

CLUPDATE 00399 All records of the input file, beginning with the 'seqno1'
CLUPDATE 00400 record, up to and including the 'seqno2' record, are
CLUPDATE 00400 replaced in the output file by the records following the './
CLUPDATE 00401 R' card in the update file, up to the next control card. As
CLUPDATE 00401 with the './ D' function, if the 'seqno2' field is omitted,
CLUPDATE 00402 only a single record is replaced, but it may be replaced by
CLUPDATE 00402 more than a single inserted record. The './ R' (replace)
CLUPDATE 00403 function is performed as a delete followed by an insert,
CLUPDATE 00403 such that the number of cards substituted need not match the
CLUPDATE 00404 number replaced. The dollar sign '$', 'seqstrt', and
CLUPDATE 00404 'seqincr' processing is identical to that for the insert
CLUPDATE 00405 function.

CLUPDATE 00407 5. Comment cards

CLUPDATE 00414
CLUPDATE 00417 | ./ * [comment] |
CLUPDATE 00418

```
CLUPDATE 00427   *          specifies that this is a comment card, and is to
CLUPDATE 00428              be ignored, except that it is copied into the
CLUPDATE 00429              log file.

CLUPDATE 00434   Summary of Input and Output Files to UPDATE Command

CLUPDATE 00436   1. Input files when a single update is to be applied.

CLUPDATE 00437   When the 'CTL' option is not specified in the UPDATE command
CLUPDATE 00438   line, then only one update is applied to the source file, in
CLUPDATE 00438   accordance with the methods described in the preceding two
CLUPDATE 00440   sections.  There are two input files:

CLUPDATE 00444        a. The source file, which is to be updated.  The
CLUPDATE 00444           filename of this file must be specified in the
CLUPDATE 00445           command line.  The filetype and filemode will
CLUPDATE 00445           default to ASSEMBLE and A1, respectively, unless
CLUPDATE 00447           overridden by the command line.

CLUPDATE 00451        b. The update file, whose control cards have been
CLUPDATE 00452           described in the preceding section. The filename of
CLUPDATE 00452           this file will default to the filename of the source
CLUPDATE 00453           file, and the filetype and filemode will default to
CLUPDATE 00453           UPDATE and A1, respectively. All three may be
CLUPDATE 00454           overridden by the command line.

CLUPDATE 00457   2. Output files when a single update is applied.

CLUPDATE 00458   When a single update is applied to the source file, the
CLUPDATE 00459   following output files are created:

CLUPDATE 00462        a. An updated source file is created.  The filename of
CLUPDATE 00463           this file will be "$fn", where "fn" is the filename
CLUPDATE 00463           of the original source file, unless the REP option
CLUPDATE 00463           was specified.  In the latter case, the filename of
CLUPDATE 00464           this file will be "fn".  (For modifications of these
CLUPDATE 00465           rules when an error occurs, see the section below on
CLUPDATE 00466           update error handling.)

CLUPDATE 00468        b. An update log, showing all transactions and errors,
CLUPDATE 00468           will be created.  The filename of the file will be
CLUPDATE 00469           "fn", where "fn" is the filename of the original
CLUPDATE 00469           source file, and the filetype of this file will be
CLUPDATE 00470           UPDLOG. Note, however, that if the PRINT option is
CLUPDATE 00471           specified with the command line, then the update log
CLUPDATE 00471           will be printed directly on the virtual spooled
CLUPDATE 00472           printer, and no disk file will be created.

CLUPDATE 00475   3. Input files when multi-level updates are applied.

CLUPDATE 00476   When the CTL option is specified on the command line, then
```

multi-level updates are applied to the source file. In this case, the following files are inputted to the UPDATE command:

a. A source file, specified in exactly the same way as the source file for a single update.

b. A control file, which controls what updates are applied, and the order in which they are to be applied. The filename of this file will default to the file name of the source file, and the filetype and filemode will default to CNTRL and A1, respectively. All three may be overridden by the command line. This file will contain, in its control cards, pointers to update files, PTM files, and 'AUX' auxiliary files, which are listed in the following sections.

c. One or more update files, as specified by the control file. The filename of these files will be the same as the filename of the source file. The filetype of these files will "UPDTxxxx", where the "xxxx" portion is specifed by the control file.

d. 'AUX' auxiliary files, as specified by the control file. The filename of these files will be the same as the filename of the source file. The filetype of these files will be "AUXxxxx", where the "xxxx" portion is specified by the control file. The AUX files contain additional control cards pointing to PTM files. The format of the AUX files is described in a later section.

e. PTM files, as specified by either the control file or the AUX files. The filename of these files is the same as the filename of the source file. The filetype will be specified in full by the control file or the AUX file. In format, these files are identical to ordinary update files. They are given the special name "PTM files," in order to conform to IBM APAR processing procedures.

4. Output files when multi-level updates are applied.

When the CTL option is specified, then the following output files are created by the UPDATE command:

a. An updated source file, as in the case of a single update.

b. An update log, as in the case of a single update.

c. An UPDATES file. This file will have the filename of the original source file, and a filetype of UPDATES. It contains summary information about which updates were applied to the file, and is intended to be concatenated onto the assembly text deck for documentation and information purposes.

d. Although not a disk file, additional "output" is produced in the form of lines placed in the terminal read stack, for interrogation by an EXEC file which may have invoked the UPDATE command. These lines are placed there only if the STK option is specified.

5. Disk mode of output files

If there are several read/write disks accessed when the UPDATE command is invoked, then the following steps are taken to determine the disk upon which the output files are to be placed (the search stops as soon as one of the following steps is successful):

a. If the disk on which the original source file lies is read/write, then the output files are placed on that disk.

b. Otherwise, if that disk is a read/only extension of a read/write disk, then the output files are placed on that particular read/write disk.

c. Otherwise, the output files are place on the primary read/write disk (the A-disk).

The 'CTL' Option

If the NOCTL option is specified or defaulted, UPDATE will process one input file and one update file to produce an updated source output file and an update log file containing a record of what changes were made. This mode of operation is suitable for testing fixes and modifications prior to incorporating them in the base source code, providing that only one set of changes has to be tested at a time. If, for any reason, more than one set of changes is outstanding against a single source input file, the difficulties in managing that base code can multiply very rapdily. For this reason, UPDATE provides the CTL option, which embodies a multiple-update control and management scheme developed for use in maintaining VM/370 distributed source code, and may be used wherever its advantages are felt.

The major components of the multi-level update scheme are as follows:

1. A set of base source code which is not permanently changed.
2. A set of update files for each source file which must be applied in a specific order.
3. One or more 'CNTRL' files which describe the order or priority of updates to be applied to each source file.
4. Optionally, one or more auxiliary ('AUX') control files, each specific to a certain source file.

An integral part of the multi-level update scheme is a naming convention for the update files themselves, and for any TEXT files produced by assembling or compiling the updated output files. In normal usage, any update file will have the filename of the source file to which it applies and the filetype of 'UPDATE'. When the CTL option is used to invoke the multi-level update controls, the filename usage becomes a requirement, such that the update files must have the filename of the source file to which they apply, but the filetypes are modified to distinguish between separate update levels. The filetype for an update file is constructed from 'UPDT' plus a one- to four-character update identifier. For example, for a source file of 'DMSUPD ASSEMBLE', there might be three update files, named 'DMSUPD UPDT750', 'DMSUPD UPDTX4', and 'DMSUPD UPDT009'. In this example, the update identifiers are '750', 'X4', and '009'. The 'CNTRL' file, mentioned in (3) above, specifies which update files are to be applied to the source file and in what order they are to be applied, on the basis of the update identifier. Another identification parameter, the update level identifier, is used when naming a TEXT file produced from the updated source file. The update level identifier is specified by the 'CNTRL' file and is associated with a specific update identifier, also in the 'CNTRL' file. For example, a file named 'X4 CNTRL', to apply the above mentioned updates to 'DMSUPD ASSEMBLE', might look like the following:

```
00D  MACS  DMSLIB  SYSLIB
X4D  X4
75X  750
009X 009
```

This control file would apply the updates 'DMSUPD UPDT009', 'DMSUPD UPDT750', and 'DMSUPD UPDTX4', in that order, to the file 'DMSUPD ASSEMBLE'. The updates are applied in reverse order as they appear in the 'CNTRL' file, that is, the 'lowest-level' of update is at the bottom of the file, and the 'highest-level' update is at the top. As the CNTRL file and update files are processed, UPDATE will type at the terminal the message:

```
CLUPDATE 00629          DMSUPD178I UPDATING <'fn ft fm'> WITH 'fn ft fm'

CLUPDATE 00631  for each update file which is applied to the source input
CLUPDATE 00632  during the multi-level update, where the bracketed
CLUPDATE 00633  expression will be typed only for the first update.

CLUPDATE 00634  In the above example, the fields 'X4D', '75X', and '009X'
CLUPDATE 00636  are the update level identifiers, associated with the 'X4',
CLUPDATE 00637  '750', and '009' update identifiers, respectively. According
CLUPDATE 00637  to the naming convention for VM/370 TEXT files, the
CLUPDATE 00638  resultant of assembling the updated '$DMSUPD ASSEMBLE' file
CLUPDATE 00638  would be named 'DMSUPD TXTX4D', where the 'X4D' is the
CLUPDATE 00640  update level identifier of the highest-level update applied.
CLUPDATE 00640  The 'TXT' portion of the filetype indicates that this is a
CLUPDATE 00641  TEXT file, but allows up to a five-character update level
CLUPDATE 00642  identifier.

CLUPDATE 00644  CNTRL File Format

CLUPDATE 00645  The CNTRL files used by the multi-level update scheme must
CLUPDATE 00646  be 80-byte, card-image files. There are six different
CLUPDATE 00647  control record formats recognized by UPDATE, as follows:

CLUPDATE 00651      1. The 'MACS' record, normally the first line in the
CLUPDATE 00652         CNTRL file.
CLUPDATE 00654      2. Comment cards, containing an asterisk (*) in column
CLUPDATE 00655         1, which are ignored.
CLUPDATE 00657      3. Update level control records, containing both an
CLUPDATE 00658         update level identifier and an update identifier.
CLUPDATE 00660      4. TEXT-file control records, containing only an update
CLUPDATE 00661         level identifier, which are ignored by UPDATE.
CLUPDATE 00663      5. Auxiliary file control records, containing an update
CLUPDATE 00664         level identifier, an update identifier, and the
CLUPDATE 00665         keyword 'AUX'.
CLUPDATE 00667      6. Single PTF control records, containing the keyword
CLUPDATE 00668         'PTF' in place of the update level identifier, and
CLUPDATE 00669         up to an eight-character update identifier.

CLUPDATE 00672  All fields on CNTRL and AUX file records are free-format,
CLUPDATE 00673  blank-delimited fields which may not extend beyond record
CLUPDATE 00673  position 72. Any sequence fields on CNTRL or AUX file
CLUPDATE 00674  records are ignored. The formats and uses of each record
CLUPDATE 00675  format, by number, are as follows:

CLUPDATE 00678  1. The 'MACS' record - must precede any other records
CLUPDATE 00679     except comments.

CLUPDATE 00681                  uplevel MACS library1 library2 ...

CLUPDATE 00682     This record defines the default update level
CLUPDATE 00683     identifier, i.e. the update level identifier which is
CLUPDATE 00683     associated with the application of no updates at all,
```

CLUPDATE 00684 and it defines the list of macro libraries which should
CLUPDATE 00684 be used when assembling or compiling the updated source
CLUPDATE 00685 files. The keyword 'MACS' identifies this record
CLUPDATE 00686 format.

CLUPDATE 00689 2. Comments - may occur anywhere within either the CNTRL
CLUPDATE 00690 file or any AUX files.

CLUPDATE 00692 * comments...

CLUPDATE 00694 The asterisk must appear in record position one.

CLUPDATE 00697 3. Update level control records - specifies an update to
CLUPDATE 00698 be applied.

CLUPDATE 00700 uplevel upid

CLUPDATE 00701 The update file named 'fname UPDTupid', if it exists,
CLUPDATE 00702 is to be applied to the source input file, and the
CLUPDATE 00702 level identifier 'uplevel' is to be associated with the
CLUPDATE 00704 application of that update file. The 'uplevel' field is
CLUPDATE 00704 a maximum of five characters, and the 'upid' field is a
CLUPDATE 00705 maximum of four characters. If the specified 'UPDTupid'
CLUPDATE 00705 file is not found by UPDATE, no error occurrs and the
CLUPDATE 00706 next higher update level control record is processed,
CLUPDATE 00706 until the top of the CNTRL file is reached or the
CLUPDATE 00707 'MACS' record is encountered.

CLUPDATE 00710 4. TEXT-file control records - ignored by the UPDATE
CLUPDATE 00711 command.

CLUPDATE 00713 uplevel

CLUPDATE 00714 This record format is intended for use by the VM/370
CLUPDATE 00715 service program VMFLOAD, which will produce a system
CLUPDATE 00715 load deck from multiple 'TXTuplevel' files. The UPDATE
CLUPDATE 00716 command ignores any TEXT-file control records since
CLUPDATE 00717 they do not specify an update identifier.

CLUPDATE 00720 5. Auxiliary file control records - specifies 'fname
CLUPDATE 00721 AUXupid' files.

CLUPDATE 00723 uplevel upid AUX

CLUPDATE 00724 The keyword 'AUX', following the update identifier,
CLUPDATE 00725 indicates that this single update level may, in itself,
CLUPDATE 00725 consist of multiple update files, specified in list
CLUPDATE 00726 form in the file 'fname AUXupid'. This feature is used
CLUPDATE 00726 where a set of update files are inter-dependent (e.g.
CLUPDATE 00727 related PTF's). The auxiliary file may contain comment

records and/or update specification records, as follows:

```
        *  PTF A34216CA FIXES MNOP BUG IN DMKCCW
        *       A34216CA
        *  PTF  A09156CA  REPAIR TRANSLATE TABLE
        *
```

As noted in the above example, the auxiliary file mechanism is used for applying PTF's, in this case the files 'fname A09156CA', and 'fname A34216CA' would be applied to the input file as updates, in that order. The major difference in processing when the 'AUX' option is used, is that all updates specified in the auxiliary file must be present, or an error is indicated, and that the full eight-character filetype is specified instead of the 'UPDTxxxx' modifier. The filetypes shown above are in IBM standard APAR-answer format, consisting of 'A....6CA', with the APAR number in the middle. The updates specified in the auxiliary file are also applied in reverse order, from the lowest to the highest level, as they are in the normal CNTRL file.

6. Single PTF control records - specifies a PTF to be applied.

```
        PTF  filetype
```

The keyword 'PTF' in the update level identifier field indicates that a full filetype was specified in the 'upid' field instead of a four-character update identifier, and that this update file has no update level identifier associated with its application. Since the update level identifier will not indicate whether or not the PTF was applied, this form is not recommended for applying PTF's which are specific to certain source files. The purpose of this format is to provide a means of applying a 'global' PTF to many source files, without requiring a separate 'AUXupid' file for each source file affected. No error is indicated if the PTF specified is not found.

Each CNTRL file to be used by the UPDATE command must contain a single 'MACS' record as the first non-comment record, but thereafter may contain any of the records specified in any order.

The SIK/NOSIK Options

The STK option is provided for use with the multi-level update invoked via the CTL option, primarily for communication with CMS EXEC procedures which invoke UPDATE. If the CTL and STK options are specified, UPDATE will place two lines of data in the CMS terminal read stack, as follows:

first line = * update level identifier
second line = * library list from 'MACS' record

These lines are placed in the terminal read stack via the CMS 'ATTN' function, and are available to an invoking EXEC procedure via the EXEC control words '&READ ARGS' or '&READ VARS'. The first line, the update level identifier, is the level identifier of the highest-level update applied; this is the TEXT-file filetype modifier used by the VM/370 maintenance procedures. The second line consists of the list of libraries specified on the 'MACS' record in the CNTRL file. The library search order for an assembly or compilation can be established by issuing the 'GLOBAL' command using the library list returned.

If the NOSTK option is used with the multi-level update, no data is made available to external procedures.

Warning and Error Handling

The UPDATE command detects a number of invalid requests, and decides whether they should be treated as warning situations or as errors. The following is a general description of the handling of these situations, and the return codes associated with each:

1. Sequencing and Update Control Card Errors

These errors are treated as "warning situations." That is, a warning message is generated, and processing continues. The warning messages will be printed in the update log, and will be typed on the terminal unless the NOTERM option is specified in the command line.

a. Input sequencing errors (return code = 4). These errors are detected when the input source file contains sequence errors (sequence numbers in non-ascending order).

b. Output sequencing errors (return code = 8). These errors are detected when the updating procedure introduces new sequencing errors into the output source file.

c. Invalid update control cards (return code = 12). These errors are detected when the update file contains invalid control cards. (As described below, erroneous cards in control files cause the update procedure to abort.)

If more than one such error is detected, the update will return the maximum return code (4, 8 or 12) encountered.

If any such error is detected, then the REP option, if specified, will be ignored, and the output source file will remain with the filename "$fn", as if NOREPLACE were in effect.

2. Other errors.

These errors include such things as invalid control file cards, invalid file formats, and disk i/o errors. The UPDATE command processing is aborted as soon as the error is detected. The return code will always be 20 or greater.

If any such error is detected, then the output file will be left with the filename UPDATE and the filetype CMSUT1, so that the user may, if he wishes, examine or otherwise make use of it. This file must be erased before the UPDATE command can be invoked again.

CLIMCOM 00004 **IMMEDIATE COMMANDS**

CLIMCOM 00007 An IMMEDIATE Command is issued after an Attention interrupt
CLIMCOM 00008 has been given to CMS by pressing the ATTN key (or its
CLIMCOM 00010 equivalent). Such commands are processed immediately upon
CLIMCOM 00011 entry at the console or on being 'stacked' by an EXEC
CLIMCOM 00012 procedure. Any program execution in progress is suspended
CLIMCOM 00013 until the immediate command is processed. The commands are
CLIMCOM 00014 HO (halt SVC tracing), HT (halt typing), HX (halt
CLHO 00001 execution), and RT (resume typing).

CLHO 00009 **HQ**

CLHO 00012 The HO command may be issued during the execution of a
CLHO 00013 command or user program to stop the recording of trace
CLHO 00014 information. In order for the HO command to be recognized it
CLHO 00015 must be entered after stopping program execution by an
CLHO 00016 Attention interrupt. Program execution continues to its
CLHO 00017 normal completion, and all recorded trace information is
CLHO 00018 spooled to the printer.

CLHO 00020
CLHO 00021 | HO |
CLHO 00022

CLHO 00026 **Responses**

CLHX 00001 None.

CLHX 00008 **HX**

CLHX 00011 The HX command causes the execution of any CMS command or
CLHX 00011 user program executing under CMS to stop, closes any open
CLHX 00012 files or I/O devices, and returns the user to the CMS
CLHX 00013 command environment. In order for the HX command to be
CLHX 00014 recognized, it must be issued after stopping program
CLHX 00015 execution by an Attention interrupt.

CLHX 00017 ┌──────────────────────────────┐
CLHX 00018 | HX |
CLHX 00019 └──────────────────────────────┘

CLHX 00023 **Responses**

CLHT 00001 None.

CLHT 00008 **HT**

CLHT 00011 The HT command causes all terminal output generated by the
CLHT 00012 CMS command or user program in progress to be suppressed. In
CLHT 00013 order to enter the HT command meaningfully, an Attention
CLHT 00014 interrupt must be simulated by pressing the ATTN key.
CLHT 00015 Program execution continues, but the keyboard is unlocked to
CLHT 00016 accept user input which is stacked and held until the
CLHT 00016 current command completes. Execution continues to normal
CLHT 00017 program completion. When the READY message is typed out,
CLHT 00019 normal terminal output resumes.

CLHT 00021 ┌──────────────────────┐
CLHT 00022 | HT | |
CLHT 00023 └──────────────────────┘

CLHT 00027 **Responses**

CLRT 00001 None.

CLRT 00008 RT

CLRT 00011 The RT command restores terminal typeout from an executing
CLRT 00012 CMS command or user program that was previously suppressed
CLRT 00013 by the HT command. In order to enter the RT command
CLRT 00013 meaningfully, an Attention interrupt must be simulated by
CLRT 00014 pressing the ATTN key. Program execution continues, and
CLRT 00015 previously suppressed typeout continues. Execution continues
CLRT 00017 to normal program completion.

CLRT 00019 ┌──────────────────────────────┐
CLRT 00020 | RT |
CLRT 00021 └──────────────────────────────┘

CLRT 00025 Responses

CLPRIVCL 00005 None.

CP COMMANDS

CP PRIVILEGE CLASSES

The CP commands are divided into eight groups, each being represented by a privilege class which indicates the source from which the commands will be accepted. Each user is assigned, as part of his entry in the user directory, one or more privilege classes. Figure 27 shows the function of each class and Figure 28 shows the commands which are accepted from users in each class.

A	System Operator: The function of the Class A user is to control the VM/370 system. Class A is assigned to the user who is controlling the CPU console typewriter at VM/370 IPL time. Note: The Class A Systems Operator who is automatically logged in at CP initialization is designated as the Primary System Operator.
B	System Resource Operator: The function of the Class B user is to control the real resources of the System/370 computer.
C	System Programmer: The function of the Class C user is to maintain and modify certain functions of the VM/370 system.
D	Spooling Operator: The function of the Class D user is to control the system unit record equipment and the spool data files.
E	System Analyst: The function of the Class E user is to examine certain data in the VM/370 storage area.
F	Service Representative: The function of the Class F user is to obtain and examine in detail certain data about input/output devices connected to the System/370 computer.
G	General User: The function of the Class G user is to control functions associated with the execution of his virtual machine.
H	reserved for future use.

Figure 27. CP Privilege Class Descriptions

ALL	DIAL, DISCONN, LOGIN, LOGOUT, MSG, QUERY LOGMSG, QUERY NAMES, QUERY userid, QUERY USERS, SLEEP, *
A	ACNT, DISABLE, ENABLE, FORCE, HALT, LOCK, QUERY PAGING, QUERY PRIORITY, SET FAVORED, SET PAGING, SET PRIORITY, SET RESERVE, SHUTDOWN, UNLOCK, WNG
B	ATTACH, ATTACH CHANNEL, DETACH, DETACH CHANNEL, DISABLE, ENABLE, MSG ALL, QUERY DASD, QUERY DUMP, QUERY LINES, QUERY raddr, QUERY STORAGE, QUERY SYSTEM, QUERY TAPE, QUERY UR, SET DUMP, SET LOGMSG, VARY, WNG
C	STCP
D	BACKSPAC, CHANGE, DRAIN, FLUSH, FREE, HOLD, LOADBUF, ORDER, PURGE, QUERY FILES, QUERY HOLD, QUERY PRINTER, QUERY PUNCH, QUERY READER, REPEAT, SPACE, START, TRANSFER
E	DCP, DMCP, LOCATE, QUERY PAGING, QUERY PRIORITY, SAVESYS
F	SET RECORD
G	ADSTOP, BEGIN, CHANGE, CLOSE, COUPLE, DEFINE, DETACH, DISPLAY, DUMP, ECHO, EXTERNAL, IPL, LINK, NOTREADY, ORDER, PURGE, QUERY FILES, QUERY LINKS, QUERY PRINTER, QUERY PUNCH, QUERY READER, QUERY SET, QUERY TERMINAL, QUERY TIME, QUERY VIRTUAL, READY, RESET, REWIND, SET ACNT, SET EMSG, SET LINEDIT, SET MSG, SET NOTRAN, SET RUN, SET TIMER, SET WNG, SPOOL, STORE, SYSTEM, TERMINAL, TRACE, TRANSFER, LOADVFCB

Figure 28. Commands Used by Each Privilege Class

CP COMMAND SUMMARY

This section contains descriptions of the commands acceptable in the control program environment. Figure 29 presents an alphabetical list of the commands, the privilege classes which may execute the command, and a brief statement about the use of each command.

COMMAND	PRIV CLASS	USAGE
*	ALL	annotate the console sheet
ACNT	A	create accounting records for logged on users and reset accounting data
ADSTOP	G	halt execution at a specific virtual machine instruction address
ATTACH	B	attach a real device to a virtual machine attach a DASD device for CP control dedicate all devices on a particular channel to a virtual machine
BACKSPAC	D	restart or reposition the output of a spooling device
BEGIN	G	continue or resume execution of the virtual machine at either a specific storage location or at the address in the current PSW
CHANGE	D,G	alter one or more external attributes of a closed spool file
CLOSE	G	terminate spooling operations on a virtual card reader,punch,or printer
COUPLE	G	connect channel-to-channel adapters
DCP	E	display real storage at terminal
DEFINE	G	reconfigure a virtual machine

Figure 29. CP Command Summary (Part 1 of 4)

COMMAND	PRIV CLASS	USAGE
DETACH	B,G	disconnect a real device from a virtual machine
		detach a DASD device from CP
		detach a channel from a specific user
		detach a virtual device from a virtual machine
DIAL	ALL	connect a terminal to virtual machine virtual communication line
DISABLE	A,B	disable communication lines
DISCONN	ALL	disconnect user's terminal from a virtual machine
DISPLAY	G	display virtual storage on user's terminal
DMCP	E	dump real storage on user's virtual printer
DRAIN	D	halt operations of specified spool devices upon completion of current operation
DUMP	G	print on virtual printer: virtual PSW, general registers, floating-point registers, storage keys, and contents of specified virtual storage locations
ECHO	G	test terminal hardware
ENABLE	A,B	enable communication lines
EXTERNAL	G	simulate an external interrupt for a virtual machine and return control to that machine
FLUSH	D	cancel output on a specific real unit record device
FORCE	A	cause logout of a specific user
FREE	D	remove spool HOLD status

Figure 29. CP Command Summary (Part 2 of 4)

COMMAND	PRIV CLASS	USAGE
HALT	A	terminate the active channel program on specified real device
HOLD	D	defer real spooled output of a particular user
IPL	G	simulate IPL for a virtual machine
LINK	G	provide access to a specific DASD device by a virtual machine
LOADBUF	D	load real UCS or FCB printer buffer
LOADVFCB	G	load virtual forms control buffer
LOCATE	E	find control blocks
LOCK	A	bring virtual pages into real storage and lock them thus excluding them from future paging
LOGIN	ALL	provide access to CP
LOGOUT	ALL	disable access to CP
MSG	B,ALL	transmit messages from user to user
NOTREADY	G	simulate 'not ready' for a device to a virtual machine
ORDER	D,G	rearrange closed spool files in a specific order
PURGE	D,G	remove closed spool file from system
QUERY	A,B,D,E,G,ALL	request information about machine configuration and system status
READY	G	simulates device end interrupt to specified virtual address
REPEAT	D	repeat printing or punching of current real spool output file
RESET	G	clear and reset all pending interrupts for a specified virtual device and reset all error conditions

Figure 29. CP Command Summary (Part 3 of 4)

COMMAND	PRIV CLASS	USAGE
REWIND	G	perform operation equivalent to rewinding and readying a tape
SAVESYS	E	save virtual machine memory space, registers, and PSW
SET	A,B,F,G	operator - establish system parameters user - control various functions within the virtual system
SHUTDOWN	A	terminate all CP functions and checkpoint CP system for warm start
SLEEP	ALL	place virtual machine's terminal in dormant state with keyboard locked
SPACE	D	force single spacing on printer
SPOOL	G	alter spooling control option
START	D	start spooling device after draining or changing output classes
STCP	C	allow alteration of real storage
STORE	G	alter specified virtual storage and registers
SYSTEM	G	simulates RESET, CLEAR STORAGE and RESTART buttons on console
TERMINAL	G	control terminal I/O processing
TRACE	G	trace specified virtual machine activity at the terminal, spooled printer or both
TRANSFER	D,G	direct spooled files to specified user's card reader
UNLOCK	A	unlock previously locked page frames
VARY	B	mark a device unavailable or available to CP
WNG	A,B	transmit a high priority message to a specified user or to all users

Figure 29. CP Command Summary (Part 4 of 4)

CLCPERR 00006 CP Error Messages and Responses

CLCPERR 00008 Refer to IBM Virtual Machine Facility/370, System Messages
CLCPERR 00009 (Order No. GC20-1808) for a description of the messages
CLCPERR 00010 produced by the CP commands. All messages from CP commands
CLACNT 00001 have the prefixes DMK.

CLACNT 00007 **ACNT**

CLACNT 00010 Privilege Class: A

CLACNT 00011 The ACNT command produces for each user specified, an
CLACNT 00012 accounting record giving the time used by his virtual
CLACNT 00013 machine since the last ACNT command was issued for him or
CLACNT 00014 since he logged in. The record produced is identical to the
CLACNT 00016 record given when the user logs out of VM/370. Each user
CLACNT 00017 receives a message at his terminal giving the accumulated
CLACNT 00018 accounting data, and the accumulated accounting data for
CLACNT 00019 each user is reset. A user can prevent his terminal from
CLACNT 00019 receiving the accounting message by having entered the CP
CLACNT 00021 command SET ACNT OFF.

CLACNT 00030 ┌──────────────────────────────────┐
CLACNT 00031 | ACNT | { userid1 userid2. .. } |
CLACNT 00032 | | { ALL } |
CLACNT 00033 └──────────────────────────────────┘

CLACNT 00039 userid specifies the users for which acccounting records
CLACNT 00041 are to be created.

CLACNT 00044 ALL specifies that accounting records are to be
CLACNT 00045 created for all logged-on users.

CLACNT 00048 Responses

CLACNT 00050 Format of accounting information message:

CLACNT 00052 CONNECT=hh:mm:ss VIRTCPU=mmm:ss.hh TOTCPU=mmm:ss.hh

CLACNT 00057 CONNECT hh:mm:ss the actual clock time spent in the
CLACNT 00057 user's current terminal session or the
CLACNT 00058 time elapsed since the ACNT command was
CLACNT 00059 last issued for the user.

CLACNT 00062 VIRTCPU mmm:ss.hh the CPU time spent in executing
CLACNT 00063 instructions within the virtual machine.

CLACNT 00066 TOTCPU mmm:ss.hh the CPU time spent in executing
CLACNT 00066 instructions within the virtual machine
CLACNT 00067 plus CPU time used by the control
CLACNT 00067 program in maintaining the virtual
CLACNT 00068 machine (paging, I/O, and so forth).

CLADSTOP 00001 COMMAND COMPLETE

CLADSTOP 00003 **ADSTOP**

CLADSTOP 00006 **Privilege Class:** G

CLADSTOP 00008 The ADSTOP command specifies a virtual instruction address
CLADSTOP 00009 where execution is to be halted. Execution halts when the
CLADSTOP 00010 instruction at the specified address is the next instruction
CLADSTOP 00011 to be executed.

CLADSTOP 00012 At the time execution halts, CP console function mode is
CLADSTOP 00014 entered and a message is typed. At this point the user may
CLADSTOP 00016 invoke other CP debugging commands. To resume operation of
CLADSTOP 00017 the virtual machine, the BEGIN command can be issued. Once
CLADSTOP 00018 an ADSTOP location is set, it may be removed by:

CLADSTOP 00022 1. reaching the virtual storage location specified in the
CLADSTOP 00023 ADSTOP command

CLADSTOP 00026 2. performing a virtual IPL or SYSTEM RESET,

CLADSTOP 00030 3. issuing the ADSTOP OFF command, or

CLADSTOP 00033 4. specifying a different location with a new ADSTOP
CLADSTOP 00034 hexloc command.

CLADSTOP 00044 ┌──────────────────────────┐
CLADSTOP 00045 | ADSTOP | { hexloc } |
CLADSTOP 00046 | | { OFF } |
CLADSTOP 00047 └──────────────────────────┘

CLADSTOP 00054 hexloc is the address, specified in hexadecimal digits,
CLADSTOP 00055 of the virtual machine instruction where execution
CLADSTOP 00056 is to be halted. An address stop location cannot
CLADSTOP 00057 be specified at a virtual address that is in a
CLADSTOP 00058 segment shared with other users, since the ADSTOP
CLADSTOP 00059 function modifies storage.

CLADSTOP 00061 OFF specifies that any previous ADSTOP setting is to
CLADSTOP 00062 be cancelled.

CLADSTOP 00064 **Note:** Since the ADSTOP function modifies storage (by placing
CLADSTOP 00065 a CP SVC X'B3' at the specified location) the user's program
CLADSTOP 00066 should not examine the two bytes at the instruction address.
CLADSTOP 00067 CP does not verify that the location specified contains a
CLADSTOP 00068 valid CPU instruction.

CLADSTOP 00071 **Responses**

CLADSTOP 00073 ADSTOP AT xxxxxx

CLADSTOP 00075 The instruction whose address is xxxxxx is the next
CLADSTOP 00076 instruction scheduled for execution. The virtual machine is
CLADSTOP 00077 in a stopped state. The CP command BEGIN may be be entered
CLADSTOP 00078 to resume execution at the instruction location xxxxxx, or
CLATTACH 00001 at any other location desired.

CLATTACH 00003 **ATTACH**

CLATTACH 00006 **Privilege Class:** B

CLATTACH 00009 The ATTACH command is used to logically connect a real
CLATTACH 00010 device to a virtual machine for that virtual machine's
CLATTACH 00011 exclusive use, or to logically connect a DASD device for CP
CLATTACH 00012 access and control.

CLATTACH 00023 ┌──┐
CLATTACH 00024 | ATTACH | raddr [TO] {userid [AS] vaddr [R/O] } |
CLATTACH 00025 | ATT | {SYSTEM [AS] volid } |
CLATTACH 00026 └──┘

CLATTACH 00033 raddr — real address (ccu) of the device to be attached.

CLATTACH 00036 TO
CLATTACH 00037 T — is an optional reserved keyword. If this word is
CLATTACH 00038 omitted and userid is specified, tserid cannct be "TO" or "T".

CLATTACH 00041 userid — is the user identification of the virtual machine
CLATTACH 00042 that is to receive the device. The user must be
CLATTACH 00043 logged in to receive the device. userid can be
CLATTACH 00043 specified as asterisk (*) if the operator is
CLATTACH 00045 attaching the device to his own virtual machine.

CLATTACH 00048 AS
CLATTACH 00049 A — is an optional reserved keyword. If this word is
CLATTACH 00051 omitted, vaddr, if specified, cannot be "A", and the volid, if specified, cannot be "AS" or "A".

CLATTACH 00053 vaddr — is the virtual address (ccu) to be assigned to the
CLATTACH 00054 device.

CLATTACH 00057 R/O
CLATTACH 00058 R — is optionally used for DASD devices that are
CLATTACH 00059 attached to a user. This allows the system
CLATTACH 00060 operator to apply read-only protection to a user's disk.

CLATTACH 00063 SYSTEM — this specification, along with volid, makes the
CLATTACH 00063 DASD device available to CP for accessing users
CLATTACH 00065 disks and/or for paging and spooling functions.

CLATTACH 00067 volid — is the volume label of the disk to be attached for
CLATTACH 00068 use by CP. This label will be verified before
CLATTACH 00070 attachment of the device.

CLATTACH 00074 **Responses**

CLATTACH 00076 Messages are sent to the user, the operator who issued the

command (if different from the user), and the primary system operator (if different from the operator who issued the command), notifying them that the ATTACH was successful.

type vaddr ATTACHED

is the response to the user indicating that the real device represented by address vaddr has been attached to his machine.

type raddr ATTACH TO userid vaddr
or
type raddr ATTACH TO SYSTEM volid

is the response to the Class B user who issued the command indicating that the real device represented by address raddr has been attached either to the user's virtual machine at virtual address vaddr or to CP.

type raddr ATTACH TO SYSTEM volid BY userid
or
type raddr ATTACH TO userid vaddr BY userid

is the response to the primary system operator (if he did not issue the command) indicating that the real device has been attached either to the specified user's machine at virtual address vaddr or to the CP system by the specified Class B user.

In the above responses 'type' is one of the following:

DASD	Direct access storage device
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel-to-channel adapter
DEV	Any other device

CLATCHAN 00003 **ATTACH CHANNEL**

CLATCHAN 00006 **Privilege Class:** B

CLATCHAN 00008 The ATTACH CHANNEL command dedicates all devices on a
CLATCHAN 00010 particular channel to a specified user. The addresses by
CLATCHAN 00011 which the virtual machine user refers to these devices must
CLATCHAN 00012 be the same as the real addresses. The devices on the
CLATCHAN 00013 attached channel do not have to be individually attached to
CLATCHAN 00014 the virtual user; he may use any or all of the devices. The
CLATCHAN 00015 virtual machine must have all I/O stopped before executing
CLATCHAN 00016 the command.

CLATCHAN 00026
CLATCHAN 00027
CLATCHAN 00028
CLATCHAN 00029

ATTACH ATT	CHANNEL c [TO] userid

CLATCHAN 00037 CHANNEL required reserved word

CLATCHAN 00039 c is the real address of the channel to be attached
CLATCHAN 00040 to the user.

CLATCHAN 00044 TO is an optional reserved keyword. If this word
CLATCHAN 00046 T is not specified, the userid cannot be "T or "TC".

CLATCHAN 00050 userid is the user identification of the virtual machine
CLATCHAN 00051 that is to use the dedicated channel. The user
CLATCHAN 00051 must be logged in in order to receive the
CLATCHAN 00052 channel. userid can be specified as asterisk (*)
CLATCHAN 00053 to indicate that the channel is to be attached to
CLATCHAN 00053 the virtual machine of the operator issuing the
CLATCHAN 00054 command.

CLATCHAN 00057 **Responses**

CLATCHAN 00059 Messages are sent to the user to whom the channel is
CLATCHAN 00060 attached, and to the operator who issued the command.

CLATCHAN 00062 CHANNEL c ATTACHED

CLATCHAN 00063 is the response to the user who receives dedicated access to
CLATCHAN 00064 the channel.

```
CLATCHAN 00067    CHANNEL c ATTACH TO userid

CLATCHAN 00069    is the response to the operator who issued the command.


CLATCHAN 00071    CHANNEL c ATTACH TO userid BY operator

CLATCHAN 00072    is the response to the primary system operator if he did not
CLATCHAN 00073    issue the command.
```

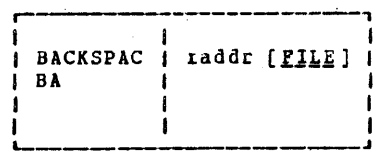
BACKSPAC

Privilege Class: D

The BACKSPAC command restarts or repositions the current output on a real punch or printer.

Printer BACKSPAC Format

Punch BACKSPAC Format



raddr is the address (cuu) of the real unit record device which is to be backspaced. This address must be that of a punch or a printer.

FILE
F specifies that the output spool printer or punch file is to be restarted from the beginning.

pages specifies the number of pages to be backspaced on the printer. If the field is not specified, the printer is backspaced to the beginning of the page being printed when the command is received.

Note: If the BACKSPAC command is issued while an intervention required condition is pending on the printer or punch, the interrupted channel program is terminated and the backspace is performed.

A BACKSPAC command for a punch is always treated as a BACKSPAC FILE.

Note: Pages for backspacing are counted starting from the end of the current spooling buffer. Output containing a small number of lines per page may require a count of more than one to backspace one page.

Pages are counted in BACKSPAC operations by looking for skip to channel 1 specifications (page ejects). If a file does not contain skip to channel 1 specifications, processing of the file will be restarted regardless of the number of pages indicated. Counting begins at the end of the current buffer.

CLBACKSP 00072 **Responses**

CLBACKSP 00076 { PRT } raddr BACKSPACE userid FILE = spoolid
CLBACKSP 00077 { PUN }

CLBACKSP 00080 raddr is the real device address and userid is the
CLBACKSP 00081 identification of the user whose file (spoolid) was active
CLBEGIN 00001 and backspaced.

CLBEGIN 00003 **BEGIN**

CLBEGIN 00006 **Privilege Class:** G

CLBEGIN 00009 The BEGIN command continues or resumes execution of the
CLBEGIN 00010 user's virtual machine at either a specified storage
CLBEGIN 00010 location or the location pointed to by the virtual machines
CLBEGIN 00011 current PSW.

CLBEGIN 00020 ┌──────────────────────┐
CLBEGIN 00021 | BEGIN | [hexloc] |
CLBEGIN 00022 | B | |
CLBEGIN 00023 └──────────────────────┘

CLBEGIN 00029 hexloc is the storage location, specified in hexadecimal
CLBEGIN 00031 digits, at which execution is to begin. If no
CLBEGIN 00032 location is specified, execution resumes at the
CLBEGIN 00035 location contained in the virtual machine current
CLBEGIN 00037 PSW. This would be the location at which the
CLBEGIN 00038 virtual machine was stopped when CE was last
CLBEGIN 00039 entered, unless the PSW had been altered since
CLBEGIN 00040 that time.

CLBEGIN 00043 **Responses**

CLCHANGE 00001 None. The virtual machine begins execution.

CHANGE

Privilege Class: D,G

The CHANGE command alters one or more of the external attributes of a closed spool file or files. The current attributes of a file may be determined by issuing a QUERY command. In order to change an output file, the file must have been closed but not yet selected for printing or punching. An input (READER) file may be changed at any time before it is opened, that is before the first read is issued for the file.

```
┌──────────────────────────────────────────────────────────────────────────────────────┐
|         |          {          {CLASS a}                  }                           |
| CHANGE  |          { READER   {spoolid}  CLASS b         }                           |
| CH      |          {          {ALL    }                  }                           |
|         |          {                                     }                           |
|         |[userid]  {                                     }[NAME{fn [ft]}}            |
|         |[SYSTEM]  {                                     }[     { dsname}]           |
|         |          {                                     }                           |
|         |          {{PRINTER} {CLASS a} {CLASS b      }*}                           |
|         |          {{PUNCH  } {spoolid} {COPY nn      } }                           |
|         |          {          {ALL    } {[HOID|NCHCID]} }                           |
|         |          {                    {DIST distccde} }                           |
|         |          {                                     }                           |
└──────────────────────────────────────────────────────────────────────────────────────┘
```

* one of these options must be chosen; however, more than one may be specified and they may be combined in any order. This is contrary to the notation normally used in this publication.

userid — (for use by Class D users only) this allows the spooling operator to manipulate the spool files for a particular user.

SYSTEM — (for use by Class D users only) this allows the spooling operator to manipulate all spool files in the system regardless of userid.

READER
R
RDR — indicates the type of file to be changed.

PRINTER
P
PRT

PUNCH
PU
PCH

```
CLCHANGE 00081    CLASS a    is used to designate an existing class. A is a one
CLCHANGE 00083    CL         alphanumeric character field from A to Z or from
CLCHANGE 00085               0-9. Refer to IBM Virtual Machine Facility/370,
CLCHANGE 00087               Operators Guide (Order No. GC20-1806) for a
CLCHANGE 00088               detailed description of spool classes.

CLCHANGE 00093    spoolid    is the spoolid number of the file that is to be
CLCHANGE 00094               changed. Each file has a unique spoolid.

CLCHANGE 00096    ALL        specifies that all files belonging to the user are
CLCHANGE 00097               to be changed.

CLCHANGE 00100    CLASS b    is specified to designate the new class of the
CLCHANGE 00102    CL         file.

CLCHANGE 00104    NAME       is a reserved word which is required if fn, ft, or
CLCHANGE 00106    NA         dsname is to be changed. This is specified to
CLCHANGE 00106               indicate that the name of the file with the given
CLCHANGE 00108               spoolid is to be altered.

CLCHANGE 00110    fn         is a 1 to 8 alphameric character name of a CMS
CLCHANGE 00111               file to be assigned to the file for identification
CLCHANGE 00112               purposes.

CLCHANGE 00117    ft         is a optional 1 to 8 alphameric character field,
CLCHANGE 00117               used for further file identification for CMS
CLCHANGE 00118               files.

CLCHANGE 00120    dsname     a 1 to 24 character field which is used to
CLCHANGE 00121               designate a non-CMS file.

CLCHANGE 00123    Note: If the NAME function is selected and only fn is
CLCHANGE 00124    specified, ft is set to blanks. The dsname specification
CLCHANGE 00125    allows a user to specify a file name suitable for OS or DOS
CLCHANGE 00126    files such as SYS1.SYSLIB.MYMAC.

CLCHANGE 00129    COPY nn    is specified to alter the number of real copies
CLCHANGE 00131    CO         of the file to be made. This option is valid for
CLCHANGE 00132               printer and punch files only. The value of nn
CLCHANGE 00133               (number of copies) must be a number from 1 through
CLCHANGE 00134               99. For nn less than 10, the leading zero is
CLCHANGE 00135               optional.

CLCHANGE 00138    HOLD       specifies that the file is not to be printed or
CLCHANGE 00139    HO         punched until released by the user by using the
CLCHANGE 00140               NOHOLD parameter.

CLCHANGE 00143    NOHOLD     specifies that the specified spool file currently
CLCHANGE 00144    NOH        being held is to be released.

CLCHANGE 00145    Note: If a Class D user specifies HOLD or NOHOLD for
```

CLCHANGE 00146 another users spool file, than that file is put in a
CLCHANGE 00147 "system" and "user" HOLD status (or in the case of NOHOLD
CLCHANGE 00148 specification, released from both statuses). A spool file
CLCHANGE 00149 in a system HOLD status can only be released (using the FREE
CLCHANGE 00150 or CHANGE command) by a Class D user.

CLCHANGE 00153 DIST is a keyword required if the distribution code of
CLCHANGE 00155 DI the file is to be changed.

CLCHANGE 00158 distcode is a 1 to 8 character alphameric identification
CLCHANGE 00158 that will appear on the output separators of
CLCHANGE 00159 printer and punch output instead of the usual
CLCHANGE 00159 identification specified in the user directory.
CLCHANGE 00160 The distribution code is changed for this file
CLCHANGE 00161 only, and does not affect other files or change
CLCHANGE 00162 the normal user code.

CLCHANGE 00165 Responses

CLCHANGE 00170 {nnnn} FILES CHANGED
CLCHANGE 00171 { NO }

CLCLOSE 00001 is the response to the user who issued the CHANGE command.

CLCLOSE 00004 CLOSE

CLCLOSE 00006 Privilege Class: G

CLCLOSE 00008 The CLOSE command terminates the spooling activity on any
CLCLOSE 00008 virtual spooled unit record device. If the file is an input
CLCLOSE 00009 reader file, the file being processed is purged unless the
CLCLOSE 00010 user has previously specified SPOOL READER HOLD (See SPOOL
CLCLOSE 00010 command). The effect of HOLD or NOHOLD for a particular
CLCLOSE 00011 file established by the SPOOL command can be overridden by
CLCLOSE 00012 specifying NOHOLD OR HOLD, respectively, in the CLOSE
CLCLOSE 00013 command. If the file is an output file on a printer or
CLCLOSE 00014 punch, the file is either queued for output on a real unit
CLCLOSE 00014 record device, or, if the virtual output device is
CLCLOSE 00015 transferred (SPOOL vaddr TO userid command), the file is
CLCLOSE 00016 queued for input to the receiving user. As optional
CLCLOSE 00017 arguments to the CLOSE command for output files, the user
CLCLOSE 00017 may provide a filename and filetype and an optional
CLCLOSE 00018 distribution code to aid in later identification of the file
CLCLOSE 00019 and its contents.

CLCLOSE 00030
CLCLOSE 00031 | CLOSE | { {READER} }
CLCLOSE 00032 | C | { { } [HOLD|NOHOLD] }
CLCLOSE 00033 | | { {vaddr } }
CLCLOSE 00034 | | { }
CLCLOSE 00035 | | { {PRINTER} | PURGE }
CLCLOSE 00036 | | { {PUNCH } | [HOLD | [DIST distcode] [NAME {fn [ft]} | }
CLCLOSE 00037 | | { {vaddr } | [NOHOLD] | {dsname } | }
CLCLOSE 00038 | | { | | }
CLCLOSE 00039 | | { | | }
CLCLOSE 00040

CLCLOSE 00046 vaddr is the virtual address (cuu) of the device to be
CLCLOSE 00047 closed. The address must represent a reader,
CLCLOSE 00048 printer or punch.

CLCLOSE 00052 READER specifies that all devices of the given type are
CLCLOSE 00054 R to be closed.
CLCLOSE 00056 RDR

CLCLOSE 00058 PRINTER
CLCLOSE 00060 P
CLCLOSE 00062 PRT

CLCLOSE 00064 PUNCH
CLCLOSE 00066 PU
CLCLOSE 00070 PCH

CLCLOSE 00073 HOLD specifies that the spool file is to be closed and
CLCLOSE 00074 HO is not available for further processing unless
CLCLOSE 00076 specifically requested or changed. This option,

```
CLCLOSE  00077              specified in the CLOSE command will override any
CLCLOSE  00077              previously specified HOLD or NOHOLD option for the
CLCLOSE  00078              files being closed.

CLCLOSE  00081    NOHOLD    specifies that the spool file is to be made
CLCLOSE  00082    NOH       available for further processing. This should
CLCLOSE  00083              only be specified if a HOLD established by the
CLCLOSE  00084              SPOOL command is still in effect and the current
CLCLOSE  00085              active file is not to be held.

CLCLOSE  00087              Note: HOLD and NOHOLD have no effect if the output
CLCLOSE  00089              file is subsequently transferred to another user.
CLCLOSE  00089              An output file that is held can be released by the
CLCLOSE  00090              user (or spooling operator) using the CHANGE
CLCLOSE  00091              command. An input file that is held is available
CLCLOSE  00091              as a complete spool input file to any spool reader
CLCLOSE  00092              of the virtual machine. Input files that are
CLCLOSE  00094              closed are normally purged from the system.

CLCLOSE  00098    PURGE     specifies that the output spool files is to be
CLCLOSE  00101    PU        closed and immediately purged from the system. No
CLCLOSE  00103              output file is produced.

CLCLOSE  00106    DIST      is a reserved keyword which is required if
CLCLOSE  00108    DI        the distribution code is to be changed.

CLCLOSE  00110    distcode  is a 1 to 8 character alphameric identification
CLCLOSE  00111              that will appear on the output separators of
CLCLOSE  00112              printer and punch instead of the usual
CLCLOSE  00112              identification specified in the user directory.
CLCLOSE  00113              The distribution code is changed for this file
CLCLOSE  00114              only and does not affect other files or change the
CLCLOSE  00114              normal user code. If the file is transferred to
CLCLOSE  00116              another user, this option has no effect.

CLCLOSE  00119    NAME      is a reserved word which is required if a name is
CLCLOSE  00121    NA        to be assigned to the spool file.

CLCLOSE  00123    fn        is a 1 to 8 alphameric character name to be
CLCLOSE  00125              assigned to the file for identification purposes.

CLCLOSE  00127    ft        is an optional 1 to 8 alphanumeric character field
CLCLOSE  00129              used for further file identification.

CLCLOSE  00131    dsname    a 1 to 24 character field which is used to
CLCLOSE  00133              designate a non-CMS file.

CLCLOSE  00135              Note:If the NAME function is selected and only fn
CLCLOSE  00136              is specified, ft is set to blanks The dsname
CLCLOSE  00136              specification allows the user to specify a file
CLCLOSE  00137              name suitable for OS or DOS files such as SYS1.
CLCLOSE  00139              SYSLIB. MUMAC for further file identification.
```

CLCLOSE 00143 **Responses**

CLCLOSE 00145 If multiple copies of the file are being processed, or if
CLCLOSE 00146 the file is being transferred to another user, the following
CLCLOSE 00147 message appears:

CLCLOSE 00152 {PRT} {TO userid}
CLCLOSE 00153 { } spoolid { }
CLCLOSE 00154 {PUN} {COPY nn }

COUPLE

Privilege Class: G

The COUPLE command allows a user with a virtual (non-dedicated) channel-to-channel adapter to connect that device to another user's (or his own, different) virtual channel-to-channel adapter.

```
┌──────────────────────────────────────────────────┐
│ COUPLE | vaddr1 [TO] userid vaddr2               │
└──────────────────────────────────────────────────┘
```

vaddr1 specifies the virtual address (cuu) of the channel-to-channel adapter of the user issuing the command

TO is an optional reserved keyword. If this word
T is not specified, userid may not be "TO" or "T".

userid is the user identification of the virtual machine to which vaddr1 is to be connected. If vaddr1 is to be connected to the user issuing the COUPLE command, userid may be specified as asterisk (*). The user must be logged on and have a virtual channel-to-channel adapter defined.

vaddr2 specifies the virtual address (cuu) of the channel-to-channel adapter to be connected to vaddr1 of the virtual machine user issuing the command.

Responses

CTCA vaddr1 COUPLE TO userid vaddr2

this is the response to the user issuing the COUPLE command

vaddr1 is the address of the issuing users channel-to-channel adapter; userid is the identification of the receiving virtual machine; and vaddr2 is the address of the channel-to-channel adapter of the receiving user.

CTCA vaddr2 COUPLE BY userid vaddr1

is the response to the user specified by userid in the COUPLE command. vaddr2 is the address of the issuing user's channel-to-channel adaptor; userid is the identification of the user who issued the COUPLE command; and vaddr1 is the

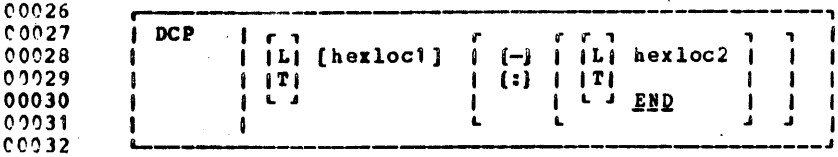
```
CLCOUPLE 00069     address of  the channel-to-channel adaptor of  the receiving
CLDCP    00001     user.
```

CLDCP 00004 **DCP**

CLDCP 00007 **Privilege Class:** E

CLDCP 00008 The DCP command allows a user to display the contents of
CLDCP 00010 real storage locations at the terminal.

CLDCP 00011 If an invalid argument is entered, the DCP command
CLDCP 00012 terminates; however, any previous valid arguments are
CLDCP 00013 processed before termination occurs. The operands for this
CLDCP 00014 command may be combined in any order desired, separated by
CLDCP 00015 one or more blanks.

CLDCP 00026
CLDCP 00027 
CLDCP 00028
CLDCP 00029
CLDCP 00030
CLDCP 00031
CLDCP 00032

CLDCP 00039 L specifies that the contents of the specified
CLDCP 00039 storage locations are to be displayed in
CLDCP 00040 hexadecimal.

CLDCP 00043 T specifies that, in addition to the hexadecimal
CLDCP 00043 display of the specified storage locations, an
CLDCP 00044 EBCDIC translation of the contents is to be
CLDCP 00045 displayed.

CLDCP 00048 hexloc1 specifies the first or only storage location to be
CLDCP 00048 displayed. If this field is not specified, L or T
CLDCP 00049 must be specified; the display will begin with
CLDCP 00050 storage location 0. If hexloc is specified and L
CLDCP 00051 or T is not specified, the display will be the
CLDCP 00052 same as if L were specified.

CLDCP 00055 : one of these delimiters must be specified in a
CLDCP 00056 - range specification. Either may be surrounded by
CLDCP 00058 none, one, or more blanks.

CLDCP 00061 hexloc2 specifies that a range of locations is to be
CLDCP 00062 **END** displayed. Real storage locations from hexloc1
CLDCP 00063 through hexloc2 is displayed. If END is
CLDCP 00064 specified, real storage from hexloc1 through the
CLDCP 00065 end of real storage is displayed. If this field is
CLDCP 00066 not specified, END is assumed by default.

CLDCP 00070 **Responses**

```
CLDCP    00072    Requested locations are typed in the following format:

CLDCP    00075    xxxxxx = word1 word2 word3 word4  *EBCDIC translation*

CLDCP    00077    Where xxxxxx is the real storage location of word1. Word1 is
CLDCP    00078    displayed (word aligned) for a single hexloc specification.
CLDCP    00079    Up to 4 words are displayed on a line.  If required,
CLDCP    00080    multiple lines are printed. The EBCDIC translation is
CLDCP    00080    displayed aligned to the next lower 16 byte boundary if
CLDCP    00081    Thexloc is specified. The output can be stopped and the
CLDCP    00082    command terminated by pressing the ATTN key (or its
CLDEFINE 00001    equivalent).
```

CLDEFINE 00004 **DEFINE**

CLDEFINE 00007 **Privilege Class:** G

CLDEFINE 00009 The DEFINE command provides the user with the ability to
CLDEFINE 00011 alter his virtual machine configuration. The user can expand
CLDEFINE 00012 his configuration without having to make permanent changes,
CLDEFINE 00013 since the definitions are in effect for the current terminal
CLDEFINE 00014 session only. If storage is redefined, the virtual machine
CLDEFINE 00015 is reset and must be re-IPLed.

CLDEFINE 00026
CLDEFINE 00027 | DEFINE | { {READER } } |
CLDEFINE 00028 | DEF | { {PRINTER} } |
CLDEFINE 00029 | | { {PUNCH } [AS] vaddr } |
CLDEFINE 00030 | | { {CONSOLE} } |
CLDEFINE 00031 | | { {CTCA } } |
CLDEFINE 00032 | | { {TIMER } } |
CLDEFINE 00033 | | { } |
CLDEFINE 00034 | | { LINE [AS] vaddr {IBM1 } } |
CLDEFINE 00035 | | { {TELE2} } |
CLDEFINE 00036 | | { } |
CLDEFINE 00037 | | { } |
CLDEFINE 00038 | | { vaddr1 [AS] vaddr2 } |
CLDEFINE 00039 | | { } |
CLDEFINE 00040 | | { {T2314} } |
CLDEFINE 00041 | | { {T2319} [AS] vaddr [CYL] nnn } |
CLDEFINE 00042 | | { {T3330} } |
CLDEFINE 00043 | | { {T2305} } |
CLDEFINE 00044 | | { } |
CLDEFINE 00045 | | { STORAGE [AS] nnnnnK } |
CLDEFINE 00046

CLDEFINE 00052 AS is an optional reserved keyword.
CLDEFINE 00054 A

CLDEFINE 00056 vaddr specifies the virtual address (ccu) to be assigned
CLDEFINE 00058 to the device being added.

CLDEFINE 00061 READER specifies that a spooling card reader with the
CLDEFINE 00063 R address specified by vaddr is to be added to the
CLDEFINE 00065 RDR virtual machine configuration.

CLDEFINE 00067 PRINTER specifies that a spooling printer with the address
CLDEFINE 00069 P specified by vaddr is to be added to the virtual
CLDEFINE 00071 PRT machine configuration.

CLDEFINE 00073 PUNCH specifies that a spooling card punch with the
CLDEFINE 00075 PU address specified by vaddr is to be added to the
CLDEFINE 00077 PCH virtual machine configuration.

CLDEFINE 00079 CONSOLE specifies that a virtual system console is to be

```
CLDEFINE 00080     CON        added to the virtual machine at the address
CLDEFINE 00081                specified by vaddr.

CLDEFINE 00083     CTCA       specifies that a virtual channel-to-channel
CLDEFINE 00084                adapter with the address specified by vaddr is to
CLDEFINE 00086                be added to the virtual machine configuration.

CLDEFINE 00089     TIMER      specifies that a pseudo timing device with the
CLDEFINE 00091     TIM        address specified by vaddr is to be added to the
CLDEFINE 00092                virtual machine configuration.

CLDEFINE 00095     LINE       specifies that a virtual 270X communication line
CLDEFINE 00096     LI         with the address specified by vaddr is to be added
CLDEFINE 00098                to the virtual machine configuration.

CLDEFINE 00101     IBM1       specifies that an IBM type of terminal (2741,
CLDEFINE 00103     IBM        1050, or equivalent) is on the 270X line.

CLDEFINE 00106     TELE2      specifies that a teletypewriter is on the 270X
CLDEFINE 00108     TEL        line.

CLDEFINE 00111     vaddr 1 AS vaddr2
CLDEFINE 00112                specifies that the device represented by vaddr1 is
CLDEFINE 00113                to be redefined as vaddr2. The first vaddr
CLDEFINE 00114                specified must be a defined device in the virtual
CLDEFINE 00115                machine configuration.

CLDEFINE 00118     T2314      specifies that a temporary virtual 2314 is to be
CLDEFINE 00119                added to the virtual machine configuration. The
CLDEFINE 00119                address of the 2314 is specified by vaddr and the
CLDEFINE 00120                disk contains the number of cylinders specified by
CLDEFINE 00121                CYL nnn.

CLDEFINE 00123     T2319      specifies that a temporary virtual 2319 is to be
CLDEFINE 00125                added to the virtual machine configuration. The
CLDEFINE 00125                address of the 2319 is specified by vaddr and the
CLDEFINE 00126                disk contains the number of cylinders specified by
CLDEFINE 00127                CYL nnn.

CLDEFINE 00129     T3330      specifies that a temporary virtual 3330 is to be
CLDEFINE 00131                added to the virtual machine configuration. The
CLDEFINE 00132                address of the 3330 is specified by vaddr and the
CLDEFINE 00133                disk contains the number of cylinders specified by
CLDEFINE 00134                CYL nnn.

CLDEFINE 00137     T2305      specifies that a temporary virtual 2305 is to be
CLDEFINE 00137                added to the virtual machine configuration. The
CLDEFINE 00138                address of the 2305 is specified by vaddr and the
CLDEFINE 00139                disk contains the number of cylinders specified by
CLDEFINE 00140                CYL nnn.

CLDEFINE 00143     CYL        is an optional reserved keyword.

CLDEFINE 00145     nnn        specifies the number of cylinders the disk is to
```

CLDEFINE 00146 contain.

CLDEFINE 00149 **STORAGE** specifies that the size of the virtual storage for
CLDEFINE 00151 **STOR** the virtual machine is to be redefined. The value
CLDEFINE 00152 specified will become the new virtual storage
CLDEFINE 00152 size. Sizes must be in 4K increments and are
CLDEFINE 00153 limited by the maximum value in the user directory
CLDEFINE 00154 entry. Changing the virtual storage size
CLDEFINE 00154 (increasing or decreasing) causes a virtual system
CLDEFINE 00155 reset and the clearing of all virtual storage to
CLDEFINE 00156 binary zeros.

CLDEFINE 00158 nnnnnK specifies the new size of the machine's virtual
CLDEFINE 00160 storage, where K represents 1024 bytes.

CLDEFINE 00163 Responses

CLDEFINE 00164 Responses are generated for the user to confirm that the
CLDEFINE 00166 desired configuration change has taken place.

CLDEFINE 00168 type vaddr DEFINED

CLDEFINE 00170 STORAGE = nnnnnK

CLDEFINE 00174 In the first response, 'type' is one of the following:

CLDEFINE 00177 DASD Direct access storage device
CLDEFINE 00178 TAPE Magnetic tape
CLDEFINE 00179 LINE Communication line
CLDEFINE 00180 RDR Card reader
CLDEFINE 00181 PRT Line printer
CLDEFINE 00182 PUN Card punch
CLDEFINE 00183 GRAF Graphics device
CLDEFINE 00184 CONS Console
CLDEFINE 00185 CTCA Channel-to-channel adapter
CLDEFINE 00186 DEV Any other device

CLDETACH 00004 **DETACH**

CLDETACH 00007 **Privilege Class:** B,G

CLDETACH 00009 The DETACH command is used to remove a virtual device from
CLDETACH 00010 the virtual machine or to remove a real device from the CP
CLDETACH 00011 system. A previously attached device can be detached from a
CLDETACH 00013 user even if the device is currently in use. The user may
CLDETACH 00014 also detach devices that were attached via user directory
CLDETACH 00016 entries or from CP system initialization. Devices attached
CLDETACH 00017 to the system and currently in use for spooling, paging, or
CLDETACH 00018 users minidisks cannot be detached. When the user detaches
CLDETACH 00019 a virtual device, it becomes inaccessible to the virtual
CLDETACH 00020 machine. If the device had previously been attached to the
CLDETACH 00021 user by an ATTACH command, it is released and becomes
CLDETACH 00021 available for attachment to the same user, another user, or
CLDETACH 00022 the CP system. Tape devices are automatically rewound and
CLDETACH 00023 unloaded when detached. For devices that were attached by
CLDETACH 00024 an ATTACH command and then detached by the user, a message
CLDETACH 00025 is sent to the operator informing him that the device is
CLDETACH 00026 free.

CLDETACH 00037 CLASS B / CLASS G
CLDETACH 00038
CLDETACH 00039
CLDETACH 00040
CLDETACH 00041

CLASS B

DETACH		{userid}
DET	raddr [FROM]	{SYSTEM}

CLASS G

DETACH	vaddr
DET	

CLDETACH 00047 raddr — is the real address (ccu) of the device to be
CLDETACH 00048 detached. Only class B users may use this form of
CLDETACH 00049 the command.

CLDETACH 00052 FROM — is an optional reserved keyword. If this word is
CLDETACH 00053 F — omitted and userid is specified, userid may not be
CLDETACH 00055 FROM or any of its truncations.

CLDETACH 00057 userid — specifies the user identification of the virtual
CLDETACH 00058 machine to which the real device (raddr) is
CLDETACH 00059 currently attached.

CLDETACH 00062 SYSTEM — indicates that the device is a DASD device in use
CLDETACH 00063 by CP for allocation and control.

CLDETACH 00065 vaddr — specifies the virtual address (ccu) of the device
CLDETACH 00067 to be detached from the virtual machine.

CLDETACH 00071 **Responses**

CLDETACH 00073 Messages are sent to the user, the operator who issued the

command, and the primary system operator (if different from the operator who issued the command), notifying them that the DETACH was successful.

type vaddr DETACHED

is the response to the user when he detaches one of his own devices.

type vaddr DETACHED BY operator

is the response to the user if an operator detaches one of his devices.

type raddr DETACHED [userid|SYSTEM]

is the response to the operator if the user detaches a previously attached device or if the operator detaches it from a user or the system.

type raddr DETACHED [userid|SYSTEM] BY operator

is the response to the primary system operator if he did not issue the DETACH command and the device had been previously attached.

In the above responses, 'type' is one of the following:

DASD	Direct access storage device
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel-to-channel adapter
DEV	Any other device

DETACH CHANNEL

Privilege Class: B

The DETACH CHANNEL command detaches a previously dedicated channel from a user. The virtual machine must have all I/O stopped before the function is invoked.

```
┌──────────┬──────────────────────────────┐
│  DETACH  │  CHANNEL c [FROM] userid     │
│          │                              │
└──────────┴──────────────────────────────┘
```

CHANNEL is a required reserved keyword.

c is the real address of the channel which is to be detached.

FROM
F is an optional reserved keyword. If this word is omitted, userid may not be FROM or any of its truncations.

userid specifies the user identification of the virtual machine from whom the channel is to be detached.

Responses

CHANNEL c DETACHED BY operator

is sent to the user from whose virtual machine the channel is being detached.

CHANNEL c DETACHED userid

is sent to the operator issuing the command.

CHANNEL c DETACHED userid BY operator

is sent to the primary system operator if he did not issue the command.

CLDIAL 00003 **DIAL**

CLDIAL 00006 **Privilege Class:** ALL

CLDIAL 00007 The DIAL command is used to connect a user's terminal to a
CLDIAL 00009 multi-access virtual machine. The system being dialed into
CLDIAL 00010 must be logged in with virtual communication lines available
CLDIAL 00011 and enabled before a connection can be made. The CP system
CLDIAL 00012 matches the terminal type with an equivalent virtual line.
CLDIAL 00013 Once a connection is made, the terminal is under exclusive
CLDIAL 00014 control of the virtual machine it is dialed into. It
CLDIAL 00015 remains this way until the virtual machine issues a DISABLE
CLDIAL 00017 command to the line, detaches the line, or logs out; then
CLDIAL 00019 the user is free to Login to VM/370 or dial another user.
CLDIAL 00020 Refer to the IBM Virtual Machine Facility/370, Terminal
CLDIAL 00021 User's Guide (Order No. GC20-1810) for a description of
CLDIAL 00022 multi-access virtual machines.

CLDIAL 00031
CLDIAL 00032 | DIAL | userid [vaddr] |
CLDIAL 00033 | D | |
CLDIAL 00034

CLDIAL 00040 userid specifies the userid of a multi-access virtual
CLDIAL 00042 machine currently logged in.

CLDIAL 00044 vaddr specifies the address (ccu) of the virtual
CLDIAL 00045 communication line to which the terminal is to be
CLDIAL 00046 connected. If not specified, CP will match the
CLDIAL 00047 user's terminal type to the first available
CLDIAL 00048 communication line of similar type defined for the
CLDIAL 00049 virtual machine.

CLDIAL 00052 **Note:** The DIAL command is only accepted as an alternate to
CLDIAL 00052 the LOGIN command. It cannot be used once a user has issued
CLDIAL 00053 a CP LOGIN command. The user must issue a LOGOUT command
CLDIAL 00054 after logging in to issue the DIAL command.

CLDIAL 00056 **Responses**

CLDIAL 00058 DIALED TO userid vaddr

CLDIAL 00059 is the message sent to the user indicating that a connection
CLDIAL 00060 has been made.

CLDIAL 00062 LINE raddr DIALED TO userid DIALED = nnn

CLDIAL 00064 is the response to the primary system operator. It indicates

```
CLDIAL   00065    a successful connection to the  virtual machine (userid) and
CLDIAL   00065    the total number  of VM/370 lines (nnn)  currently connected
CLDIAL   00067    to other virtual machines.

CLDIAL   00069    Note: The terminal will remain connected tc and under
CLDIAL   00070    control of the virtual machine  dialed to until that virtual
CLDIAL   00071    machine terminates the communication.  At that time the user
CLDIAL   00072    will receive the following message:

CLDIAL   00074         DROP FROM userid vaddr

CLDIAL   00075    this  message will  be sent  to the  user when  the line  is
CLDIAL   00076    disabled.


CLDIAL   00078    LINE raddr DROP FROM userid DIALED=nnn

CLDISABL 00001    is the message sent to the Primary System Operator
```

DISABLE

Privilege Class: A,B

The DISABLE command is used to prevent low-speed communication lines from accessing the VM/370 system. If the line or lines selected are not active (not dialed or logged in), the line is disabled immediately. If the selected line is active, it is disabled when the user has completed his terminal session and does a LOGOUT. An ENABLE command issued to the line before the line is physically disabled reinstates the enabled status, thus negating the DISABLE command.

DISABLE DISA	{raddr... } {ALL }

raddr specifies the addresses (ccu) of any communication lines the operator wishes to disable.

ALL specifies that all communication lines are to be disabled.

Response

COMMAND COMPLETE

CLDISCON 00004 **DISCONN**

CLDISCON 00007 **Privilege Class:** ALL

CLDISCON 00009 The DISCONN command disconnects the user's terminal from the
CLDISCON 00010 VM/370 system, but allows the virtual machine to continue
CLDISCON 00011 operation. When DISCONN is issued from the virtual machine,
CLDISCON 00012 a disconnect-time message is printed at the user's terminal
CLDISCON 00013 and at the primary system operator's console. The terminal
CLDISCON 00014 remains disconnected until the user reconnects via a LOGIN
CLDISCON 00016 command. The virtual machine is logged out if an attempt is
CLDISCON 00016 made to read from the terminal or if the virtual machine
CLDISCON 00017 goes into a disabled wait state. During the disconnect
CLDISCON 00018 time, no messages to the virtual console are saved. When
CLDISCON 00019 the user reconnects via the normal LOGIN procedure, the
CLDISCON 00021 terminal is placed in CP console function mode. To resume
CLDISCON 00021 execution of the virtual machine, the terminal user must
CLDISCON 00023 enter the BEGIN command.

CLDISCON 00032
CLDISCON 00033 | DISCONN | [HOLD] |
CLDISCON 00034 | DISC | |
CLDISCON 00035

CLDISCON 00042 HOLD specifies that the communication line is not to be
CLDISCON 00044 HO disabled. This option is effective for switched
CLDISCON 00045 lines only. If specified, control will return to
CLDISCON 00046 CP and the "VM/370 online" message will type.

CLDISCON 00049 **Responses**

CLDISCON 00050 The user receives a DISCONNECT time message when the command
CLDISCON 00052 is issued.

CLDISCON 00054 DISCONNECT AT hh:mm:ss zone weekday mm/dd/yy

CLDISCON 00056 is the response to the user who issued the command.

CLDISCON 00058 LINE raddr DSCONNECT userid USERS = nnn

CLDISCON 00060 is the response to the primary system operator informing him
CLDISCON 00060 that the user respresented by "userid" has been disconnected
CLDISCON 00061 from the VM/370 system. "nnn" is the number of users
CLDISCON 00063 remaining in the system.

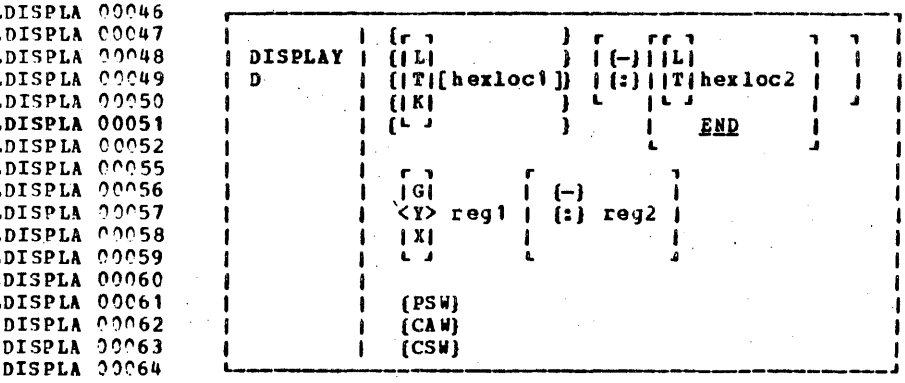
DISPLAY

Privilege Class: G

The DISPLAY command makes available to the user the contents of selected components of his virtual machine. He may display the contents of:

- virtual storage locations
- general registers
- floating-point registers
- extended control registers
- program status word (PSW)
- channel address word (CAW)
- channel status word (CSW)

If an invalid argument is entered, the DISPLAY command terminates; however, any previous valid arguments are processed before termination occurs. The operands for this command may be combined in any order desired, separated by one or more blanks.



L specifies that the contents of the specified storage locations are to be displayed in hexadecimal.

T specifies that, in addition to the hexadecimal

```
CLDISPLA 00075                 display of the specified storage locations, an
CLDISPLA 00076                 EBCDIC translation of the contents is to be
CLDISPLA 00077                 displayed.

CLDISPLA 00080     K           specifies that the storage key of the specified
CLDISPLA 00081                 storage locations is to be displayed.

CLDISPLA 00084     hexloc1     specifies the first or only storage location to be
CLDISPLA 00084                 displayed. If this field is not specified; L, T,
CLDISPLA 00085                 or K must be specified; the display will begin at
CLDISPLA 00086                 storage location 0. If hexloc1 is specified and
CLDISPLA 00086                 L, K, or T is not specified, the display will be
CLDISPLA 00088                 the same as if L were specified.

CLDISPLA 00092     -           are delimiters used to indicate range
CLDISPLA 00096     :           specifications. Either delimiter may be
CLDISPLA 00098                 surrounded by none, one, or more blanks.

CLDISPLA 00102     hexloc2     specifies that a range of storage locations from
CLDISPLA 00106     END         hexloc1 to and including hexloc2 are to be
CLDISPLA 00107                 displayed. If END is specified, all storage
CLDISPLA 00107                 locations from hexloc1 to the end of virtual
CLDISPLA 00108                 storage are displayed. If Lhexloc1 is specified,
CLDISPLA 00109                 Lhexloc2 may be used; if Thexloc1 is specified,
CLDISPLA 00110                 Thexloc2 may be used. However, the first argument
CLDISPLA 00110                 determines the display format so the L or T
CLDISPLA 00111                 preceding hexloc2 is not necessary.

CLDISPLA 00114     G           specifies that general registers are to be
CLDISPLA 00114                 displayed, where reg1, if specified, is specified
CLDISPLA 00115                 as a decimal number from 0-15 or a hexadecimal
CLDISPLA 00116                 digit from 0-F. If register is not specified
CLDISPLA 00116                 (that is, DISPLAY G is specified), all of the
CLDISPLA 00118                 user's general registers are displayed.

CLDISPLA 00121     Y           specifies that floating-point registers are to be
CLDISPLA 00121                 displayed where reg1 is a number from 0-6. If
CLDISPLA 00122                 reg1 is odd, it is adjusted to the preceding even
CLDISPLA 00123                 number. If reg1 is not specified, (that is,
CLDISPLA 00124                 DISPLAY Y is specified), all of the user's
CLDISPLA 00126                 floating-point registers are displayed.

CLDISPLA 00129     X           specifies that the extended control registers are
CLDISPLA 00129                 to be displayed, where reg1, if specified, is
CLDISPLA 00130                 either a decimal number from 0-15 or a hexadecimal
CLDISPLA 00130                 digit from 0-F. If reg1 is not specified, (that
CLDISPLA 00131                 is, DISPLAY X is specified) all of the user's
CLDISPLA 00132                 control registers are displayed. If the user does
CLDISPLA 00133                 not have extended mode operations available, only
CLDISPLA 00134                 control register 0 is displayed.

CLDISPLA 00136     reg1        specifies the first or only register to be
CLDISPLA 00137                 displayed.
```

reg2		specifies that a range of registers is to be displayed beginning with reg1 and ending with reg2.
PSW P		displays the current virtual machine PSW (program status word) as two hexadecimal words.
CAW		displays as one hexadecimal word the contents of hexadecimal location 48 (channel address word).
CSW		displays as two hexadecimal words the contents of the channel status word (doubleword at hexadecimal location 40).

Responses

One or more of the following responses is typed, depending upon the operands specified.

LOCATIONS:

xxxxxx = word1 word2 word3 word4 *EBCDIC TRANSLATION*

Where xxxxxx is the storage location of word1. Word1 is displayed (word-aligned) for a single hexloc specification. Up to 4 words are displayed on a line followed, optionally, by an EBCDIC translation of those 4 words. Multiple lines are used if required for a range of locations. If translation to EBCDIC is requested (Thexloc), alignment is made to the next lower sixteen-byte boundary; otherwise alignment is made to the next lower fullword boundary.

KEYS:

xxxxxx TO xxxxxx KEY = xx

GREG:

GPR 0 = reg0 reg1 reg2 reg3

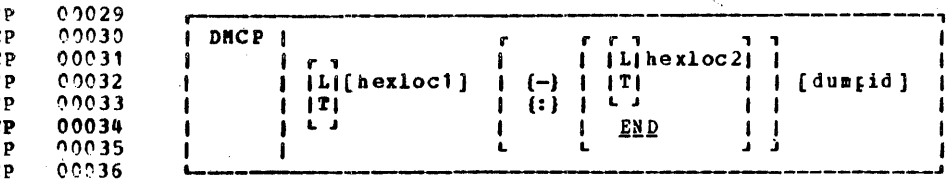
Up to 4 registers per line are displayed for a range of registers. Multiple lines are printed if required, with a maximum of four lines needed to display all sixteen general registers.

```
CLDISPLA 00193    YREGS:

CLDISPLA 00195         FPR 0 = xxxxxxxxxxxxxxxx  .xxxxxxxxxxxxxxxx  E xx

CLDISPLA 00198         The contents of the requested floating-point registers
CLDISPLA 00199         are displayed in both the internal hexadecimal format
CLDISPLA 00200         and the E format. One register per line is typed.
CLDISPLA 00203         Multiple lines are typed for a range of registers.


CLDISPLA 00206    XREGS:

CLDISPLA 00208         ECR 0 = reg0 reg1 reg2 reg3

CLDISPLA 00209         The contents of the requested extended control register
CLDISPLA 00210         are typed in hexadecimal. Up to 4 regisers per line are
CLDISPLA 00212         typed. Multiple lines are typed if required.


CLDISPLA 00215    PSW:

CLDISPLA 00217         PSW = xxxxxxxx  xxxxxxxx

CLDISPLA 00219         The contents of the PSW are typed in hexadecimal.


CLDISPLA 00222    CAW:

CLDISPLA 00224         CAW = xxxxxxxx

CLDISPLA 00225         The contents of the CAW (location 48) are typed in
CLDISPLA 00226         hexadecimal.


CLDISPLA 00229    CSW:

CLDISPLA 00231         CSW = xxxxxxxx  xxxxxxxx

CLDISPLA 00232         The contents of the CSW (location 40) are typed in
CLDISPLA 00233         hexadecimal.

CLDISPLA 00236    To terminate this function while data is typing at the
CLDISPLA 00237    terminal, the ATTN key (or its equivalent) may be pressed.
CLDMCP   00001    Typing terminates and another command may be entered.
```

CLDMCP 00003 **DMCP**

CLDMCP 00006 **Privilege Class:** E

CLDMCP 00008 The DMCP command prints the contents of real storage
CLDMCP 00010 locations on the user's virtual spooled printer. The output
CLDMCP 00011 format is eight words per line with EBCDIC translation. If
CLDMCP 00012 a descriptive header line (dumpid) is desired as the first
CLDMCP 00014 line of dump output, the user may enter as many characters
CLDMCP 00015 as will fit on the remainder of the input line. To get the
CLDMCP 00017 output printed on the real printer, the virtual spooled
CLDMCP 00018 printer must be terminated with a CLOSE command.

CLDMCP 00029
CLDMCP 00030 
CLDMCP 00031
CLDMCP 00032
CLDMCP 00033
CLDMCP 00034
CLDMCP 00035
CLDMCP 00036

CLDMCP 00044 L
CLDMCP 00048 T — either of these constants may be specified preceding hexloc1 or hexloc2 in which case they have no meaning. If either is specified in the first argument and hexloc1 is not specified dumping will begin at storage location 0. They may not be specified following the delimiter if hexloc2 is not specified.

CLDMCP 00054 **hexloc1** specifies the first or only storage location to be
CLDMCP 00055 dumped. If this field is not specified, L or T
CLDMCP 00056 must be specified.

CLDMCP 00059 -
CLDMCP 00061 : — either of these delimiters may be used to indicate a range of addresses to be dumped. The delimiter may be surrounded by none, one, or more blanks.

CLDMCP 00065 **hexloc2**
CLDMCP 00067 **END** indicates a range of real storage locations to be dumped. To dump to the end of real storage, hexloc2 may be specified as END or not specified at all, in which case END is assumed by default.
CLDMCP 00068
CLDMCP 00069

CLDMCP 00071 **dumpid** is specified for identification purposes. If text
CLDMCP 00072 is entered, it becomes the first line printed on
CLDMCP 00073 the output. As many characters as will fit on the
CLDMCP 00074 input line may be specified. If dumpid is
CLDMCP 00075 specified, hexloc2 may not be omitted if the
CLDMCP 00075 contents of a range of addresses are to be
CLDMCP 00076 printed.

```
CLDHCP   00080    Response

CLDRAIN  00001    COMMAND COMPLETE
```

DRAIN

Privilege Class: D

The DRAIN command stops spooling operations on the specified real unit record devices after the file currently being processed has completed. This command is used by the spooling operator to bring the spooling system or a specified device to a controlled halt, or to halt the activities on a device whose spooling status is to be changed. For example, all unit record devices are usually drained before system shutdown, and a printer must be drained before the contents of the UCS printer buffer are changed.

A device is drained immediately if it is not active when the DRAIN command is issued; otherwise, it is drained when processing of the current file has completed. A drained device can be restarted by the START command. If START is issued to a device before draining is complete, the device does not enter the drained status but continues processing.

```
┌───────────┬──────────────────────┐
│           │ ┌                 ┐  │
│  DRAIN    │ │READER           │  │
│  DR       │ │PRINTER          │  │
│           │ │PUNCH            │  │
│           │ │raddr...         │  │
│           │ │ALL              │  │
│           │ └                 ┘  │
└───────────┴──────────────────────┘
```

READER
R
RDR — specifies the type of device to be drained.

PRINTER
P
PRT

PUNCH
PU
PCH

raddr — specifies the addresses (ccu) of real spooled devices that are to be drained. Multiple addresses may be specified.

ALL — All spooled unit record devices in the system are to be drained.

CLDRAIN 00077 Responses

CLDRAIN 00081 type raddr SPOOL CLS [a...] DRAINED

CLDRAIN 00083 This response is given for each device which enters the
CLDRAIN 00084 drained status;

CLDRAIN 00087 type - one of the following:

CLDRAIN 00090 RDR
CLDRAIN 00092 PRT
CLDRAIN 00094 PUN

CLDRAIN 00098 a - identifies the output classes which the drained
CLDRAIN 00099 device was servicing. This field does not print for
CLDRAIN 00100 RDR devices. From one to four classes will print for
CLDRAIN 00101 PRT or PUN devices.

CLDUMP 00003 **DUMP**

CLDUMP 00006 **Privilege Class:** G

CLDUMP 00008 The DUMP command causes the contents of various components
CLDUMP 00008 of the virtual machine to be printed on the virtual spooled
CLDUMP 00010 printer. The user may print the contents of the following:

CLDUMP 00014 • virtual program status word (PSW)

CLDUMP 00016 • general registers

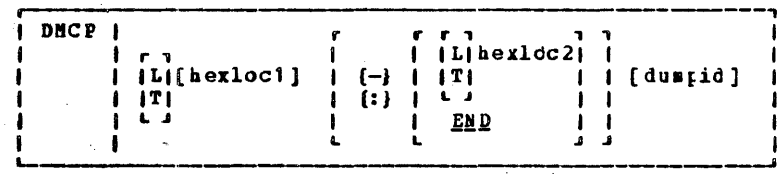
CLDUMP 00018 • floating-point registers

CLDUMP 00020 • extended control registers (optional)

CLDUMP 00022 • storage keys

CLDUMP 00024 • virtual storage locations

CLDUMP 00026 The DUMP command prints the virtual PSW and the virtual
CLDUMP 00027 registers (general, floating-point, and control). If only
CLDUMP 00028 this information is desired, at least one virtual address
CLDUMP 00030 must be specified, such as:

CLDUMP 00032 ```
DUMP 0
```

CLDUMP 00033 The output format for the virtual storage locations is 8
CLDUMP 00034 words per line with EBCDIC translation on the right. Each
CLDUMP 00035 fullword consists of 8 hexadecimal characters. All the rest
CLDUMP 00036 of the information (PSW, general registers, etc.) is printed
CLDUMP 00037 in hexadecimal. If the virtual machine is in virtual EC
CLDUMP 00038 (extended control) mode, the control registers are also
CLDUMP 00040 printed. To print the dump on the real printer, a CLOSE
CLDUMP 00041 must be issued to the spooled virtual printer.

CLDUMP 00052
CLDUMP 00053 

CLDUMP 00059

CLDUMP 00065 L, T — either of these constants may be specified preceding hexloc1 or hexloc2 in which case they have no meaning. If either is specified in the first argument and hexloc1 is not specified dumping will begin at storage location 0. They may not be specified following the delimiter if hexloc2 is not specified.
```

```
CLDUMP   00075     hexloc1    specifies the first or only storage location to be
CLDUMP   00076                dumped. If this field is not specified, L or T
CLDUMP   00077                must be specified.

CLDUMP   00080     -          either of these delimiters may be used to specify
CLDUMP   00082     :          that a range of addresses is to be dumped. The
CLDUMP   00082                delimiters may be surrounded by none, one, or more
CLDUMP   00083                blanks.

CLDUMP   00086     hexloc2    indicates that a range of addresses is to be
CLDUMP   00088     END        printed. To dump to the end of virtual storage,
CLDUMP   00089                hexloc2 may be specified as END or not specified
CLDUMP   00090                at all, in which case END is assumed by default.

CLDUMP   00093     dumpid     may be entered for descriptive purposes. If
CLDUMP   00094                specified, it becomes the first line printed on
CLDUMP   00095                the dump. As many characters as will fit on the
CLDUMP   00097                DUMP command input line may be specified. If
CLDUMP   00098                dumpid is specified, hexloc2 may not be omitted if
CLDUMP   00098                the contents of a range of addresses are to be
CLDUMP   00099                printed.
```

CLDUMP 00102 **Response**

CLECHO 00001 COMMAND COMPLETE

CLECHO 00003 ECHO

CLECHO 00006 Privilege Class: G

CLECHO 00008 The ECHO command places the user's terminal in the ECHO
CLECHO 00008 environment. When a user is in the ECHO environment, any
CLECHO 00009 input line he enters is transmitted unchanged back to his
CLECHO 00011 terminal a specified number of times. Should the user wish
CLECHO 00011 to terminate this transmission to enter a different data
CLECHO 00012 line, he must press the ATTN key (or its equivalent). When
CLECHO 00013 the specified number of lines has been typed or the
CLECHO 00014 Attention key has been hit, another read to the terminal is
CLECHO 00015 issued to accept another data line. Note that no
CLECHO 00015 line-editing is done; thus, the output line is the same as
CLECHO 00016 the input line and may contain any of the logical line
CLECHO 00017 editing characters.

CLECHO 00024 ┌──────────────┐
CLECHO 00025 │ │ ┌ ┐ │
CLECHO 00026 │ ECHO │ │nn│ │
CLECHO 00027 │ EC │ │1 │ │
CLECHO 00028 │ │ └ ┘ │
CLECHO 00029 └──────────────┘

CLECHO 00034 nn Specifies the number of times the line is to be
CLECHO 00035 sent. The default is 1. An invalid entry (that
CLECHO 00036 is, one that is greater than 99 or contains
CLECHO 00037 non-numeric characters) is treated as 1.

CLECHO 00040 Responses

CLECHO 00042 ECHO ENTERED; TO TERMINATE TEST, TYPE END

CLECHO 00044 This message is printed after the ECHO command is invoked to
CLECHO 00045 inform the user that he is in the ECHO environment.

CLECHO 00047 ENTER LINE

CLECHO 00049 This message is typed to indicate that the user should enter
CLECHO 00049 a line to be transmitted back to his terminal. If he
CLECHO 00050 replies "END", his terminal returns to the CP console
CLENABLE 00001 function mode.

CLENABLE 00003 ENABLE

CLENABLE 00006 Privilege Class: A,B

CLENABLE 00007 The ENABLE command allows the operator to enable previously
CLENABLE 00009 disabled or non-enabled communication lines so users may
CLENABLE 00010 access the VM/370 system. Previously enabled lines are not
CLENABLE 00011 affected by this command. This command affects only low
CLENABLE 00012 speed communication lines (not greater than 600 bps (bits
CLENABLE 00013 per second)).

CLENABLE 00022
CLENABLE 00023 | ENABLE | {raddr...} |
CLENABLE 00024 | EN | {ALL } |
CLENABLE 00025

CLENABLE 00031 raddr specifies the addresses (ccu) of the lines that
CLENABLE 00032 are to be enabled. If more than one raddr is
CLENABLE 00033 specified, each must be separated from the others
CLENABLE 00034 by one or more blanks.

CLENABLE 00037 ALL specifies that all previously disabled or
CLENABLE 00038 non-enabled lines are to be enabled.

CLENABLE 00041 Response

CLEXTERN 00001 COMMAND COMPLETE

EXTERNAL

Privilege Class: G

The EXTERNAL command simulates an external interrupt to the virtual machine and return control to that machine. This simulates pressing the interrupt key on the real computer console, or other functions which cause an external interrupt. Control is given to the virtual machine immediately.

EXTERNAL EXT	[code]

code   is a hexadecimal number which is to become the interrupt code associated with the external interrupt. Valid codes are 1005 (CPU Timer), 1004 (Clock Comparator), and all codes less than or equal to X'FF'.

Responses

None. Since control is given to the virtual machine, any response will be from virtual machine processing.

CLFLUSH 00003 **FLUSH**

CLFLUSH 00006 **Privilege Class:** D

CLFLUSH 00008 The FLUSH command halts and immediately purges or,
CLFLUSH 00009 optionally, holds the current output on a specified real
CLFLUSH 00010 unit record device. The device will resume activity with the
CLFLUSH 00011 next scheduled spool file.

CLFLUSH 00020
CLFLUSH 00021
CLFLUSH 00022
CLFLUSH 00023

FLUSH F	raddr [ALL] [HOLD]

CLFLUSH 00029 raddr is the address (ccu) of the real unit record
CLFLUSH 00031 output device whose activity is to be terminated
CLFLUSH 00032 (that is, the real printer or punch).

CLFLUSH 00035 ALL if more than one copy of the current output file
CLFLUSH 00035 is being printed and this option is specified, all
CLFLUSH 00036 copies are deleted; if ALL is not specified, only
CLFLUSH 00037 the current copy is deleted and the next copy, if
CLFLUSH 00038 any, is printed or punched.

CLFLUSH 00042 HOLD if this option is specified, the terminated spool
CLFLUSH 00044 HO file is not purged, but is put in a system hold
CLFLUSH 00045 status and can be reset using the CHANGE command.

CLFLUSH 00050 **Responses**

CLFLUSH 00052 type raddr FLUSHED userid FILE = spoolid

CLFLUSH 00054 type either PRT or PUN

CLFLUSH 00056 raddr the real address of the printer or punch to be
CLFLUSH 00057 affected.

CLFLUSH 00059 userid the identification of the user who is to receive
CLFLUSH 00060 the output.

CLFLUSH 00062 spoolid a unique number which identifies the file to the
CLFLUSH 00063 VM/370 system.

CLFORCE 00008 **FORCE**

CLFORCE 00011 **Privilege Class:** A

CLFORCE 00012 The FORCE command allows the system operator to force a
CLFORCE 00014 LOGOUT of any other user on the system.

CLFORCE 00018
CLFORCE 00019 | FORCE | userid [HOLD] |
CLFORCE 00020
CLFORCE 00021

CLFORCE 00026 userid specifies the user identification of the user who
CLFORCE 00028 is to be logged out.

CLFORCE 00031 HOLD specifies that the communication line is to remain
CLFORCE 00032 HO connected to VM/370 once the user has been logged
CLFORCE 00033 out.

CLFORCE 00036 **Responses**

CLFORCE 00037 The user receives the normal accounting message produced at
CLFORCE 00039 LOGOUT and a logout message of the following format:

CLFORCE 00041 LOGOUT AT hh:mm:ss zone weekday mm/dd/yy BY operator

CLFORCE 00044 where "operator" is the identification of the system
CLFORCE 00045 operator who issued the FORCE command.

CLFORCE 00047 The primary system operator receives the following message:

CLFORCE 00049 LINE raddr LOGOUT AS userid USERS = nnn BY operator

CLFORCE 00052 Where raddr is the line address, userid is the
CLFORCE 00053 identification of the user who was logged out, nnn is the
CLFORCE 00054 number of users remaining on the system, and operator is the
CLFREE 00001 system operator who issued the FORCE command.

FREE

Privilege Class: D

The FREE command removes a set of spool files belonging to a specified user from a system hold status. A spool file held by a spooling operator can only be freed by a spooling operator; a spool file held by a user can be freed by that user or the spooling operator. A spool file can be in a double hold status and can only be processed after both holds (system and user) are removed.

userid specifies the user whose spool files are to be released from a system hold status. The user need not be logged on, but his files may have previously been held by the system HOLD command.

PRINTER specifies the type of files to be freed. If ALL is
P specified, all files for the specified user are
PRT freed.

PUNCH
PU
PCH

ALL

Responses

None.

CLHALT 00003 **HALT**

CLHALT 00006 **Privilege Class:** A

CLHALT 00007 The HALT command allows the system operator to terminate any
CLHALT 00009 active channel program on a specified real device.

CLHALT 00016 ┌──────────────┐
CLHALT 00017 | HALT | raddr |
CLHALT 00018 └──────────────┘

CLHALT 00024 raddr is the address (ccu) of the real device which the
CLHALT 00026 system operator wishes to stop.

CLHALT 00028 **Note:** This command should be used only in extreme cases and
CLHALT 00029 after careful consideration. Indiscriminate use of this
CLHALT 00031 command may cause unpredictable results.

CLHALT 00033 **Responses**

CLHALT 00035 type raddr HALTED

CLHALT 00037 'type' is one of the following device types:

CLHALT 00040 DASD Direct access storage device
CLHALT 00041 TAPE Magnetic tape
CLHALT 00042 LINE Communication line
CLHALT 00043 RDR Card reader
CLHALT 00044 PRT Line printer
CLHALT 00045 PUN Card punch
CLHALT 00046 GRAF Graphics device
CLHALT 00047 CONS Console
CLHALT 00048 CTCA Channel-to-channel adapter
CLHALT 00049 DEV Any other device

CLHOLD 00003 **HOLD**

CLHOLD 00008 **Privilege Class:** D

CLHOLD 00009 The HOLD command is used to place user spool files in a
CLHOLD 00010 system hold status. The spooling operator has the ability to
CLHOLD 00012 HOLD the files of any user of the system.

CLHOLD 00014 The system HOLD status for a user is remembered even if the
CLHOLD 00014 user is not currently logged in. Both the user's files and
CLHOLD 00016 his HOLD status can be reset with the FREE command.

CLHOLD 00018 A spool file can be held by a user or by the system spooling
CLHOLD 00019 operator. A spool file held by a spooling operator or a
CLHOLD 00020 user can only be freed by the spooling operator or the user,
CLHOLD 00021 respectively. A spool file can be in a double hold status
CLHOLD 00022 and can only be processed if both holds (system and user)
CLHOLD 00023 are removed. Non-class D users cannot free files held by a
CLHOLD 00024 Class D user.

CLHOLD 00031
CLHOLD 00032 | HOLD | | |
CLHOLD 00033 | HO | | PRINTER |
CLHOLD 00034 | | userid | PUNCH |
CLHOLD 00035 | | | ALL |
CLHOLD 00036 | | | |
CLHOLD 00037

CLHOLD 00044 userid specifies the user whose spool files are to be
CLHOLD 00044 placed in a system hold status. The user need not
CLHOLD 00046 be logged on when the command is issued.

CLHOLD 00049 PRINTER indicates the type of file that is to be held.
CLHOLD 00051 P If ALL is specified, all files for the specified
CLHOLD 00053 PRT user are held.

CLHOLD 00055 PUNCH
CLHOLD 00057 PU
CLHOLD 00059 PCH

CLHOLD 00061 **ALL**

CLHOLD 00063 **Responses**

CLIPL 00001 None.

CLIPL 00004 **IPL**

CLIPL 00009 **Privilege Class:** G

CLIPL 00011 The IPL command simulates an initial program load function
CLIPL 00012 for a virtual machine. IPL simulates the LOAD key and the
CLIPL 00012 device address switches on the real computer console. The
CLIPL 00013 specified virtual address is accessed and the required
CLIPL 00014 input/output operations are performed to retrieve the IPL
CLIPL 00015 data. When the simulated load function is complete, CP
CLIPL 00015 inititiates execution of the virtual machine by loading the
CLIPL 00017 IPL PSW which was brought in during the simulation process.

CLIPL 00028
CLIPL 00029 | IPL | { | CLEAR | } |
CLIPL 00030 | I | { | NOCLEAR | } |
CLIPL 00031 | | { vaddr [cyl-no] | | } |
CLIPL 00032 | | { | | } |
CLIPL 00033 | | { system-name | | } |
CLIPL 00034

CLIPL 00041 vaddr specifies the virtual address (cuu) of the device
CLIPL 00041 which contains the nucleus of the system to be
CLIPL 00042 loaded.

CLIPL 00045 cyl-no If this operand is specified, CP loads the IPL
CLIPL 00046 data from the specified virtual cylinder instead
CLIPL 00046 of the default, which is virtual cylinder zero.
CLIPL 00047 This operand is valid only for virtual direct
CLIPL 00048 access storage devices.

CLIPL 00051 system-name
CLIPL 00053 specifies that a copy of the named system has been
CLIPL 00054 previously saved by using the CP SAVESYS command,
CLIPL 00056 and is to be brought into virtual storage and
CLIPL 00057 given control.

CLIPL 00060 CLEAR specifies that the virtual storage space is to be
CLIPL 00061 CL cleared to binary zeros before the system is
CLIPL 00062 loaded.

CLIPL 00065 **NOCLEAR** specifies that the virtual storage space is not to
CLIPL 00067 NOCL be cleared to binary zeros before the system is
CLIPL 00068 loaded.

CLIPL 00071 **Responses**

CLIPL 00072 After a successful IPL, any responses are those from the
CLLINK 00001 IPLed system.

CLLINK 00004 **LINK**

CLLINK 00007 **Privilege Class:** G

CLLINK 00009 The LINK command is used to make a device available to a
CLLINK 00010 user's virtual machine configuration, based upon information
CLLINK 00012 in the user directory.

CLLINK 00021 ┌──┐
CLLINK 00022 | LINK | [TO] userid vaddr1 [AS] vaddr2 [mode] [[PASS=] password] |
CLLINK 00023 | | |
CLLINK 00024 └──┘

CLLINK 00032 **TO** is an optional reserved word. If this word is
CLLINK 00037 **T** omitted, the userid may not be TO or T.

CLLINK 00040 **userid** specifies the name of the user whose directory is
CLLINK 00040 to be searched for device vaddr1. An asterisk (*)
CLLINK 00041 may be used to specify that the device is in the
CLLINK 00043 directory of the user issuing the LINK command.

CLLINK 00046 **vaddr1** specifies the virtual device address (ccu) to be
CLLINK 00047 given in the user directory for userid.

CLLINK 00051 **AS** is an optional reserved word. If this word is
CLLINK 00055 **A** omitted, vaddr2 may not be A.

CLLINK 00057 **vaddr2** specifies the virtual address (ccu) which is to be
CLLINK 00059 assigned to the device for this virtual machine.

CLLINK 00062 **mode** specifies the primary access requested (read/only,
CLLINK 00063 write, or multiple), and the alternate access
CLLINK 00063 (read/only or write) desired if the primary access
CLLINK 00064 is not available. Valid modes are:

CLLINK 00070 **R** specifies that read/only access is requested.
CLLINK 00070 The link is not given if any other user has
CLLINK 00071 the disk in write status. "R" is the default
CLLINK 00073 mode if the LINK is to another userid.

CLLINK 00075 **RR** specifies that read/only access is requested,
CLLINK 00076 even if another user has the disk in write
CLLINK 00077 status.

CLLINK 00079 **W** specifies that write access is requested.
CLLINK 00080 The link is not given if any other user has
CLLINK 00081 the disk in read or write status.

CLLINK 00083 **WR** specifies that write (and read) access is
CLLINK 00084 requested if no other user has the disk in
CLLINK 00085 read or write status, but that an alternate

```
CLLINK   00086                  access of read/only is acceptable if others
CLLINK   00087                  do have a link to the disk.

CLLINK   00089           M      specifies that "multiple" access is
CLLINK   00090                  requested. This means that a write-link is to
CLLINK   00091                  be given to the disk unless another user
CLLINK   00092                  already has write access to it, in which case
CLLINK   00093                  no link is to be given.

CLLINK   00095           MR     specifies that a write-link is to be given to
CLLINK   00096                  the disk unless another user already has
CLLINK   00097                  write access to it; in this case, a read-link
CLLINK   00098                  is to be given.

CLLINK   00100           MW     specifies that a write-link is to be given to
CLLINK   00102                  the disk in all cases.

CLLINK   00104                  Note: If the mode is omitted, the default is to R
CLLINK   00105                  if the userid is another user; if the user is
CLLINK   00106                  linking to one of his own disks, the default is
CLLINK   00107                  the "user access mode" of either R, W, or M as
CLLINK   00108                  specified in the user directory for his disks.

CLLINK   00111   PASS=          is an optional reserved word

CLLINK   00114   password       specifies a one to eight character string which
CLLINK   00115                  must match the access mode password for device
CLLINK   00116                  "vaddr1" in the directory for user "userid". This
CLLINK   00117                  should normally be specified only when the LINK is
CLLINK   00118                  executed by a virtual machine (for example, from
CLLINK   00119                  CMS), since the password is not print suppressed
CLLINK   00120                  when included with the LINK command. The password
CLLINK   00120                  cannot be one of the access mode options if the
CLLINK   00122                  default mode is to be used.
CLLINK   00123                  Note: The access mode password should not be
CLLINK   00124                  confused with a user password.

CLLINK   00127   If the user is linking to one of his own disks, no password
CLLINK   00128   is required. Also, if the LINK is to a device whose
CLLINK   00129   password is "ALL", meaning the device may be used by all
CLLINK   00130   users, the password is not required. Otherwise, if the LINK
CLLINK   00131   is to any other userid, a password for the desired device
CLLINK   00132   must be provided.
```

CLLINK 00136 Note

CLLINK 00137 The access allowed by LINK to userid vaddr1 device is
CLLINK 00138 summarized below:

CLLINK 00140	Primary access requested	R	R	R	R	R	W	W	W	W	W	W	M	M	M	M	M	M
CLLINK 00141	Alternate access (if any)					R				R		R					R	W
CLLINK 00142	Read password in directory	N	Y	Y	Y	Y												
CLLINK 00143	Write password in directory						N	Y	Y	Y	Y	Y						
CLLINK 00144	Mult. password in directory												N	Y	Y	Y	Y	Y
CLLINK 00145	Any existing links		N	R	W	W		N	R	R	W	W		N	R	W	W	W
CLLINK 00146	---																	
CLLINK 00147	Access established:	N	R	R	N	R	N	W	M	R	N	R	N	W	W	N	R	W

CLLINK 00149 where: N=no or none; R=read; W=write; M=multiple; Y=yes

CLLINK 00151 Responses

CLLINK 00154 ENTER READ PASSWORD:
CLLINK 00158 ********

CLLINK 00160 The read password should be typed in over the mask to
CLLINK 00162 obtain read access to the desired disk.

CLLINK 00165 ENTER WRITE PASSWORD:
CLLINK 00167 ********

CLLINK 00170 The write password should be typed in over the mask to
CLLINK 00171 obtain write access to the desired disk.

CLLINK 00174 ENTER MULT PASSWORD:
CLLINK 00176 ********

CLLINK 00179 The multiple password should be typed in over the mask
CLLINK 00180 to obtain write access to a disk for which other users
CLLINK 00181 may already have access.

CLLINK 00185 Note: If LINK is issued from a virtual machine with the
CLLINK 00186 password included on a command line, and the password is
CLLINK 00187 incorrect, then a count of these incorrect passwords is
CLLINK 00188 maintained by CP. If a total of ten such incorrect
CLLINK 00189 passwords have been entered, then the LINK command from a
CLLINK 00190 virtual machine is subsequently disallowed for that user for
CLLINK 00191 the remainder of the session. LINK can still be issued
CLLINK 00192 directly from the terminal (that is, in CP console function
CLLINK 00193 mode), or LINK can be reinstated as a valid command from the
CLLINK 00194 user's virtual machine by logging out and logging in again.
CLLINK 00194 (This procedure is designed to protect password security
CLLINK 00195 from repeated calls to LINK from a virtual machine with
CLLINK 00196 trial passwords.)

```
CLLINK   00198    DASD vaddr2 LINKED R/O

CLLINK   00200    indicates that a  read/only link to the given  disk has been
CLLINK   00200    established, for a LINK request with a  mode of R or RR, and
CLLINK   00201    that no  other users  are linked  to the  same disk  in read
CLLINK   00202    write mode.


CLLINK   00204    DASD vaddr2 LINKED R/W

CLLINK   00206    indicates that a read/write link to  the given disk has been
CLLINK   00206    established, for a LINK request with a mode of W, WR, M, MR,
CLLINK   00208    or MW, and that no other users are linked to the same disk.


CLLINK   00219                                                ┌                         ┐
CLLINK   00220    DASD vaddr2 LINKED R/O; R/W BY {nnn USERS}|; R/O BY {nnn USERS}|
CLLINK   00221                                   {userid   }|          {userid   }|
CLLINK   00222                                                └                         ┘

CLLINK   00226    indicates that a  read/only link to the given  disk has been
CLLINK   00227    established, for a LINK request with a mode of RR, but warns
CLLINK   00228    the user  that the disk is  in read/write use by  some users
CLLINK   00229    and possibly  in read use by  some users.  If only  one user
CLLINK   00229    has access, the  number of users (nnn USERS)  is replaced by
CLLINK   00230    userid.


CLLINK   00232    DASD vaddr2 LINKED R/W; R/O BY {nnn USERS}
CLLINK   00233                                   {userid   }

CLLINK   00234    indicates that a read/write link to  the given disk has been
CLLINK   00235    established for a LINK request with a  mode of M, MR, or MW,
CLLINK   00236    and informs the user that the  disk is also in read/only use
CLLINK   00237    by userid or by nnn users. (No other users have a read/write
CLLINK   00238    link to the disk.)


CLLINK   00240                                                ┌                         ┐
CLLINK   00241    DASD vaddr2 LINKED R/W; R/W BY {nn USERS} |; R/O BY {nnn USERS} |
CLLINK   00242                                   {userid  } |          {userid   } |
CLLINK   00243                                                └                         ┘

CLLINK   00247    indicates that a read/write link to  the given disk has been
CLLINK   00247    established for a LINK request with  a mode of MW, but warns
CLLINK   00249    the user  that the disk  is also  in read/write use  by some
CLLINK   00250    users and possibly  in read use by some users.  If only one
CLLINK   00250    user has access, the number of users (nnn USERS) is replaced
CLLINK   00251    by userid.
```

CLLOADBF 00005 **LOADBUF**

CLLOADBF 00008 **Privilege Class:** D

CLLOADBF 00009 The LOADBUF command is used to load the universal character
CLLOADBF 00010 set (UCS) buffer of a printer with a specified print train
CLLOADBF 00010 image, or to load the forms control buffer (FCB) with a
CLLOADBF 00011 specified image. This command should be used by the CLASS D
CLLOADBF 00012 operator to load the UCS or FCB buffer of a printer any time
CLLOADBF 00014 the print train is changed, or if the buffer has been
CLLOADBF 00015 causing an excessive number of parity errors. The device
CLLOADBF 00016 must be drained before issuing the LOADBUF command. After
CLLOADBF 00017 the load has been executed, the contents of the buffer are
CLLOADBF 00018 printed on the specified printer if VER is specified. The
CLLOADBF 00019 printout must correspond to the description of the specified
CLLOADBF 00021 buffer load that is printed in the SRL publication IBM 2821
CLLOADBF 00023 Control Unit Component Description (Order No. GA24-3312).

CLLOADBF 00035
CLLOADBF 00036 | LOADBUF | raddr {UCS name [FOLD] [VER]} |
CLLOADBF 00037 | | {FCB name } |
CLLOADBF 00038

CLLOADBF 00046 raddr specifies the address of the printer whose UCS or
CLLOADBF 00047 FCB buffer is to be loaded.

CLLOADBF 00050 UCS is a required reserved word
CLLOADBF 00052 FCB

CLLOADBF 00054 name is a 1 to 4 character name of the UCS or FCB image
CLLOADBF 00056 to be loaded. The supplied names are as follows:

CLLOADBF 00058 UCS - 1403

CLLOADBF 00061 AN normal AN arrangement
CLLOADBF 00062 HN normal HN arrangement
CLLOADBF 00063 PCAN preferred character set, AN
CLLOADBF 00064 PCHN preferred character set, HN
CLLOADBF 00065 QN PL/I - 60 graphics
CLLOADBF 00066 QNC PL/I - 60 graphics
CLLOADBF 00067 RN FORTRAN, COBOL commercial
CLLOADBF 00068 YN high speed alphanumeric
CLLOADBF 00069 TN text printing - 120 graphics
CLLOADBF 00070 PN PL/I printing - 60 graphics
CLLOADBF 00071 SN text printing 84 graphics

CLLOADBF 00073 UCS - 3211

CLLOADBF 00075 A11 Standard Commercial
CLLOADBF 00076 H11 Standard Scientific
CLLOADBF 00077 G11 ASCII

```
CLLOADBF 00078              P11    PL/I
CLLOADBF 00079              T11    Text Printing

CLLOADBF 00081              FCB - 3211 only

CLLOADBF 00085              There is only one name provided for an FCB image.

CLLOADBF 00087              FCB1         Space 6 lines/inch
CLLOADBF 00088                           Length of page 66 lines
```

Line Represented	Channel Skip Specification
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

CLLOADBF 00111 **Options**

FOLD / **F** — specifies that the operator wishes the first three quadrants of the UCS buffer code to be folded into the fourth quadrant. Folding should never be specified for print trains designed to print lower case characters, as these characters will not be printed if the buffer is loaded with the FOLD option.

VER / **V** — causes contents of the buffer to be printed on the specified printer for verification of the buffer loading function.

CLLOADBF 00135 **Responses**

The contents of the UCS buffer print on the specified printer if VER was specified.

LOADVFCB

Privilege Class: G

The LOADVFCB command allows a user to specify the forms control for a virtual spooled 3211 printer.

```
┌──────────────────────────────────────┐
│ LOADVFCB │ vaddr    FCB  name        │
│          │                           │
└──────────┴───────────────────────────┘
```

vaddr specifies the virtual device address (ccu) of the user's 3211 spooled printer.

FCB is a required reserved word.

name specifies a system-defined name for the 3211 FCB image which is to be the controlling virtual image.

There is only one name provided for an FCB image.

FCB1 Space 6 lines/inch
Length of page 66 lines

Line Represented	Channel Skip Specification
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

Responses

None.

CLLOCATE 00003 **LOCATE**

CLLOCATE 00006 **Privilege Class:** E

CLLOCATE 00007 The LOCATE command provides the addresses of CP control
CLLOCATE 00009 blocks associated with a particular user, a user's virtual
CLLOCATE 00010 device, or a real system device. The control blocks and
CLLOCATE 00011 their use are described in the *IBM Virtual Machine*
CLLOCATE 00012 *Facility/370, Programmer's Guide to Debugging* (Order No.
CLLOCATE 00013 GC20-1807).

CLLOCATE 00024
CLLOCATE 00025 | LOCATE | {userid [vaddr]} |
CLLOCATE 00026 | LOC | {raddr } |
CLLOCATE 00027

CLLOCATE 00034 **userid** specifies the user identification of the logged on
CLLOCATE 00035 user. The address of this user's virtual machine
CLLOCATE 00036 block (VMBLOK) will be provided.

CLLOCATE 00038 **vaddr** causes the virtual channel block (VCHBLOK),
CLLOCATE 00039 virtual control unit block (VCUBLOK), and virtual
CLLOCATE 00039 device block (VDEVBLOK) addresses associated with
CLLOCATE 00040 this virtual device address to be printed with the
CLLOCATE 00041 VMBLOK address.

CLLOCATE 00043 **raddr** causes the real channel block (RCHBLOK), real
CLLOCATE 00044 control unit block (RCUBLOK), and the real device
CLLOCATE 00045 block (RDEVBLOK) addresses associated with this
CLLOCATE 00046 real device address to be printed.

CLLOCATE 00050 **Responses**

CLLOCATE 00052 **LOCATE userid**

CLLOCATE 00054 VMBLOK = xxxxxx

CLLOCATE 00056 **LOCATE userid vaddr**

CLLOCATE 00058 VMBLOK VCHBLOK VCUBLOK VDEVBLOK
CLLOCATE 00059 xxxxxx xxxxxx xxxxxx xxxxxx

CLLOCATE 00061 **LOCATE raddr**

CLLOCATE 00063 RCHBLOK RCUBLOK RDEVBLOK
CLLOCATE 00064 xxxxxx xxxxxx xxxxxx

CLLOCK 00005 LOCK

CLLOCK 00008 Privilege Class: A

CLLOCK 00009 The LOCK command allows the system operator to bring into
CLLOCK 00010 real storage the virtual pages specified for the user and
CLLOCK 00011 lock these pages, thus excluding them from future paging
CLLOCK 00012 activity. Contiguous virtual storage pages from the first to
CLLOCK 00013 last page specified are locked in real storage and are not
CLLOCK 00014 available for page swapping until an UNLOCK is issued or the
CLLOCK 00015 user does a LOGOUT. This command would be used for
CLLOCK 00017 frequently used pages of a high activity system. It should
CLLOCK 00018 not be used indiscriminately because an excessive number of
CLLOCK 00019 locked pages can severely degrade system perfcrmance.

```
CLLOCK 00027 ┌──────────────────────────────┐
CLLOCK 00028 │ LOCK │ userid   fpage   lpage │
CLLOCK 00029 │      │                        │
CLLOCK 00030 └──────────────────────────────┘
```

CLLOCK 00035 userid specifies the user identification of a logged in
CLLOCK 00036 user.

CLLOCK 00038 fpage specifies the hexadecimal value of the first user
CLLOCK 00040 page to be brought into storage and locked.

CLLOCK 00042 lpage specifies the hexadecimal value of the last user
CLLOCK 00043 page to be brought into storage and locked. If
CLLOCK 00044 only one page is to be brought into storage, this
CLLOCK 00045 field should the same as fpage.

CLLOCK 00047 Note: for fpage and lpage, only the page numbers are
CLLOCK 00048 specified. For example, to lock USERA's virtual storage
CLLOCK 00049 locations X'12000' to X'2C000' in real storage, the
CLLOCK 00050 following command should be given:

```
CLLOCK 00052 ┌──────────────────┐
CLLOCK 00053 │ LOCK   USERA 12 2C │
CLLOCK 00054 └──────────────────┘
```

CLLOCK 00057 Response

CLLOCK 00059 COMMAND COMPLETE

CLLOGINP 00002 **LOGIN**

CLLOGINP 00005 **Privilege Class:** ALL

CLLOGINP 00009 The LOGIN command identifies a user to the VM/370 system and
CLLOGINP 00010 allows him to access that system. On a successful LOGIN, CP
CLLOGINP 00011 creates a virtual machine configuration from information
CLLOGINP 00012 contained in the user directory. The command name may not be
CLLOGINP 00013 entered using any logical line-editing characters, but the
CLLOGINP 00014 operands may use these characters.

CLLOGINP 00023 ┌──┐
CLLOGINP 00024 | LOGIN | userid [password] [MASK] [NOIPL] |
CLLOGINP 00025 | L | |
CLLOGINP 00026 | LOGON | |
CLLOGINP 00027 └──┘

CLLOGINP 00033 **userid** specifies the identifier assigned to the user in
CLLOGINP 00034 the user directory at the time he was authorized
CLLOGINP 00036 to use the VM/370 system.

CLLOGINP 00039 **password** specifies the user's password. This field may be
CLLOGINP 00039 specified if no protection (that is, masking
CLLOGINP 00040 characters) is desired.

CLLOGINP 00044 **MASK** requests the typing of masking characters (a
CLLOGINP 00047 **M** series of overprinted characters) to cover the
CLLOGINP 00048 password on terminals not equipped with the print
CLLOGINP 00049 inhibit feature. This mask is typed on the line
CLLOGINP 00050 following the prompting message from CP requesting
CLLOGINP 00051 the user to enter his password. Should he forget
CLLOGINP 00052 to do this when he types in LOGIN, he may signal a
CLLOGINP 00052 carriage return after the prompt for the password
CLLOGINP 00053 is typed and CP will type out the masking
CLLOGINP 00054 characters.

CLLOGINP 00057 **NOIPL** specifies that the IPL device or name in the
CLLOGINP 00060 **N** user's directory entry is not to be used for an
CLLOGINP 00061 automatic IPL.

CLLOGINP 00064 **Responses**

CLLOGINP 00067 ENTER PASSWORD:

CLLOGINP 00068 indicates that the userid has been accepted. The user
CLLOGINP 00069 should type in the password, or signal a carriage return if
CLLOGINP 00070 a mask is desired for the password, and MASK was not
CLLOGINP 00071 included on the command line.

```
CLLOGINP 00074    LOGMSG- hh:mm:ss mm/dd/yy

CLLOGINP 00076    indicates the time and date at which the system log message
CLLOGINP 00077    was generated or most recently revised. If the user wishes
CLLOGINP 00077    to see all of the system log messages, he must issue the CP
CLLOGINP 00078    command QUERY LOGMSG. Any lines of the log message for which
CLLOGINP 00079    the first character is an asterisk will be displayed at this
CLLOGINP 00080    point.


CLLOGINP 00085    FILES: {nnn} RDR, {nnn} PRT, {nnn} PUN
CLLOGINP 00086           {NO }       {NO }       {NO }

CLLOGINP 00088    This is omitted if all counts are zero, otherwise it
CLLOGINP 00089    indicates the number of spool files that exist for the user
CLLOGINP 00090    at LOGIN time.


CLLOGINP 00092    LOGON AT hh:mm:ss zone weekday mm/dd/yy
CLLOGINP 00094              or
CLLOGINP 00096    RECONNECTED AT hh:mm:ss zone weekday mm/dd/yy

CLLOGINP 00098    indicates the time, day of the week, and date at which the
CLLOGINP 00099    LOGON or RECONNECT is complete.


CLLOGINP 00101    LINE raddr LOGON AS userid     USERS = nnn
CLLOGINP 00103              or
CLLOGINP 00105    LINE raddr RECONNECT userid    USERS = nnn

CLLOGINP 00107    is the response to the primary system operator.
```

CLLOGOUT 00003 **LOGOUT**

CLLOGOUT 00006 **Privilege Class:** ALL

CLLOGOUT 00007 The LOGOUT command allows the user to terminate virtual
CLLOGOUT 00008 machine execution and disconnect his virtual machine from
CLLOGOUT 00009 the VM/370 system. This command causes all active spool
CLLOGOUT 00010 files to be closed, temporary disks to be relinquished,
CLLOGOUT 00010 dedicated devices to be detached, and an accounting record
CLLOGOUT 00019 to be created for the user.

CLLOGOUT 00021 ┌──────────────────┐
CLLOGOUT 00022 | LOGOUT | [HOLD] |
CLLOGOUT 00023 | LOG | |
CLLOGOUT 00024 | LOGOFF | |
CLLOGOUT 00025 └──────────────────┘

CLLOGOUT 00028 HOLD specifies that the communication line is to be
CLLOGOUT 00032 HO left connected so that the user does not have to
CLLOGOUT 00033 re-dial the VM/370 system before logging in.

CLLOGOUT 00037 **Responses**

CLLOGOUT 00038 Session times will be printed at the user's terminal when he
CLLOGOUT 00040 logs off.

CLLOGOUT 00042 CONNECT = hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU = mmm:ss.hs

CLLOGOUT 00043 is the actual clock and CPU times spent in the user's
CLLOGOUT 00044 current session or the elapsed time since the ACNT command
CLLOGOUT 00045 was last entered for the user.

CLLOGOUT 00049 CONNECT hh:mm:ss gives the actual clock time spent in the
CLLOGOUT 00049 current terminal session in
CLLOGOUT 00050 hours:minutes:seconds.

CLLOGOUT 00053 VIRTCPU mmm:ss.hs gives the virtual CPU time used in the
CLLOGOUT 00053 current terminal session in
CLLOGOUT 00054 minutes:seconds.hundredths of seconds.

CLLOGOUT 00057 TOTCPU mmm:ss.hs gives the total CPU time (including
CLLOGOUT 00057 virtual and overhead) used in the
CLLOGOUT 00057 current terminal session in
CLLOGOUT 00059 minutes:seconds.hundredths of seconds.

```
CLLOGOUT 00064     LOGOFF AT hh:mm:ss zone weekday mm/dd/yy

CLLOGOUT 00066     is the response for a logout.


CLLOGOUT 00068     LINE raddr LOGOFF AS userid USERS = nnn FORCED

CLLOGOUT 00070     is the response if the logout is forced by a line timeout or
CLLOGOUT 00071     a terminal power-off.


CLLOGOUT 00073     LINE raddr LOGOFF AS userid  USERS = nnn

CLLOGOUT 00075     is the normal response to the primary system operator.


CLLOGOUT 00077     LINE DSC  LOGOFF AS userid  USERS = nnn

CLLOGOUT 00079     is the response  to the primary system  operator when logout
CLLOGOUT 00079     occurs for a user who  had previously disconnected using the
CLLOGOUT 00080     DISCONN command.
```

CLMSG 00003 **MSG**

CLMSG 00006 **Privilege Class:** B, ALL

CLMSG 00007 The MSG command allows users to transmit message text to a
CLMSG 00008 specified userid or to the primary system operator, and to
CLMSG 00009 allow the primary system operator, to send message text to
CLMSG 00010 one or all logged in users. If the user designated to
CLMSG 00011 receive the message is not logged on or has suppressed the
CLMSG 00012 receiving and typing of messages, the message is not
CLMSG 00013 transmitted and the sender receives a diagnostic message to
CLMSG 00014 this effect. If the operator sends a message to **ALL**, he
CLMSG 00015 receives a diagnostic message for every user with message
CLMSG 00016 typing suppressed. A message which is not received by a
CLMSG 00017 user is not saved and must be sent at a later time when the
CLMSG 00018 user is receiving messages. The message will be typed out
CLMSG 00018 at the terminal when the terminal is ready to receive
CLMSG 00019 output. If the terminal is entering data, the message is
CLMSG 00021 held until a carriage return is received.

CLMSG 00030 CLASS B:
CLMSG 00031
CLMSG 00032 | MSG | {ALL } msg-text |
CLMSG 00033 | M | {userid } |
CLMSG 00034 | | {OPERATOR} |
CLMSG 00035

CLASS ALL:

| MSG | {userid } msg-text |
| M | {OPERATOR} |

CLMSG 00041 ALL is for operator use only. It allows him to
CLMSG 00042 broadcast a message to all logged in users.

CLMSG 00044 userid specifies a single user who is to receive the
CLMSG 00045 message.

CLMSG 00048 OPERATOR specifies that the message is to be sent to the
CLMSG 00052 OP primary system operator regardless of his userid.

CLMSG 00055 msg-text specifies the text of the message which is to be
CLMSG 00055 transmitted. As many characters may be entered as
CLMSG 00057 will fit on the remainder of the input line.

CLMSG 00060 **Responses**

CLMSG 00062 MSG FROM OPERATOR : message-text

CLMSG 00064 is the response to the user to whom a message from the
CLMSG 00065 system operator is directed.

```
CLMSG    00067    MSG FROM userid   :   message-text

CLMSG    00068    This is the format of the message sent to another user or to
CLMSG    00070    the system operator, where userid is the name of the sender.

CLMSG    00072    If the user receiving the message is the primary system
CLMSG    00073    operator, the alarm bell at the central computer console
CLNOTRDY 00002    will ring.
```

NOTREADY

Privilege Class: G

The NOTREADY command causes a virtual device to appear as if it had changed from ready to not ready status. This command is for use with spooled unit record devices and virtual consoles only. Any I/O operation to the specified device in progress at the time of the command will be completed. On the next STARTIO (SIO) instruction, the not ready condition will be in effect.

vaddr specifies the virtual device address (ccu).

Response

INVALID DEVICE TYPE

is the response if the device specified by vaddr is not a spooled unit record device or a virtual console.

CLORDER 00004 **ORDER**

CLORDER 00007 **Privilege Class:** D, G

CLORDER 00009 The ORDER command places closed spool files of a device type
CLORDER 00010 in a specific order. A user may determine via the QUERY
CLORDER 00011 command the filename, filetype, originating userid and other
CLORDER 00012 attributes of all of his files. For G users, the files are
CLORDER 00013 ordered as they are passed to the spool device within the
CLORDER 00014 user's domain only. Class G users may affect only their own
CLORDER 00015 files.

CLORDER 00024
CLORDER 00025 | ORDER | [ ] {READER } {CLASS c1 CLASS c2...}* |
CLORDER 00026 | ORD | |userid| {PRINTER } {spoolid1 spoolid2...} |
CLORDER 00027 | | |SYSTEM| {PUNCH } |
CLORDER 00028 | | [ ] |
CLORDER 00029

CLORDER 00034 * Ordering may be done using a combination of CLASS c
CLORDER 00035 specifications and spoolid specifications. For example:

CLORDER 00037
CLORDER 00038 | ORDER PRINTER CLASS A 1963 CLASS C |
CLORDER 00039

CLORDER 00044 specifies that printer files will be processed in the
CLORDER 00044 following order: all Class A files, the file with spoolid
CLORDER 00046 1963, all Class C files.

CLORDER 00050 userid (for use by Class D users only) this allows the
CLORDER 00050 spooling operator to manipulate the spool files
CLORDER 00051 for a particular user.

CLORDER 00054 SYSTEM (for use by Class D users only) this allows the
CLORDER 00054 spooling operator to manipulate all spool files in
CLORDER 00056 the system, regardless of userid.

CLORDER 00059 READER specifies the type of files to be ordered.
CLORDER 00061 R
CLORDER 00063 RDR

CLORDER 00065 PRINTER
CLORDER 00067 P
CLORDER 00069 PRT

CLORDER 00071 PUNCH
CLORDER 00073 PU
CLORDER 00075 PCH

CLORDER 00077 CLASS is a required reserved word which indicates
CLORDER 00079 CL that files are to be ordered according to their

```
CLORDER  00080              class.

CLORDER  00083    c         specifies class designations for  input and output
CLORDER  00084              spool files. This specification allows files to be
CLORDER  00085              placed in the order of  classes specified by CLASS
CLORDER  00086              c1, CLASS c2,  and so forth. c  is a one-character
CLORDER  00086              alphameric field with values from A  tc Z and 0 to
CLORDER  00087              9.

CLORDER  00090    spoolid   specifies  that  by  the  files  represented  the
CLORDER  00091              spoolids are to be processed in the order in which
CLORDER  00093              the spoolids are given.


CLORDER  00096    Response

CLORDER  00100    {nnnn} FILES ORDERED
CLORDER  00102    {NO  }

CLPURGE  00001    is the response to the user who issued the ORDER command.
```

```
CLPURGE  00005     PURGE

CLPURGE  00008     Privilege Class:   D, G

CLPURGE  00010     The PURGE command removes closed spool files from the system
CLPURGE  00011     before they are printed or  punched by the spooling devices,
CLPURGE  00013     or before they are read by a user. If the user does not have
CLPURGE  00014     spooling operator (Class  D) privileges, he may  affect only
CLPURGE  00015     his own files; the spooling operator, however, may purge any
CLPURGE  00016     spool file  in the  system.  Any closed  file may  be purged
CLPURGE  00017     regardless  of  its status,  as  long  as  it has  not  been
CLPURGE  00018     selected for processing.

CLPURGE  00030     ┌──────────────────────────────────────────────────────────────────────┐
CLPURGE  00031     | PURGE | ┌       ┐ {{READER }    ┌                             ┐} |
CLPURGE  00032     | PUR   | |userid| {{PRINTER}    |  ALL                        |} |
CLPURGE  00033     |       | |SYSTEM| {{PUNCH  }    |{CLASS c1 CLASS c2 ...}*     |} |
CLPURGE  00034     |       | └       ┘ {             |{spoolid1 spoolid2 ...}      |} |
CLPURGE  00035     |       |           { ALL         └                             ┘} |
CLPURGE  00036     └──────────────────────────────────────────────────────────────────────┘

CLPURGE  00041     *  Purging  may be done using  a combination of CLASS  c and
CLPURGE  00042     spoolid specifications.  For example:

CLPURGE  00044     ┌──────────────────────────────────────┐
CLPURGE  00045     | PURGE PRINTER CLASS A 1932 CLASS D 619 |
CLPURGE  00046     └──────────────────────────────────────┘

CLPURGE  00050     specifies that all Class A and  Class D files and files with
CLPURGE  00051     spoolids 1932 and 619 are to be purged.

CLPURGE  00055     userid      (for  use  by  Class  D  users  only)  allows  the
CLPURGE  00055                 spooling operator  to manipulate  the spool  files
CLPURGE  00056                 for a particular user.

CLPURGE  00059     SYSTEM      (for  use by  the Class  D user  only) allows  the
CLPURGE  00059                 spooling operator to manipulate all spool files on
CLPURGE  00061                 the system, regardless of userid.

CLPURGE  00065     READER      specifies the type of files that are to be purged.
CLPURGE  00066     R           If ALL  is  specified, files  of  all  types  are
CLPURGE  00067     RDR         purged, and  any   other  options   specified  are
CLPURGE  00069                 ignored.

CLPURGE  00071     PRINTER
CLPURGE  00073     P
CLPURGE  00074     PRT

CLPURGE  00076     PUNCH
CLPURGE  00077     PU
CLPURGE  00078     PCH
```

```
CLPURGE  00080     ALL

CLPURGE  00082     ALL         specifies that all files of the given type are  to
CLPURGE  00083                 be purged.

CLPURGE  00087     CLASS       is a required  reserved word which  indicates that
CLPURGE  00090     CL          all  files of  the  specified  classes are  to  be
CLPURGE  00091                 purged.

CLPURGE  00094     c           specifies the  classes of  files of  the indicated
CLPURGE  00095                 device  type  which  are  to be  purged.  c  is  a
CLPURGE  00095                 one-character alphameric field with  values from A
CLPURGE  00096                 to Z and from 0 to 9.

CLPURGE  00098     spoolid     specifies that  only the files represented  by the
CLPURGE  00100                 spoolids are to be purged.


CLPURGE  00104     Responses


CLPURGE  00108     (nnnn) FILES PURGED
CLPURGE  00110     (NO    )

CLQUERYP 00001     is the response to the user who issued the PURGE command.
```

CLQUERYP 00005 **QUERY**

CLQUERYP 00008 **Privilege Class:** A, B, D, E, G, ALL

CLQUERYP 00009 The QUERY command allows the user to request system status
CLQUERYP 00011 and machine configuration information. Not all operands are
CLQUERYP 00012 available in every privilege class. Operands available to
CLQUERYP 00013 the specified privilege classes are given below.

CLQUERYP 00024 CLASS A:

CLQUERYP 00027
CLQUERYP 00028 | QUERY |{ PAGING }
CLQUERYP 00029 | Q |{ PRIORITY userid }
CLQUERYP 00030

CLQUERYP 00032 PAGING displays the current system paging activity index.
CLQUERYP 00034 PAG

CLQUERYP 00036 PRIORITY displays the current priority of the specified
CLQUERYP 00040 PRIOR userid. This is established in the user directory
CLQUERYP 00040 but may be overridden by the SET PRIORITY nn
CLQUERYP 00041 command.

CLQUERYP 00045 CLASS B:

CLQUERYP 00048
CLQUERYP 00049 | QUERY |{ {DASD } }
CLQUERYP 00050 | Q |{ {TAPES } }
CLQUERYP 00051 | |{ {LINES } }
CLQUERYP 00052 | |{ {UR } }
CLQUERYP 00053 | |{ {STORAGE} }
CLQUERYP 00054 | |{ {ALL } }
CLQUERYP 00055 | |{ {raddr } }
CLQUERYP 00056 | |{ }
CLQUERYP 00057 | |{ SYSTEM raddr }
CLQUERYP 00058 | |{ DUMP }
CLQUERYP 00059

CLQUERYP 00064 DASD displays a list showing the real status of all the
CLQUERYP 00065 DA devices within the requested class. If raddr is
CLQUERYP 00067 requested, the status of just that device is
CLQUERYP 00068 TAPES displayed. A STORAGE request displays the size
CLQUERYP 00070 TA of real main storage. DASD represents disk or
CLQUERYP 00072 drum devices, TAPE represents communication
CLQUERYP 00073 LINES lines, and UR represents unit record devices
CLQUERYP 00076 (card reader, printers, card punches).
CLQUERYP 00077 UR

CLQUERYP 00079 STORAGE

```
CLQUERYP 00081    STOR

CLQUERYP 00083    ALL


CLQUERYP 00085    raddr

CLQUERYP 00089    SYSTEM raddr
CLQUERYP 00090    SYS         returns to the operator the userid, address, and
CLQUERYP 00091                access mode of virtual disks which reside on the
CLQUERYP 00092                specified raddr (ccu) belonging tc logged-cn
CLQUERYP 00093                users.

CLQUERYP 00096    DUMP        returns to the operator the type cf device and
CLQUERYP 00098    DU          device address of the ABEND dump unit.


CLQUERYP 00101    CLASS D:

CLQUERYP 00105    ┌──────────────────────────────┐
CLQUERYP 00106    | QUERY |{ FILES      ┌      ┐ } |
CLQUERYP 00107    | Q     |{{READER }  |spoolid| } |
CLQUERYP 00108    |       |{{PRINTER}  |ALL    | } |
CLQUERYP 00109    |       |{{PUNCH  }  └      ┘ } |
CLQUERYP 00110    |       |{ HOLD               } |
CLQUERYP 00111    └──────────────────────────────┘

CLQUERYP 00116    FILES       returns to the requestor the number cf spooled
CLQUERYP 00118    F           input and output files. The user receives the
CLQUERYP 00119                total count in the system. Files which are
CLQUERYP 00120                currently being processed are not included in the
CLQUERYP 00121                totals.

CLQUERYP 00125    READER      specifies that information is to be returned
CLQUERYP 00127    R           concerning spool files of a specified type. The
CLQUERYP 00130    RDR         user receives information on all spocl files in
CLQUERYP 00131                the system.
CLQUERYP 00133    PRINTER
CLQUERYP 00135    P
CLQUERYP 00137    PRT

CLQUERYP 00139    PUNCH
CLQUERYP 00141    PU
CLQUERYP 00143    PCH

CLQUERYP 00145    HOLD        returns to the operator a list cf users whcse
CLQUERYP 00148    H           output is being held by the HOLD ccmmand.

CLQUERYP 00151    spoolid     for PRINTER or PUNCH files, the number of records,
CLQUERYP 00152    ALL         filename, filetype, number of copies, output
CLQUERYP 00154                class, and HOLD status will be returned. Fcr
CLQUERYP 00154                READER files, number of records, filename,
CLQUERYP 00155                filetype, logical record length, ard userid of the
CLQUERYP 00156                file originator will be returned. If ALL is
```

CLQUERYP 00156 specified, the status of all files of the
CLQUERYP 00157 specified type is given.

CLQUERYP 00160 CLASS E:

CLQUERYP 00162 Same as Class A.

CLQUERYP 00165 CLASS G:

CLQUERYP 00168
CLQUERYP 00169 | QUERY |{ { TIME } }
CLQUERYP 00170 | Q |{ { FILES } }
CLQUERYP 00171 | |{ { SET } }
CLQUERYP 00172 | |{ { TERMINAL} }
CLQUERYP 00173 | |{ }
CLQUERYP 00174 | |{ [VIRTUAL] {DASD } }
CLQUERYP 00175 | |{ {TAPES } }
CLQUERYP 00176 | |{ {LINES } }
CLQUERYP 00177 | |{ {UR } }
CLQUERYP 00178 | |{ {STORAGE} }
CLQUERYP 00179 | |{ *{ALL } }
CLQUERYP 00180 | |{ {vaddr } }
CLQUERYP 00181 | |{ }
CLQUERYP 00182 | |{ LINKS vaddr }
CLQUERYP 00183 | |{ ┌ ┐ }
CLQUERYP 00184 | |{ { READER } |spoolid| }
CLQUERYP 00185 | |{ { PRINTER} |ALL | }
CLQUERYP 00186 | |{ { PUNCH } └ ┘ }
CLQUERYP 00187

CLQUERYP 00191 * This is the default only if the keyword VIRTUAL is
CLQUERYP 00192 entered. If VIRTUAL is omitted, one of the choices must be
CLQUERYP 00193 made.

CLQUERYP 00196 TIME returns to the requestor the connect and CPU time
CLQUERYP 00198 T spent in the current terminal session.

CLQUERYP 00200 FILES returns to the requestor the number of spooled
CLQUERYP 00203 F input and output files for his virtual machine.
CLQUERYP 00203 Files which are currently being processed are not
CLQUERYP 00204 included in the totals.

CLQUERYP 00207 SET returns to the requestor the status of the SET
CLQUERYP 00209 S command functions.

CLQUERYP 00211 TERMINAL returns to the requestor the options for terminal
CLQUERYP 00213 TERM input and output as specified in the TERMINAL
CLQUERYP 00214 command.

CLQUERYP 00217 VIRTUAL returns to the requestor the status of the virtual
CLQUERYP 00218 V devices within the class selected. If STORAGE is
CLQUERYP 00219 requested, the size of the virtual storage is

```
CLQUERYP 00220              printed.  If vaddr is specified, the status of the
CLQUERYP 00223              one device is returned.  This keyword is optional.
CLQUERYP 00223              However, if  it is not  specified, ALL is  not the
CLQUERYP 00224              default; an  error message is generated  if device
CLQUERYP 00225              type is not specified.

CLQUERYP 00227              Virtual Device Types:

CLQUERYP 00230              DASD       virtual direct access storage devices
CLQUERYP 00232              DA

CLQUERYP 00234              TAPES      virtual magnetic tape device
CLQUERYP 00236              TA

CLQUERYP 00238              LINES      virtual communication lines

CLQUERYP 00240              UR         virtual unit record devices

CLQUERYP 00242              STORAGE    virtual storage
CLQUERYP 00244              STOR

CLQUERYP 00246              ALL        return status of all virtual devices

CLQUERYP 00249    LINKS     The userid, device address,  and access mode  are
CLQUERYP 00251    L         printed at  the terminal for  all users  linked to
CLQUERYP 00252              the specified virtual address.

CLQUERYP 00256    READER    specifies  that  information  is  to be typed out
CLQUERYP 00257    R         concerning  the user's spool files.
CLQUERYP 00259    RDR

CLQUERYP 00261    PRINTER
CLQUERYP 00263    P
CLQUERYP 00265    PRT

CLQUERYP 00267    PUNCH
CLQUERYP 00269    PU
CLQUERYP 00272    PCH

CLQUERYP 00275    spoolid   for PRINTER or PUNCH files, the number of records,
CLQUERYP 00276    ALL       filename,  filetype,  number  of  copies,  output
CLQUERYP 00277              class, and  HOLD status are returned.  For READER
CLQUERYP 00278              files,  number  of  records,  filename,  filetype,
CLQUERYP 00279              logical  record length,  and  userid  of the  file
CLQUERYP 00280              originator are returned. If  ALL is specified, the
CLQUERYP 00280              status  of  all  files  of  the specified  type  is
CLQUERYP 00281              given.
```

CLQUERYP 00287 CLASS ALL:

CLQUERYP 00291
CLQUERYP 00292 | QUERY | {LOGMSG } |
CLQUERYP 00293 | Q | {NAMES } |
CLQUERYP 00294 | | {USERS [userid] } |
CLQUERYP 00295 | | {userid } |
CLQUERYP 00296

CLQUERYP 00299 LOGMSG returns to the requestor the log messages of the
CLQUERYP 00302 LOG day.

CLQUERYP 00304 NAMES returns to the requestor a list of all the users
CLQUERYP 00307 N logged in and the real address of the line to
CLQUERYP 00308 which each is connected. If the user is
CLQUERYP 00309 disconnected, DSC is printed instead of the line
CLQUERYP 00310 address.

CLQUERYP 00314 USERS returns to the requestor the number of logged in
CLQUERYP 00317 U users and the number of users dialed to other
CLQUERYP 00318 virtual machines. If userid is specified, the
CLQUERYP 00319 userid and device address of the user's terminal
CLQUERYP 00319 are printed if he is logged on. If the specified
CLQUERYP 00320 user is not logged on, a message to that effect
CLQUERYP 00321 will be printed.

CLQUERYP 00324 userid prints the userid and the device address of the
CLQUERYP 00325 user's terminal if he is logged on. If the user
CLQUERYP 00326 is not logged on, a message to this effect is
CLQUERYQ 00002 issued.

CLQUERYQ 00005 Responses

CLQUERYQ 00007 The following are formats of responses to the QUERY command.
CLQUERYQ 00008 vaddr denotes a virtual address and raddr a real address.

CLQUERYQ 00010 QUERY ALL - Class B

CLQUERYQ 00012 Produces the same results as if the following commands were
CLQUERYQ 00013 issued:

CLQUERYQ 00016 QUERY STORAGE
CLQUERYQ 00018 QUERY UR
CLQUERYQ 00020 QUERY LINES
CLQUERYQ 00022 QUERY DASD
CLQUERYQ 00024 QUERY TAPES

CLQUERYQ 00027 QUERY DASD - Class B

CLQUERYQ 00029 DASD raddr FREE

CLQUERYQ 00030 This response is given if the real device designated by
CLQUERYQ 00032 raddr is not allocated to the CP system or to any user.

CLQUERYQ 00034 DASD raddr ATTACH TO userid vaddr

CLQUERYQ 00035 This response is given if the real device specified by raddr
CLQUERYQ 00036 is attached to a user's (userid) virtual machine at virtual
CLQUERYQ 00037 address vaddr.

CLQUERYQ 00039 DASD raddr CP SYSTEM volid nnn

CLQUERYQ 00040 This response is given if the real device designated by
CLQUERYQ 00041 raddr is allocated to the system for use as user's
CLQUERYQ 00042 minidisks. nnn is the number of active user's minidisks on
CLQUERYQ 00043 the physical disk and volid is the volume serial number of
CLQUERYQ 00044 the real disk.

CLQUERYQ 00046 DASD raddr CP OWNED volid nnn

CLQUERYQ 00048 This response is given if the real device designated by
CLQUERYQ 00049 raddr is used by the system for paging and spooling
CLQUERYQ 00050 activity. nnn is the number of active user's minidisks (if
CLQUERYQ 00050 any) on the physical disk and volid is the volume serial
CLQUERYQ 00051 number of the real disk.

CLQUERYQ 00054 QUERY TAPES - Class B

CLQUERYQ 00056 TAPE raddr FREE

CLQUERYQ 00057 This response is given if the real tape device designated by
CLQUERYQ 00059 raddr is not allocated to the system or to any user.

CLQUERYQ 00061 TAPE raddr CP SYSTEM

CLQUERYQ 00062 This response is given if the real tape device designated by
CLQUERYQ 00064 raddr is attached to CP for its exclusive use.

CLQUERYQ 00066 TAPE raddr ATTACH TO userid vaddr

CLQUERYQ 00067 This response is given if the real tape device designated by
CLQUERYQ 00068 raddr is attached to a user's (userid) virtual machine at
CLQUERYQ 00069 virtual address vaddr.

```
CLQUERYQ 00079    QUERY UR - Class B

CLQUERYQ 00081
CLQUERYQ 00083    type raddr SPOOL CLS [a...] |DRAINED|
CLQUERYQ 00084                                |STARTED|
CLQUERYQ 00085
CLQUERYQ 00089    This response is given for each unit record device attached
CLQUERYQ 00090    to the system for spooling activity.

CLQUERYQ 00093    type      is one of the following: RDR
CLQUERYQ 00094                                       PRT
CLQUERYQ 00095                                       PUN

CLQUERYQ 00097    raddr     is the real device address (ccu).

CLQUERYQ 00099    a         specifies the classes serviced by the output
CLQUERYQ 00100              device. This field does not type for a RDR device.
CLQUERYQ 00101              Up to four classes may be serviced by an output
CLQUERYQ 00102              device.

CLQUERYQ 00105    DRAINED   indicates that the device is not currently
CLQUERYQ 00105              available for processing. A START command must be
CLQUERYQ 00107              issued to activate the device.

CLQUERYQ 00109    STARTED   indicates that the device is available for
CLQUERYQ 00110              spooling activity.


CLQUERYQ 00113    type raddr ATTACH TO userid vaddr

CLQUERYQ 00115    This response is given if the device is attached to a user's
CLQUERYQ 00116    virtual machine at vaddr.


CLQUERYQ 00120    QUERY LINES - Class B

CLQUERYQ 00122    CONS  raddr LOGIN AS userid

CLQUERYQ 00124    indicates that the user represented by userid is currently
CLQUERYQ 00125    logged on at the system console located at raddr.

CLQUERYQ 00127    LINE  raddr FREE

CLQUERYQ 00128    indicates that the communication line at real address raddr
CLQUERYQ 00129    is not in use.

CLQUERYQ 00131    LINE  raddr LOGIN AS userid

CLQUERYQ 00132    indicates that the user specified by userid is currently
CLQUERYQ 00133    logged on at the device connected to the communication line
CLQUERYQ 00134    located at raddr.

CLQUERYQ 00136    LINE  raddr ATTACH TO userid vaddr
```

CLQUERYQ 00137 indicates that the communication line at raddr is attached
CLQUERYQ 00138 to the virtual machine represented by userid at virtual
CLQUERYQ 00139 address vaddr.

CLQUERYQ 00142 QUERY STORAGE - Class B

CLQUERYQ 00144 STORAGE = xxxxxK

CLQUERYQ 00146 xxxxx - size of real storage in multiples of 1024 bytes.

CLQUERYQ 00149 QUERY raddr - Class B

CLQUERYQ 00151 The response to this command depends upon the type of device
CLQUERYQ 00152 located at raddr.

CLQUERYQ 00154 See QUERY DASD, TAPES, UR, and LINES responses.

CLQUERYQ 00158 QUERY SYSTEM raddr - Class B

CLQUERYQ 00159 This command requests the number of user minidisks residing
CLQUERYQ 00160 on the physical disk located at raddr. One entry of the
CLQUERYQ 00161 form:

CLQUERYQ 00163 userid mode vaddr

CLQUERYQ 00165 is given for each minidisk.

CLQUERYQ 00166 "mode" is the type of access the user has: either R (Read)
CLQUERYQ 00167 or W (Write).

CLQUERYQ 00170 userid is the identification of the user who owns the
CLQUERYQ 00171 minidisk.

CLQUERYQ 00173 vaddr is the virtual address by which the user refers to
CLQUERYQ 00174 the minidisk.

CLQUERYQ 00180 QUERY DUMP - Class D

CLQUERYQ 00182 type raddr DUMP UNIT (CP)
CLQUERYQ 00183 (ALL)

CLQUERYQ 00185 indicates that the device of device type 'type' located at
CLQUERYQ 00186 raddr is the System Dump Unit.

```
CLQUERYQ 00190    QUERY FILES - Class D

CLQUERYQ 00197
CLQUERYQ 00199    FILES: |NO | RDR, |NO | PRT, |NO | PUN
CLQUERYQ 00200           |nnn|      |nnn|      |nnn|
CLQUERYQ 00201           |   |      |   |      |   |

CLQUERYQ 00203    This response gives the total number of spool files in the
CLQUERYQ 00204    system.


CLQUERYQ 00207    QUERY READER - Class D

CLQUERYQ 00210    USERID  FILE   CLS RECDS ORIGIN   DATE     TIME     NAME      TYPE
CLQUERYQ 00212    userid1 spoolid    norec userid2 mm/dd/yy hh:mm:ss filename filetype
CLQUERYQ 00215    .                                                            .
CLQUERYQ 00217    .                                                            .
CLQUERYQ 00219    .                                                            .
CLQUERYQ 00221    Only one  file will  be listed  for a  QUERY READER  spoolid
CLQUERYQ 00222    command.

CLQUERYQ 00227    userid1   specifies the identification of the user who is to
CLQUERYQ 00228              receive the file.

CLQUERYQ 00231    spoolid   a unique, system assigned number which is used by
CLQUERYQ 00232              VM/370 to identify the file.

CLQUERYQ 00234    norec     gives the  number of logical records contained in
CLQUERYQ 00235              the file.

CLQUERYQ 00237    userid2   specifies  the  identification  of  the  user  who
CLQUERYQ 00239              created the file.

CLQUERYQ 00241    mm/dd/yy  gives  the  date  the  file  was  created  in
CLQUERYQ 00242              month/day/year.

CLQUERYQ 00244    hh:mm:ss  gives  the  time  of  file  creation  in
CLQUERYQ 00245              hours:minutes:seconds.

CLQUERYQ 00248    filename  gives the filename assigned to the file (if any).

CLQUERYQ 00251    filetype  gives the filetype assigned to the file (if any).


CLQUERYQ 00256    QUERY PRINTER - Class D

CLQUERYQ 00258    USERID  FILE   CLS RECDS CPY HOLD DATE     TIME     NAME      TYPE
CLQUERYQ 00260    userid1 spoolid a  norec  nn hold mm/dd/yy hh:mm:ss filename filetype
CLQUERYQ 00262    .                                                            .
CLQUERYQ 00264    .                                                            .
CLQUERYQ 00266    .                                                            .
CLQUERYQ 00268    Only one file will be listed for a QUERY PRINTER spoolid command.

CLQUERYQ 00272    userid1   specifies the identification of the user who is
```

```
CLQUERYQ 00273              to receive the file.

CLQUERYQ 00276    spoolid   a unique, system assigned number which is used
CLQUERYQ 00277              by VM/370 to identify the file.

CLQUERYQ 00280    a         the output class of the file.

CLQUERYQ 00283    norec     gives the number of logical records contained in the file.

CLQUERYQ 00286    nn        is the number of copies of the output file
CLQUERYQ 00287              to be produced.

CLQUERYQ 00290    hold      is the file hold status and is either:

CLQUERYQ 00292                   NONE - no hold
CLQUERYQ 00294                   USER - user hold
CLQUERYQ 00296                   SYS  - system hold
CLQUERYQ 00298                   USYS - system and user hold

CLQUERYQ 00301    mm/dd/yy  gives the date the file was created in month/day/year.

CLQUERYQ 00304    hh:mm:ss  gives the time of file creation in hours:minutes:seconds.

CLQUERYQ 00307    filename  gives the filename assigned to the file (if any).

CLQUERYQ 00310    filetype  gives the filetype assigned to the file (if any).


CLQUERYQ 00314    QUERY PUNCH - Class D

CLQUERYQ 00317    USERID  FILE   CLS RECDS CPY HOLD DATE     TIME     NAME     TYPE
CLQUERYQ 00319    userid1 spoolid a   norec  nn  hold mm/dd/yy hh:mm:ss filename filetype
CLQUERYQ 00321    .                                                              .
CLQUERYQ 00323    .                                                              .
CLQUERYQ 00325    .                                                              .
CLQUERYQ 00327    Only one file will be listed for a QUERY PUNCH specific command.

CLQUERYQ 00332    userid1   specifies the identification of the user who is
CLQUERYQ 00333              to receive the file.

CLQUERYQ 00336    spoolid   a unique, system assigned number which is used
CLQUERYQ 00337              by VM/370 to identify the file.

CLQUERYQ 00340    a         the output class of the file.

CLQUERYQ 00343    norec     gives the number of logical records contained in the file.

CLQUERYQ 00346    nn        is the number of copies of the output file to be produced.

CLQUERYQ 00349    hold      is the file hold status and is either:

CLQUERYQ 00351                   NONE - no hold
CLQUERYQ 00353                   USER - user hold
CLQUERYQ 00355                   SYS  - system hold
```

```
CLQUERYQ 00357                    USYS - system and user hold

CLQUERYQ 00360     mm/dd/yy  gives the data the file was created in month/day/year.

CLQUERYQ 00363     hh:mm:ss  gives the time of file creation in hours:minutes:seconds.

CLQUERYQ 00366     filename  gives the filename assigned to the file (if any).

CLQUERYQ 00369     filetype  gives the filetype assigned to the file (if any).

CLQUERYQ 00373     QUERY HOLD - Class D

CLQUERYQ 00375
CLQUERYQ 00377     HELD : |NO | RDR, |NO | PRT, |NO | PUN
CLQUERYQ 00378            |nnn|      |nnn|      |nnn|
CLQUERYQ 00379            |___|      |___|      |___|
CLQUERYQ 00380
CLQUERYQ 00381              |ALL|
CLQUERYQ 00383     userid - |RDR| , ...
CLQUERYQ 00384              |PRT|
CLQUERYQ 00385              |PUN|
CLQUERYQ 00386              |___|

CLQUERYQ 00388     The first response gives the total number of files within the
CLQUERYQ 00389     system which are held.  The second response gives the type
CLQUERYQ 00390     of hold (if any) for any user in the system for which a
CLQUERYQ 00391     hold is in effect.

CLQUERYQ 00394     QUERY PAGING - Class A

CLQUERYQ 00397     PAGING nn, SET mm, RATE nnn/SEC

CLQUERYQ 00400     nn        a percentage specifying the paging overhead threshold
CLQUERYQ 00401               at which the system is currently running.

CLQUERYQ 00404     mm        a percentage specifying the paging overhead threshold
CLQUERYQ 00405               which was established by default or by the SET PAGING command.

CLQUERYQ 00408     nnn/SEC   specifies the current paging rate in pages (nn) per second.

CLQUERYQ 00411     QUERY PRIORITY userid - Class A

CLQUERYQ 00414     userid PRIORITY = nn

CLQUERYQ 00416     nn  -  the assigned priority of the specified user.
```

```
CLQUERYQ 00419    QUERY LOGMSG - Class ALL

CLQUERYQ 00423    *  logmsg text line 1
CLQUERYQ 00424       .              .
CLQUERYQ 00425       .              .
CLQUERYQ 00426       .              .
CLQUERYQ 00427    *  logmsg text line n

CLQUERYQ 00429       logmsg additional text lines
CLQUERYQ 00430       .                          .
CLQUERYQ 00431       .                          .
CLQUERYQ 00432       .                          .

CLQUERYQ 00435    this gives all lines (both with  an asterisk and without) in
CLQUERYQ 00436    the log message file.


CLQUERYQ 00439    QUERY NAMES - Class ALL

CLQUERYQ 00443    userid - (DSC   ), ...
CLQUERYQ 00444    .        (raddr)     .
CLQUERYQ 00445    .                    .
CLQUERYQ 00446    .                    .
CLQUERYQ 00447    userid - (DSC   ), ...
CLQUERYQ 00448             (raddr)

CLQUERYQ 00454    this gives  a list of  all logged in  users; if the  user is
CLQUERYQ 00454    currently  connected,  the  real  address  tc  which  he  is
CLQUERYQ 00455    connected is  given (raddr); if he  is not connected  to the
CLQUERYQ 00456    system DSC is given.


CLQUERYQ 00459    QUERY USERS - Class ALL

CLQUERYQ 00463    nnn USERS mmm DIALED

CLQUERYQ 00466    nnn  specifies the total number of logged-in users.

CLQUERYQ 00468    mmm  specifies the total  number of users dialed  tc virtual
CLQUERYQ 00469         machines.


CLQUERYQ 00472    QUERY userid - Class ALL

CLQUERYQ 00474    userid - raddr

CLQUERYQ 00475    gives the real  address (raddr) to which  the specified user
CLQUERYQ 00476    is connected.


CLQUERYQ 00478    QUERY TIME - Class G

CLQUERYQ 00480    TIME IS NOW hh:mm:ss zone weekday mm/dd/yy
```

gives the current real clock time in hours:minutes:seconds, the time zone (e.g. EST), the day of the week, and the calendar date - month/day/year.

```
CONNECT hh:mm:ss VIRTCPU mmm:ss:hs TOTCPU mmm:ss:hs
```

gives the time spent in the current terminal session.

CONNECT	hh:mm:ss	gives the actual clock time spent in the current terminal session in hours:minutes:seconds.
VIRTCPU	mmm:ss:hs	gives the virtual CPU time used in the current terminal session in minutes:seconds:hundreths of seconds.
TOTCPU	mmm:ss:hs	gives the total CPU time (including virtual and overhead) used in the current terminal session in minutes:seconds:hundredths of seconds.

QUERY FILES - Class G

```
FILES: [nnn|NO] RDR, [nnn|NO] PRT, [nnn|NO] PUN
```

gives the total number of spool files in the system to be received by the user who issued the command.

QUERY READER - Class G

```
FILE   CLS RECDS ORIGIN   DATE     TIME     NAME      TYPE
spoolid a  norec userid mm/dd/yy hh:mm:ss filename  filetype
.                                                        .
.                                                        .
.                                                        .
```

Only one file will be listed for a QUERY READER spoolid command.

spoolid	a unique, system assigned number which is used by VM/370 to identify the file.
norec	gives the number of logical records contained in the file.
userid	specifies the identification of the user who originated the file.
mm/dd/yy	gives the date the file was created in month/day/year.

```
CLQUERYR 00047    hh:mm:ss  gives   the   time   of   file   creation   in
CLQUERYR 00048              hours:minutes:seconds.

CLQUERYR 00051    filename  gives the filename assigned to the file (if any).

CLQUERYR 00054    filetype  gives the filetype assigned to the file (if any).


CLQUERYR 00058    QUERY PRINTER - Class G

CLQUERYR 00060    FILE   CLS RECDS CPY HOLD DATE     TIME     NAME     TYPE
CLQUERYR 00062    spoolid a  norec  nn hold mm/dd/yy hh:mm:ss filename filetype
CLQUERYR 00066    .                                                     .
CLQUERYR 00068    .                                                     .
CLQUERYR 00070    .                                                     .
CLQUERYR 00071    Only one  file will  be listed for  a QUERY  PRINTER spoolid
CLQUERYR 00072    command.

CLQUERYR 00076    spoolid   a unique, system assigned number  which is used by
CLQUERYR 00077              VM/370 to identify the file.

CLQUERYR 00080    a         the output class of the file.

CLQUERYR 00082    norec     gives the  number of logical records  contained in
CLQUERYR 00083              the file.

CLQUERYR 00085    nn        is the number  of copies of the output  file to be
CLQUERYR 00086              produced.

CLQUERYR 00089    hold      is the file hold status and is either:

CLQUERYR 00091                   NONE - no hold
CLQUERYR 00093                   USER - user hold
CLQUERYR 00095                   SYS  - system hold
CLQUERYR 00097                   USYS - system and user hold

CLQUERYR 00099    mm/dd/yy  gives   the   date   the   file   was   created   in
CLQUERYR 00100              month/day/year.

CLQUERYR 00102    hh:mm:ss  gives   the   time   of   file   creation   in
CLQUERYR 00103              hours:minutes:seconds.

CLQUERYR 00106    filename  gives the filename assigned to the file (if any).

CLQUERYR 00109    filetype  gives the filetype assigned to the file (if any).
```

```
CLQUERYR 00113    QUERY PUNCH - Class G

CLQUERYR 00115    FILE   CLS RECDS CPY HOLD   DATE     TIME     NAME      TYPE
CLQUERYR 00117    spoolid a  norec  nn hold mm/dd/yy hh:mm:ss filename filetype
CLQUERYR 00121    .                                                      .
CLQUERYR 00123    .                                                      .
CLQUERYR 00125    .                                                      .
CLQUERYR 00127    Only  one file  will be  listed  for a  QUERY PUNCH  spoolid
CLQUERYR 00128    command.

CLQUERYR 00131    spoolid   a unique, system assigned number which is used by
CLQUERYR 00132              VM/370 to identify the file.

CLQUERYR 00135    a         the output class of the file.

CLQUERYR 00138    norec     gives the number of records in the file.

CLQUERYR 00140    nn        specifies the number of copies  of the output file
CLQUERYR 00141              to be produced.

CLQUERYR 00144    hold      is the file hold status and is either:

CLQUERYR 00146                   NONE - no hold
CLQUERYR 00148                   USER - user hold
CLQUERYR 00150                   SYS  - system hold
CLQUERYR 00152                   USYS - system and user hold

CLQUERYR 00154    mm/dd/yy  gives   the   date   the   file   was  created   in
CLQUERYR 00155              month/day/year.

CLQUERYR 00157    hh:mm:ss  gives    the    time    of    file    creation    in
CLQUERYR 00158              hours:minutes:seconds.

CLQUERYR 00161    filename  gives the filename assigned to the file (if any).

CLQUERYR 00164    filetype  gives the filetype assigned to the file (if any).


CLQUERYR 00170    QUERY SET - Class G

CLQUERYR 00172    gives the settings of all  functions which are controlled by
CLQUERYR 00173    the SET command.  Refer to the discussion of the SET command
CLQUERYR 00174    for explanations of the functions.

CLQUERYR 00176                                  {ON  }
CLQUERYR 00177                                  {OFF }
CLQUERYR 00178        {ON }       {ON}          {CODE}       {ON }
CLQUERYR 00179    MSG {OFF}, WNG {OFF}, EMSG {TEXT}, ACNT {OFF}

CLQUERYR 00181             {ON }       {ON }       {ON }
CLQUERYR 00182    LINEDIT {OFF}, TIMER {OFF}, RUN {OFF}
```

CLQUERYR 00185 **QUERY TERMINAL** - Class G

CLQUERYR 00189 gives the settings of all functions which are controlled by
CLQUERYR 00189 the TERMINAL command. Refer to the discussion of the
CLQUERYR 00191 TERMINAL command for explanations of the functions.

CLQUERYR 00193 LINEND n, LINEDEL n, CHARDEL n, ESCAPE n

CLQUERYR 00195 {ON } {ON } {ON }
CLQUERYR 00196 LINESIZE nnn, MASK {OFF}, APL {OFF}, ATTN {OFF}

CLQUERYR 00199 **QUERY VIRTUAL ALL** - Class G

CLQUERYR 00203 Has the same effect as if the following commands were
CLQUERYR 00204 issued:

CLQUERYR 00206 QUERY VIRTUAL STORAGE
CLQUERYR 00207 QUERY VIRTUAL LINES
CLQUERYR 00208 QUERY VIRTUAL TAPE
CLQUERYR 00209 QUERY VIRTUAL UR
CLQUERYR 00210 QUERY VIRTUAL DASD

CLQUERYR 00213 **QUERY VIRTUAL DASD** - Class G

CLQUERYR 00216 gives the status of each minidisk defined to the user's
CLQUERYR 00217 system.

CLQUERYR 00219 {R/W}
CLQUERYR 00220 DASD vaddr type volser {R/O} nnn CYL

CLQUERYR 00227 vaddr gives the virtual address to which the DASD device
CLQUERYR 00228 is attached.

CLQUERYR 00231 type = 2311
CLQUERYR 00232 2305
CLQUERYR 00233 2314
CLQUERYR 00234 3330
CLQUERYR 00235 231T (2311 at top of 2314)
CLQUERYR 00236 231B (2311 at bottom of 2314)

CLQUERYR 00241 volser gives the volume serial number of the system disk
CLQUERYR 00242 on which this minidisk resides.

CLQUERYR 00246 R/W defines the type of access the user has to
CLQUERYR 00248 R/O the disk.

CLQUERYR 00253 nnn gives the number of cylinders on the minidisk.

CLQUERYR 00257 **QUERY VIRTUAL LINES** – Class G

CLQUERYR 00258 gives status of all communication lines defined to the
CLQUERYR 00259 user's virtual machine.

CLQUERYR 00261 LINE vaddr ON DEV raddr

CLQUERYR 00264 vaddr the virtual address to which the lin is attached.

CLQUERYR 00267 raddr the real address of the line.

CLQUERYR 00272 {ENABLED }
CLQUERYR 00274 LINE vaddr {DISABLED}

CLQUERYR 00276 gives the status of virtual communication lines.

CLQUERYR 00281 **QUERY VIRTUAL UR** – Class G

CLQUERYR 00283 gives the status of all virtual unit record devices attached
CLQUERYR 00284 to the user's virtual machine.

CLQUERYR 00286 RDR vaddr CLS a
CLQUERYR 00288 {PUN} vaddr CLS a COPY nn
CLQUERYR 00290 {PRT}

CLQUERYR 00292 vaddr the virtual address at which the device is
CLQUERYR 00293 located.

CLQUERYR 00296 a the output class which the device services.

CLQUERYR 00298 nn the number of copies of each output file to be
CLQUERYR 00299 produced.

CLQUERYR 00303 **QUERY VIRTUAL STORAGE** – Class G

CLQUERYR 00305 STORAGE = nnnnnK

CLQUERYR 00306 gives the size of the virtual machine in multiples of 1024
CLQUERYR 00307 bytes.

CLQUERYR 00310 **QUERY VIRTUAL vaddr** – Class G

CLQUERYR 00312 The response will be in the same form as QUERY VIRTUAL DASD,
CLQUERYR 00314 TAPES, LINES, or UR, depending on virtual device type.

```
CLQUERYR 00317    QUERY LINKS vaddr

CLQUERYR 00319    gives a list of users who have linked to the device at virtual address vaddr.

CLQUERYR 00321    userid (R) vaddr, ...
CLQUERYR 00323    .      (W)         .
CLQUERYR 00325    .                  .
CLQUERYR 00327    .                  .
CLQUERYR 00329    userid (R) vaddr, ...
CLQUERYR 00330           (W)

CLQUERYR 00335    userid    identification of the user who originated the
CLQUERYR 00336              link.

CLQUERYR 00339    R         the type of access the user has to the device.
CLQUERYR 00342    W

CLQUERYR 00345    vaddr     the virtual address by which the user (userid)
CLQUERYR 00346              refers to the device.
```

CLREADY 00004 **READY**

CLREADY 00007 **Privilege Class:** G

CLREADY 00009 The READY command sets a device-end interrupt pending in the
CLREADY 00010 specified virtual device. The status of the virtual machine
CLREADY 00011 is unchanged. Other than having a device-end interrupt
CLREADY 00013 pending, the virtual device is unchanged.

CLREADY 00020
CLREADY 00021 | READY | vaddr |
CLREADY 00022
CLREADY 00023

CLREADY 00028 vaddr specifies a virtual device address (ccu).

CLREADY 00030 **Responses**

CLREADY 00032 None.

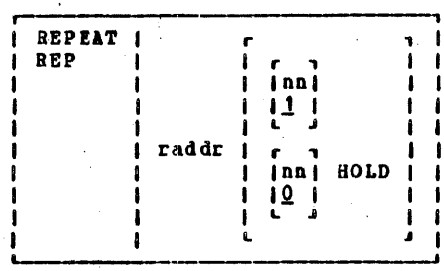
REPEAT

Privilege Class: D

The REPEAT command:

1. increases the number of copies of an output file originally requested by the file's creator.
2. places the current output file in a HOLD status, with or without increasing the number of copies to be created.

This command applies to files currently being written to a real punch or printer.



raddr specifies the address of the output device (printer or punch) whose current active spool file is to be repeated or held.

nn specifies the number of extra copies of the current file that are to be made. If nn is omitted and the HOLD option is not specified, the number of extra copies defaults to 1. The maximum value which may be specified for nn is 99.

HOLD
HO specifies that the file currently being printed or punched is not to be deleted when writing of the file is complete. If nn is not specified and this option is specified, when the last copy has completed, the file will be placed in system HOLD status until it is explicitly released.
Note: All copies indicated by a SPOOL command or by a previous REPEAT command will be printed before the HOLD option takes effect.

CLREPEAT 00073 **Responses**

CLREPEAT 00075 type raddr REPEATED userid FILE= spoolid COPY = nn

CLREPEAT 00077 type — either PRT or PUN.

CLREPEAT 00079 raddr — the real address of the device whose current
CLREPEAT 00081 output is being repeated.

CLREPEAT 00084 userid — the identification of the user who is to receive
CLREPEAT 00085 the output.

CLREPEAT 00088 spoolid — a unique, system assigned number by which VM/370
CLREPEAT 00089 refers to the file.

CLREPEAT 00091 nn — the remaining number of copies of the file to be
CLREPEAT 00092 created.

CLRESET 00003 **RESET**

CLRESET 00006 Privilege Class: G

CLRESET 00008 The RESET command clears all pending interrupts from the
CLRESET 00009 specified virtual device. In addition, all error conditions
CLRESET 00010 occurring as a result of unit checks and virtual sense bytes
CLRESET 00011 are reset.

CLRESET 00018 ┌──────────────────────────┐
CLRESET 00019 | RESET | vaddr |
CLRESET 00020 └──────────────────────────┘

CLRESET 00026 vaddr specifies a virtual device address (ccu).

CLRESET 00028 Responses

CLREWIND 00001 DEVICE RESET

CLREWIND 00004 **REWIND**

CLREWIND 00007 **Privilege Class:** G

CLREWIND 00008 The REWIND command allows the user to rewind a real tape
CLREWIND 00009 drive attached to his virtual machine at a specific virtual
CLREWIND 00010 device address. This allows the user to accomplish the
CLREWIND 00011 manual operation of rewinding and readying a tape at the
CLREWIND 00012 tape drive.

CLREWIND 00019
CLREWIND 00020 | REWIND | vaddr |
CLREWIND 00021 | REW | |
CLREWIND 00022

CLREWIND 00027 vaddr specifies the virtual device address (ccu) of the
CLREWIND 00028 tape to be rewound.

CLREWIND 00032 **Responses**

CLREWIND 00034 REWIND COMPLETE

CLREWIND 00036 REWIND NOT PERFORMED

CLREWIND 00037 is the response to the user if the real tape device is not
CLSAVSYS 00001 ready.

CLSAVSYS 00004 **SAVESYS**

CLSAVSYS 00007 **Privilege Class:** E

CLSAVSYS 00009 The SAVESYS command allows the system analyst to save a
CLSAVSYS 00010 virtual machine storage space with registers and PSW being
CLSAVSYS 00011 saved as they currently exist.

CLSAVSYS 00018 ┌──────────┬──────────────┐
CLSAVSYS 00019 | SAVESYS | system-name |
CLSAVSYS 00020 └──────────┴──────────────┘

CLSAVSYS 00027 **system-name** must be a pre-defined name representing a
CLSAVSYS 00027 definition of installation requirements of
CLSAVSYS 00028 the named system. The definition indicates
CLSAVSYS 00029 the number of pages to be saved, the DASD
CLSAVSYS 00029 volume on which the system is to be saved,
CLSAVSYS 00031 and the shared segments (if any). Refer to
CLSAVSYS 00032 the *IBM Virtual Machine Facility/370,*
CLSAVSYS 00033 *Planning and System Generation Guide* (Order
CLSAVSYS 00034 No. GC20-1801) for further information
CLSAVSYS 00035 concerning system definition.

CLSAVSYS 00038 **Response**

CLSETP 00001 SYSTEM SAVED

CLSETP 00005 **SET**

CLSETP 00008 Privilege Class: A, B, F, G

CLSETP 00010 The SET command allows the operator to establish system
CLSETP 00010 parameters, and the virtual user to control various
CLSETP 00012 functions within his virtual system. The SET commands are
CLSETP 00013 used as described to perform various functions available to
CLSETP 00014 users with specific privilege classes to control the CP
CLSETP 00015 system and virtual machine options.

CLSETP 00026 CLASS A:

CLSETP 00028 ┌──┐
CLSETP 00029 │ │ │
CLSETP 00030 │ SET │ { FAVORED userid {xx } } │
CLSETP 00031 │ S │ { {OFF} } │
CLSETP 00032 │ │ { └ ┘ } │
CLSETP 00033 │ │ { } │
CLSETP 00034 │ │ { RESERVE userid {xx } } │
CLSETP 00035 │ │ { {OFF} } │
CLSETP 00036 │ │ { } │
CLSETP 00037 │ │ { PAGING nn } │
CLSETP 00038 │ │ { } │
CLSETP 00039 │ │ { PRIORITY userid nn } │
CLSETP 00040 └──┘

CLSETP 00047 **FAVORED** The FAVORED function can be in effect for only
CLSETP 00048 **FAV** one virtual machine at any time. If xx (a
CLSETP 00049 percentage from 1-99 with 1 being the highest), is
CLSETP 00050 specified, VM/370 will attempt to provide up to
CLSETP 00051 the specified percentage of CPU time to that
CLSETP 00051 virtual machine, provided that virtual machine is
CLSETP 00052 functioning so that it can fully utilize the CPU
CLSETP 00053 time. At regular time intervals the VM/370
CLSETP 00053 dispatcher will check the CPU time used by the
CLSETP 00054 particular virtual machine; if the percentage has
CLSETP 00055 been exceeded, the machine is not run for the
CLSETP 00056 remainder of the interval. If the percentage is
CLSETP 00056 low, the virtual machine will have highest
CLSETP 00058 priority for the remainder of the interval.

CLSETP 00059 If OFF is specified, the specified virtual machine
CLSETP 00060 is removed from FAVORED status.

CLSETP 00061 If neither OFF nor xx is specified, the virtual
CLSETP 00062 machine designated will always be dispatchable;
CLSETP 00063 that is, any time he is ready to use CPU time, he
CLSETP 00064 will be given the highest priority.

CLSETP 00067 **RESERVE** allows the specified virtual machine to use the

```
CLSETP  00068    RES        specified number of page frames for its exclusive
CLSETP  00069               use. The page frames are not "locked" and the
CLSETP  00070               virtual machine can also use other available page
CLSETP  00071               frames; however, no other virtual machine can use
CLSETP  00071               a reserved page frame unless all page frames
CLSETP  00073               available have been exhausted.

CLSETP  00076    xx         specifies the number of page frames reserved.
CLSETP  00076               Maximum number is limited by the size of the
CLSETP  00077               virtual machine.

CLSETP  00080    OFF        removes the user from Reserve Page Status and
CLSETP  00081               returns the reserved pages to the system.

CLSETP  00084    PAGING     alters the system paging activity index to the
CLSETP  00086    PAG        specified value (nn). This value affects the
CLSETP  00087               paging rate and degree of multiprogramming used.
CLSETP  00088               The current value of this index may be displayed
CLSETP  00089               by the QUERY command.

CLSETP  00094    PRIORITY   sets the specified user's (userid) priority to the
CLSETP  00096    PRIOR      desired value (nn). This value affects the user's
CLSETP  00097               dispatching priority in relation to other users in
CLSETP  00098               the system.


CLSETP  00104    CLASS B:

CLSETP  00106    ┌──────────────────────────────────────┐
CLSETP  00107    │      │{                              } │
CLSETP  00108    │ SET  │{ LOGMSG    ┌nn  ┐             } │
CLSETP  00109    │      │{          │NULL│             } │
CLSETP  00110    │      │{          └    ┘             } │
CLSETP  00111    │      │{                    ┌   ┐     } │
CLSETP  00112    │      │{ DUMP     {AUTO }   │CP │     } │
CLSETP  00113    │      │{         {raddr}    │ALL│     } │
CLSETP  00114    │      │{                    └   ┘     } │
CLSETP  00115    └──────────────────────────────────────┘

CLSETP  00119    LOGMSG     LOGMSG or LOGMSG nn causes CP to respond with the
CLSETP  00121    LOG        message LOGMSG:, followed by a read to enable the
CLSETP  00122               operator to enter the line. LOGMSG with no
CLSETP  00123               parameter is used to initialize or add to a log
CLSETP  00124               message. LOGMSG nn specifies that a change is to
CLSETP  00124               be made to existing line number nn in the log
CLSETP  00127               message. A null line entered terminates LOGMSG
CLSETP  00128               entry (for adding or initializing) or deletes line
CLSETP  00129               nn if that option is used. LOGMSG NULL deletes
CLSETP  00131               the entire existing LOGMSG, allowing a new LOGMSG
CLSETP  00132               to be created.

CLSETP  00135    DUMP       DUMP sets the VM/370 system ABEND dump unit, where
CLSETP  00137    DU         raddr is a real printer or tape address. If AUTO
CLSETP  00138               is specified the system dump unit will be disk.
```

```
CLSETP  00139              The  CP  option  specifies to  dump  only  storage
CLSETP  00140              occupied by  the control program, whereas  the ALL
CLSETP  00141              option dumps all storage.

CLSETP  00147    CLASS F:

CLSETP  00150    ┌──────────────────────────────────────────────────────────────┐
CLSETP  00151    | SET |         {OFF                                          }|
CLSETP  00152    |     |         {CPU                                          }|
CLSETP  00153    |     | RECORD  {                                  ┌         ┐ }|
CLSETP  00154    |     |         {ON raddr LIMIT nn BYTE nn BIT n   |{AND} BYTE nn BIT n| }|
CLSETP  00155    |     |         {                                  |{OR }      | }|
CLSETP  00156    |     |         {                                  └         ┘ }|
CLSETP  00157    └──────────────────────────────────────────────────────────────┘

CLSETP  00161    RECORD    ON allows the Class F user to specify intensive
CLSETP  00163    REC       recording mode for the device 'raddr'.  LIMIT
CLSETP  00164              identifies the value at which a recording will be
CLSETP  00165              made, and the BYTE and BIT operands define the
CLSETP  00166              test condition in the sense bytes.  Two bits may
CLSETP  00167              be tested by an AND or OR relationship.  The OFF
CLSETP  00168              function turns off intensive recording mode.  Only
CLSETP  00169              one device at a time may have intensive recording
CLSETP  00169              specified.  The CPU operand specifies that CP is
CLSETP  00170              to resume the recording of "soft" machine errors
CLSETP  00171              previously turned off by VM/370 recovery
CLSETP  00172              management system procedures due to excessive
CLSETP  00173              occurrences.

CLSETP  00177    CLASS G:

CLSETP  00179    ┌──────────────────────────┐
CLSETP  00180    | SET |{{ACNT    }    {ON }  }|
CLSETP  00181    |     |{{MSG     }    {OFF}  }|
CLSETP  00182    |     |{{WNG     }           }|
CLSETP  00183    |     |{{RUN     }           }|
CLSETP  00184    |     |{{LINEDIT}           }|
CLSETP  00185    |     |{{NOTRANS}           }|
CLSETP  00186    |     |{                    }|
CLSETP  00187    |     |{ EMSG          {ON  }}|
CLSETP  00188    |     |{               {OFF }}|
CLSETP  00189    |     |{               {CODE}}|
CLSETP  00190    |     |{               {TEXT}}|
CLSETP  00191    |     |{                    }|
CLSETP  00192    |     |{ TIMER         {ON  }}|
CLSETP  00193    |     |{               {OFF }}|
CLSETP  00194    |     |{               {REAL}}|
CLSETP  00195    └──────────────────────────┘

CLSETP  00201    ACNT      controls whether accounting infcrmation is typed
CLSETP  00203    AC        at the user's terminal or not (CN and CFF
```

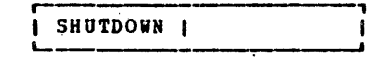
```
CLSETP  00204           respectively) when the operator issues the CP ACNT
CLSETP  00205           command.

CLSETP  00208   MSG     controls whether messages sent by the MSG command
CLSETP  00209   M       from other users are to be received at the
CLSETP  00211           terminal. If ON is specified, the messages are
CLSETP  00211           typed. OFF specifies that no messages are
CLSETP  00212           received.

CLSETP  00215   WNG     If ON is specified, all warning messages sent via
CLSETP  00217   W       the CP WNG command from other users are received
CLSETP  00218           at the terminal. If OFF is specified, no warning
CLSETP  00219           messages are received.

CLSETP  00222   RUN     ON allows the user to activate the ATTN key,
CLSETP  00224   R       (causing a read of a CP command) without stopping
CLSETP  00225           his virtual machine. When the CP function is
CLSETP  00226           typed in, it is immediately executed and the
CLSETP  00227           virtual machine resumes execution. OFF places the
CLSETP  00228           user in the normal CP environment, so that when
CLSETP  00229           ATTN is activated, the virtual machine stops.

CLSETP  00234   LINEDIT ON specifies that the line editing functions and
CLSETP  00235   L       the symbols of the VM/370 system are to be used in
CLSETP  00236           the virtual machine operating system to edit
CLSETP  00237           virtual CPU console input requests. This
CLSETP  00238           establishes line editing features in systems that
CLSETP  00239           do not normally provide them. OFF specifies that
CLSETP  00240           no character or line editing is to be used for the
CLSETP  00241           virtual machine operating system. See "Appendix
CLSETP  00241           B: Terminal Considerations" for a description of
CLSETP  00242           VM/370 Line Editing Capabilities.

CLSETP  00245   NOTRANS can only be specified by a virtual machine that
CLSETP  00247   NOTR    occupies the virt=real space. It causes all
CLSETP  00247           virtual I/O from the issuing virtual machine to
CLSETP  00249           bypass the CP CCW translation.

CLSETP  00252   EMSG    ON specifies that, when the user receives an error
CLSETP  00253   E       message, he will get both error code and text.
CLSETP  00254           TEXT specifies that only text be printed. CODE
CLSETP  00255           specifies that only the error code be printed.
CLSETP  00256           OFF specifies that no error message is to be
CLSETP  00257           printed.

CLSETP  00260   TIMER   OFF specifies that the virtual timer is not to be
CLSETP  00261   T       updated. ON specifies that the virtual timer is
CLSETP  00262           to be be updated only when the virtual CPU is
CLSETP  00263           running. REAL specifies that the virtual timer is
CLSETP  00264           to be updated during virtual CPU run time and also
CLSETP  00265           virtual wait time.
```

CLSETP	00269	Response
CLSHUTDN	00001	None.

CLSHUTDN 00005 **SHUTDOWN**

CLSHUTDN 00008 **Privilege Class:** A

CLSHUTDN 00009 The SHUTDOWN command systematically terminates all VM/370
CLSHUTDN 00011 functions and checkpoints the system for an eventual warm
CLSHUTDN 00012 start. This is the normal method used for shutting the
CLSHUTDN 00013 system down. All VM/370 functions are suspended and
CLSHUTDN 00015 communications lines are disabled.

CLSHUTDN 00022
CLSHUTDN 00023 
CLSHUTDN 00024

CLSHUTDN 00029 **Response**

CLSHUTDN 00031 The operator is notified of command completion by:

CLSLEEP 00001 SYSTEM SHUTDOWN COMPLETE

CLSLEEP 00005 **SLEEP**

CLSLEEP 00008 **Privilege Class:** ALL

CLSLEEP 00009 The SLEEP command places the virtual machine in a dormant
CLSLEEP 00010 state with the keyboard locked, but allows messages to be
CLSLEEP 00011 printed out without the user signaling a carriage return.
CLSLEEP 00011 The user's virtual machine is not run during this time, but
CLSLEEP 00012 his connect time is still being counted. The terminal can
CLSLEEP 00013 be "awakened" by pressing the ATTN key (or its equivalent).
CLSLEEP 00015 This returns the user to CP console function mode.

CLSLEEP 00022 ┌──────────────────────┐
CLSLEEP 00023 | SLEEP | |
CLSLEEP 00024 | SL | |
CLSLEEP 00025 └──────────────────────┘

CLSLEEP 00029 **Responses**

CLSPACE 00001 None.

CLSPACE 00005 **SPACE**

CLSPACE 00008 **Privilege Class:** D

CLSPACE 00009 The SPACE command forces the output on the specified printer
CLSPACE 00010 to be single-spaced for the current active spool file,
CLSPACE 00011 regardless of the carriage control commands contained in the
CLSPACE 00012 actual file.

CLSPACE 00019
CLSPACE 00020 | SPACE | raddr |
CLSPACE 00021 | SPA | |
CLSPACE 00022

CLSPACE 00028 raddr is the real address (ccu) of the printer whose
CLSPACE 00030 output is to be single spaced.

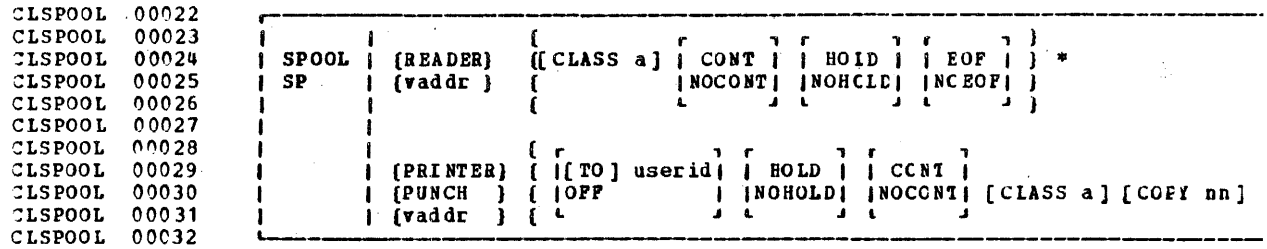
CLSPACE 00033 **Responses**

CLSPOOL 00001 None.

SPOOL

Privilege Class: G

The SPOOL command modifies the spooling control options in effect for a given virtual spooling device or for a group of devices.



* at least one of the options within braces must be selected; however, more than one may be specified and they may be entered in any order. This is contrary to the notation normally used in this publication.

vaddr — is the device address (cuu) of the virtual spooling unit record device whose options are to be modified.

READER, R, RDR — indicates that the function is to be applied to all virtual devices of the specified type.

PRINTER, P, PRT

PUNCH, PU, PCH

CONT, CON — for virtual readers, this specifies that reading be continuous with all end-of-file indicators being ignored until all files spooled to the virtual machine are read in. If this option is not in effect, a unit exception is reflected to the user's virtual machine at the end of each spooled file. CONT specified for the punch or printer causes all CLOSE requests to be ignored, until reset by NOCONT. If CONT is specified, NOCONT may not be specified

```
CLSPOOL  00082    NOCONT     resets the continuous spooling option.  If NOCCNT
CLSPOOL  00084    NOC        is specified, CONT may not be specified.

CLSPOOL  00087    COPY nn    specifies the number of copies that are tc be made
CLSPOOL  00089    CO         of the file when it is spooled tc the real unit
CLSPOOL  00090               record equipment.  This parameter is valid cnly
CLSPOOL  00090               for output files; nn must be between 1 and 99;
CLSPOOL  00091               leading zeroes need not be specified.

CLSPOOL  00094    CLASS a    specifies the class that is to be assigned tc the
CLSPOOL  00096    CL         given device.  For input files, only devices with
CLSPOOL  00096               classes matching those of the device will be read.
CLSPOOL  00097               A is a one-character alphameric field with values
CLSPOOL  00098               from A to Z or from 0-9.

CLSPOOL  00101    HOLD       specifies that all output files created by the
CLSPOOL  00102    HO         specified device are to be placed in a user HCLD
CLSPOOL  00103               status. For READER files, this optic specifies
CLSPOOL  00104               that input files for the specified reader will not
CLSPOOL  00104               be deleted from the system after they have been
CLSPOOL  00105               read.  If HOLD is specified, NOHCID may not be
CLSPOOL  00106               specified.

CLSPOOL  00109    NOHOLD     resets the HOLD option.  Future files will not be
CLSPOOL  00110    NOH        held.  NOHOLD resets the HOLD optic in effect fcr
CLSPOOL  00112               the specified reader.  This cptic can be
CLSPOOL  00113               overridden for the active file beirg clcsed by the
CLSPOOL  00115               CLOSE command using the HCLD or NCHOLD cption.  If
CLSPOOL  00116               NOHOLD is specified, HOLD may not be specified.

CLSPOOL  00119    EOF        indicates that the virtual end-of-file button cn
CLSPOOL  00120               the specified reader has been pressed, thereby
CLSPOOL  00121               ensuring that a unit exception ccrditicn will be
CLSPOOL  00122               reflected on the read that follows the reading of
CLSPOOL  00123               the last card in a file.  If EOF is specified,
CLSPOOL  00124               NOEOF may not be specified.

CLSPOOL  00127    NOEOF      specifies that the reading continues to physical
CLSPOOL  00129    NOE        end-of-file.  The virtual reader will stcp with nc
CLSPOOL  00130               cards left in the reader and with a unit
CLSPOOL  00131               check/intervention required status pending.  If
CLSPOOL  00132               NOEOF is specified EOF may not be specified.

CLSPOOL  00135    TO userid
CLSPOOL  00136    T          specifies that the output of the virtual spooling
CLSPOOL  00137               device is to be transferred to the virtual card
CLSPOOL  00138               reader of the specified userid.  If TO is cmitted,
CLSPOOL  00139               userid may not be TO or T.  TO * may be coded if
CLSPOOL  00139               the output is to be transferrad tc the virtual
CLSPOOL  00141               card reader of the user issuing the command.  If TO
CLSPOOL  00141               userid is specified, OFF may not be specified.
CLSPOOL  00142               HOLD and COPY have no effect in a transfer of
CLSPOOL  00143               files if TO userid is specified.  Hcwever, if they
CLSPOOL  00144               are specified with TO userid, and the TO userid
```

```
CLSPOOL  00145              specification is subsequently cancelled by issuing
CLSPOOL  00146              a SPOOL  command with the OFF  specification, they
CLSPOOL  00147              will be in effect.  For  example, if the following
CLSPOOL  00148              command is entered:

CLSPOOL  00150              SPOOL PUN TO usera COPY 3 CLASS B HOLD

CLSPOOL  00151              the COPY and HOLD options will have no effect.  If
CLSPOOL  00153              the command:

CLSPOOL  00155              SPOOL PUN OFF

CLSPOOL  00156              is  then entered,  the  options  specified in  the
CLSPOOL  00157              previous SPOOL PUN command will be in effect; that
CLSPOOL  00159              is, COPY 3 and HOLD will now be in effect.

CLSPOOL  00162    OFF       resets the transferred spool option.

CLSPOOL  00164    Note: CONT specifies that  reading is  to continue  without
CLSPOOL  00165    intervening end-of-file  indications until all files  in the
CLSPOOL  00166    system that belong  to the user have been read.  If CONT is
CLSPOOL  00167    not in  effect or  is reset by  specification of  NOCONT, an
CLSPOOL  00167    end-of-file indication  is reflected to the  virtual machine
CLSPOOL  00168    at the end of each SPOOL file  in the system.  The nature of
CLSPOOL  00169    the end-of-file  indication to be  reflected is set  by the
CLSPOOL  00170    EOF|NOEOF option.  If  the  EOF  option  is  in  effect,
CLSPOOL  00171    end-of-file  is  signaled  by  a  unit  exception.  This
CLSPOOL  00171    corresponds to  pressing the  end-of-file button  on a  real
CLSPOOL  00172    card reader. If the virtual EOF  button is not pressed (that
CLSPOOL  00173    is,  NOEOF is  in effect),  end-of-file is  signaled by  the
CLSPOOL  00175    reflection of a unit check/intervention required status.


CLSPOOL  00177    Responses

CLSTARTP 00001    None.
```

START

Privilege Class: D

The START command restarts a spooling device after it has been drained, or changes the output class that it may service. If the device is busy when the command is given, the changed classes take effect after the current file is complete.

```
┌───────────────────────────────────────────────────────┐
│ START │ ┌                                          ┐ │
│ STA   │ │   READER                                 │ │
│       │ │   PRINTER                                │ │
│       │ │   PUNCH                                  │ │
│       │ │   ALL                                    │ │
│       │ │                                          │ │
│       │ │ [raddr [CLASS c ] [NOSEP]]...            │ │
│       │ │                                          │ │
│       │ └                                          ┘ │
└───────────────────────────────────────────────────────┘
```

READER R RDR	specifies the type of spooling devices(s). If ALL is specified, all spooling devices will be started.
PRINTER P PRT	
PUNCH PU PCH	
ALL	
raddr	specifies the address (ccu) of a particular spooling device which is to be started. More than one device may be specified.
CLASS CL	specifies the output classes the specified device is allowed to service. From one to four classes (single characters with no intervening blanks) may be specified. If multiple classes are specified for a device, output files receive service by the device in the order that the classes were specified. For example, if the option: CLASS AD1 is specified for a printer, all Class A files

```
CLSTARTP 00077                will be spooled before Class D files, and all
CLSTARTP 00078                Class D before Class 1.

CLSTARTP 00081    NOSEP       specifies that no file separator is desired for
CLSTARTP 00083    NOS         the output files. If not specified, a separator
CLSTARTP 00084                is produced.

CLSTARTP 00086                If NOSEP is specified for the punch, separator
CLSTARTP 00087                cards are not punched; instead, the operator
CLSTARTP 00088                receives the message:

CLSTARTP 00090                PUN raddr START FOR OUTPUT

CLSTARTP 00091                The punch can then be started for the next file
CLSTARTP 00092                by a CP START command or physical activation of
CLSTARTP 00093                the START/STOP key on the device.

CLSTARTP 00095    Note: Printers and punches that are in a ready status with
CLSTARTP 00096    files available for processing will immediately start
CLSTARTP 00097    producing output when the START command is given. Readers,
CLSTARTP 00098    however, will not start until a card deck is supplied and
CLSTARTP 00099    the device is made ready.


CLSTARTP 00103    Responses


CLSTARTP 00107    type raddr SPOOL CLS [a...] STARTED

CLSTARTP 00109    type one of the following:  RDR
CLSTARTP 00110                               PRT
CLSTARTP 00111                               PUN

CLSTARTP 00115    a     (for PRT and PUN only) indicates the classes to be
CLSTARTP 00116          serviced by the output spooling device being started.
```

CLSTCP 00005 **STCP**

CLSTCP 00008 **Privilege Class:** C

CLSTCP 00009 The STCP command allows alteration of the contents of real
CLSTCP 00011 storage by system programmers. The real PSW or real
CLSTCP 00012 registers may not be altered with this command.

CLSTCP 00023 ┌──┐
CLSTCP 00024 | STCP | {{ hexloc} hexword1 [hexword2...]} |
CLSTCP 00025 | | {{Lhexloc} } |
CLSTCP 00026 | | { } |
CLSTCP 00027 | | {Shexloc hexdata } |
CLSTCP 00028 └──┘

CLSTCP 00034 **hexloc** specifies that the data given in hexword1
CLSTCP 00036 **Lhexloc** hexword2... is to be stored in successive fullword
CLSTCP 00037 locations starting at the address specified by
CLSTCP 00038 hexloc. The smallest group of hexadecimal values
CLSTCP 00038 that can be stored using this specification is one
CLSTCP 00039 fullword. Alignment is made to the nearest
CLSTCP 00040 fullword boundary. If the value being stored is
CLSTCP 00041 less than a fullword (eight hexadecimal digits),
CLSTCP 00041 it is right-adjusted in the word and the high
CLSTCP 00043 order bytes of the word are filled with zeros.
CLSTCP 00043 Either specification (hexloc or Lhexloc) may be
CLSTCP 00044 used.

CLSTCP 00046 **Shexloc** specifies that the data given in hexdata is to be
CLSTCP 00048 stored in the address specified by hexloc without
CLSTCP 00049 word alignment. The shortest string that can be
CLSTCP 00050 stored is one byte (two hexadecimal digits). If
CLSTCP 00050 the string contains an odd number of characters,
CLSTCP 00051 the last character is not stored. An error
CLSTCP 00053 message is sent and the function is terminated.

CLSTCP 00056 **hexword** specifies up to eight hexadecimal digits. If less
CLSTCP 00056 than eight digits are specified, the string is
CLSTCP 00057 right justified in a fullword and left-filled with
CLSTCP 00057 zeroes. If two or more hexwords are given, they
CLSTCP 00059 must be separated by one or more blanks.

CLSTCP 00061 **hexdata** specifies a string of two or more hexadecimal
CLSTCP 00063 digits with no embedded blanks.

CLSTCP 00066 **Response**

CLSTORE 00001 STORE COMPLETE

CLSTORE 00005 **STORE**

CLSTORE 00008 **Privilege Class:** G

CLSTORE 00010 The STORE command alters the contents of specified
CLSTORE 00011 components of the virtual machine. The contents of the
CLSTORE 00012 following may be altered:

CLSTORE 00016 • virtual storage locations
CLSTORE 00018 • general registers
CLSTORE 00020 • floating-point registers
CLSTORE 00022 • extended control registers (if available)
CLSTORE 00024 • program status word

CLSTORE 00026 The various options may be combined in any order desired,
CLSTORE 00027 separated by one or more blanks, for up to one full line of
CLSTORE 00028 input. If an invalid argument is encountered, an error
CLSTORE 00029 message is issued and the store function is terminated;
CLSTORE 00029 however, if there have been valid arguments before the
CLSTORE 00040 invalid one, they would have been processed properly.

CLSTORE 00042
CLSTORE 00043 | STORE |{ {hexloc } } |
CLSTORE 00044 | ST |{ {Lhexloc} hexword1 [hexword2...] } |
CLSTORE 00045 | |{ } |
CLSTORE 00046 | |{ Shexloc hexdata } |
CLSTORE 00047 | |{ } |
CLSTORE 00048 | |{ {Greg} } |
CLSTORE 00049 | |{ {Yreg} hexword1 [hexword2...] } |
CLSTORE 00050 | |{ {Xreg} } |
CLSTORE 00051 | |{ } |
CLSTORE 00052 | |{ PSW [hexword1] hexword2 } |
CLSTORE 00053

CLSTORE 00059 **hexloc** specifies that the data given in hexword1
CLSTORE 00061 **Lhexloc** hexword2... is to be stored in successive fullword
CLSTORE 00062 locations starting at the address specified by
CLSTORE 00063 hexloc. The smallest group of hexadecimal values
CLSTORE 00063 that can be stored using this specification is one
CLSTORE 00064 fullword. Alignment is made to the nearest
CLSTORE 00065 fullword boundary. If the value being stored is
CLSTORE 00066 less than a fullword (eight hexadecimal digits),
CLSTORE 00066 it is right-adjusted in the word and the high
CLSTORE 00068 order bytes of the word are filled with zeros.
CLSTORE 00068 Either specification (hexloc or Lhexlcc) may be
CLSTORE 00069 used.

CLSTORE 00071 **Shexloc** specifies that the data given in hexdata is to be
CLSTORE 00073 stored in the address specified by hexlcc without
CLSTORE 00074 word alignment. The shortest string that can be
CLSTORE 00075 stored is one byte (two hexadecimal digits). If
CLSTORE 00075 the string contains an odd number of characters,

```
CLSTORE  00076              the last character is not stored.  An error
CLSTORE  00078              message is sent and the function is terminated.

CLSTORE  00080   Greg       specifies that the hexadecimal data given in
CLSTORE  00082              hexword1 hexword2... is to be placed in successive
CLSTORE  00083              general registers starting at the register
CLSTORE  00083              specified by reg.  The reg parameter must be
CLSTORE  00084              either a decimal number ranging from 0-15 or a
CLSTORE  00085              hexadecimal digit ranging from 0-F.

CLSTORE  00088   Freg       specifies that the hexadecimal data given in
CLSTORE  00089              hexword1 hexword2... is to be placed in successive
CLSTORE  00089              floating-point registers starting at the register
CLSTORE  00090              specified by reg.  The reg parameter must be a
CLSTORE  00091              digit ranging from 0-6.  If reg is an odd number,
CLSTORE  00092              it is adjusted to the preceding even number.

CLSTORE  00095   Xreg       specifies that the hexadecimal data given in
CLSTORE  00096              hexword1 hexword2... is to be placed in successive
CLSTORE  00096              extended control registers starting at the
CLSTORE  00097              register specified by reg.  The reg parameter must
CLSTORE  00098              either be a decimal number from 0-15 or a
CLSTORE  00099              hexadecimal digit from 0-F.  If the user is in
CLSTORE  00100              basic control mode, he can store data in register
CLSTORE  00101              0 only.

CLSTORE  00104   PSW        specifies that the hexadecimal data given in
CLSTORE  00107   P          hexword1 hexword2... is to be placed in the first
CLSTORE  00109              and second words of the user's virtual program
CLSTORE  00110              status word (PSW).  If only hexword2 is specified,
CLSTORE  00111              it is stored into the second word of the PSW.
CLSTORE  00112              Hexword1 and hexword2 must be separated by one or
CLSTORE  00113              more blanks.

CLSTORE  00116   hexword    specifies up to eight hexadecimal digits.  If less
CLSTORE  00116              than eight digits are specified, the string is
CLSTORE  00117              right justified in a fullword and left-filled with
CLSTORE  00117              zeroes.  If two or more hexwords are given, they
CLSTORE  00119              must be separated by one or more blanks.

CLSTORE  00121   hexdata    specifies a string of two or more hexadecimal
CLSTORE  00123              digits with no embedded blanks.


CLSTORE  00126   Response

CLSYSTEM 00001   STORE COMPLETE
```

CLSYSTEM 00003 **SYSTEM**

CLSYSTEM 00006 **Privilege Class:** G

CLSYSTEM 00007 The SYSTEM command simulates the action of the RESET and
CLSYSTEM 00008 RESTART buttons on the real computer console, and also
CLSYSTEM 00009 allows storage to be cleared. The RESET function and the
CLSYSTEM 00010 CLEAR function leave the virtual machine in a stopped state.
CLSYSTEM 00011 A BEGIN command must be issued to resume operation after a
CLSYSTEM 00012 SYSTEM RESET. An IPL command must be issued after a SYSTEM
CLSYSTEM 00013 CLEAR command. The virtual machine is automatically
CLSYSTEM 00014 restarted on a RESTART function at the location loaded into
CLSYSTEM 00015 the PSW from the doubleword at virtual location zero.

CLSYSTEM 00024
CLSYSTEM 00025 | SYSTEM | {CLEAR } |
CLSYSTEM 00026 | SYS | {RESET } |
CLSYSTEM 00027 | | {RESTART} |
CLSYSTEM 00028

CLSYSTEM 00035 CLEAR specifies that virtual storage and virtual storage
CLSYSTEM 00036 keys are to be cleared to binary zeros.

CLSYSTEM 00038 RESET specifies that all pending interrupts and
CLSYSTEM 00039 conditions in the virtual machine are to be
CLSYSTEM 00040 cleared.

CLSYSTEM 00043 RESTART simulates the hardware system RESTART function by
CLSYSTEM 00044 storing the current PSW at virtual location eight
CLSYSTEM 00045 and loading, as the new PSW, the doubleword from
CLSYSTEM 00046 virtual location zero. Interrupt conditions and
CLSYSTEM 00047 storage remains unaffected.

CLSYSTEM 00050 **Responses**

CLSYSTEM 00052 STORAGE CLEARED - SYSTEM RESET

CLSYSTEM 00053 This response is given if the command SYSTEM CLEAR is
CLSYSTEM 00054 entered.

CLSYSTEM 00056 SYSTEM RESET

CLSYSTEM 00057 This response is given if the command SYSTEM RESET is
CLSYSTEM 00058 entered.

CLSYSTEM 00060 If the command SYSTEM RESTART is entered, no response is
CLSYSTEM 00061 given; the virtual machine resumes execution at the address
CLSYSTEM 00061 in the virtual PSW loaded from virtual storage location

CLTERM 00001 zero.

CLTERM 00003 **TERMINAL**

CLTERM 00006 **Privilege Class:** G

CLTERM 00007 The TERMINAL command specifies various options used to
CLTERM 00009 control terminal input and output processing.

```
CLTERM 00019 ┌──────────────────────────────────────────┐
CLTERM 00020 | TERMINAL |{ {CHARDEL}  {ON  }  }*|
CLTERM 00021 | TERM     |{ {LINEDEL}  {OFF }  }  |
CLTERM 00022 |          |{ {LINEND }  {char}  }  |
CLTERM 00023 |          |{ {ESCAPE }          }  |
CLTERM 00024 |          |{                    }  |
CLTERM 00025 |          |{ {MASK}     {ON  }  }  |
CLTERM 00026 |          |{ {APL }     {OFF}   }  |
CLTERM 00027 |          |{ {ATTN}             }  |
CLTERM 00028 |          |{                    }  |
CLTERM 00029 |          |{  LINESIZE  nnn     }  |
CLTERM 00030 └──────────────────────────────────────────┘
```

CLTERM 00035 * More than one function may be specified in a single entry
CLTERM 00036 of the TERMINAL command. For example:

```
CLTERM 00038 ┌──────────────────────────────────────┐
CLTERM 00039 | TERMINAL CHARDEL OFF MASK ON LINESIZE 90 |
CLTERM 00040 └──────────────────────────────────────┘
```

CLTERM 00048 **CHARDEL** specifies that the logical character delete
CLTERM 00050 **CH** function (normally the @ character) is to be
CLTERM 00050 invoked according to specifications in the next
CLTERM 00052 operand (ON, OFF, or char). Refer to "Appendix E:
CLTERM 00053 Terminal Considerations" for an explanation of the
CLTERM 00054 logical character delete function.

CLTERM 00057 **LINEDEL** specifies that the logical line delete function
CLTERM 00059 **LINED** (normally the ¢ character) is to be invoked
CLTERM 00060 according to sepcification in the next operand
CLTERM 00061 (ON, OFF, or char). Refer to "Appendix E: Terminal
CLTERM 00062 Considerations" for an explanation of the logical
CLTERM 00063 line delete function.

CLTERM 00066 **LINEND** specifies that the logical line end function
CLTERM 00067 **LINEN** (normally the # character) is to be invoked
CLTERM 00068 according to specification in the next operand
CLTERM 00070 (ON, OFF, or char). Refer to "Appendix E: Terminal
CLTERM 00071 Considerations" for an explanation of the logical
CLTERM 00072 line end function.

CLTERM 00075 **ESCAPE** specifies that the logical escape character
CLTERM 00076 **ES** function (normally the " character) is to be
CLTERM 00077 invoked according to the next operand (ON, OFF, or
CLTERM 00078 char). Refer to "Appendix E: Terminal

```
CLTERM   00079                Considerations" for an explanation of the logical
CLTERM   00080                escape character function.

CLTERM   00083   ON           specifies that the selected function is to be set
CLTERM   00084                to the system default.

CLTERM   00087   OFF          specifies that the selected function is to be
CLTERM   00089   OF           eliminated.

CLTERM   00092   char         specifies that this character is to be used for
CLTERM   00092                the specified function instead of the standard
CLTERM   00093                setting.

CLTERM   00096   MASK         is used to control the typing of a mask line at
CLTERM   00097   M            the terminal when a password is to be entered. If
CLTERM   00098                ON is specified in the next operand, the VM/370
CLTERM   00099                system will type the mask line. If OFF is
CLTERM   00100                specified, the mask line is not typed and it up to
CLTERM   00102                the user to preserve the security of his password.

CLTERM   00104   APL          controls the use of APL character translation
CLTERM   00106                tables. If ON is specified, CP uses the
CLTERM   00107                translation tables for terminals equipped with the
CLTERM   00108                standard APL typing element. If OFF is specified,
CLTERM   00108                CP will use the normal translation tables (that
CLTERM   00110                is, BCD or correspondence code).

CLTERM   00113   ATTN         with the ON option, normal attention handling is
CLTERM   00115   AT           in effect. The OFF option suppresses the typing
CLTERM   00115                of the (!) and carriage return for those systems
CLTERM   00116                that perform special line editing using the ATTN
CLTERM   00117                key.

CLTERM   00120   LINESIZE nnn
CLTERM   00122   LINES        allows the user to control the length of a typed
CLTERM   00123                line for his terminal output. nnn is a count of
CLTERM   00124                the number of characters per line.


CLTERM   00127   Response

CLTRACE  00001   None.
```

CLTRACE 00005 **TRACE**

CLTRACE 00008 **Privilege Class:** G

CLTRACE 00010 The TRACE command causes the CP system to trace specified
CLTRACE 00011 virtual machine activity and to record the results at the
CLTRACE 00012 user's terminal, on a virtual spooled printer, or on both
CLTRACE 00013 terminal and printer. If trace output is being recorded at
CLTRACE 00014 the terminal, the virtual machine stops execution and CP
CLTRACE 00015 console function mode is entered after each output message.
CLTRACE 00017 This affects the simulation of the single cycle function. To
CLTRACE 00017 resume operation at the virtual machine, the BEGIN command
CLTRACE 00018 must be entered. If the RUN option is specified, the virtual
CLTRACE 00020 machine is not stopped after each output message. If trace
CLTRACE 00021 output is being recorded on a virtual spooled printer, a
CLTRACE 00022 CLOSE command must be issued to that printer in order for
CLTRACE 00023 the trace output to be printed. Successful branches to the
CLTRACE 00024 next sequential instruction and branch-to-self instructions
CLTRACE 00025 are not detected by TRACE. Instructions that modify or
CLTRACE 00026 examine the first two bytes of the next sequential
CLTRACE 00027 instruction will cause erroneous processing for BRANCH and
CLTRACE 00028 INSTRUCT tracing.

CLTRACE 00040
CLTRACE 00041 | TRACE |{{SVC }*
CLTRACE 00042 | TR |{{I/O }
CLTRACE 00043 |{{PROGRAM }
CLTRACE 00044 |{{EXTERNAL}
CLTRACE 00045 |{{PRIV } | PRINTER
CLTRACE 00046 |{{SIO } |{ TERMINAL | NOBUN |
CLTRACE 00047 |{{CCW } |{ BOTH | RUN |
CLTRACE 00048 |{{BRANCH }
CLTRACE 00049 |{{INSTRUCT}
CLTRACE 00050 |{{ALL } |
CLTRACE 00051 |{{CSW } | OFF
CLTRACE 00052 |{ END
CLTRACE 00053

CLTRACE 00058 * More than one of these activities may be traced by using
CLTRACE 00059 a single TRACE command. For example:

```
CLTRACE 00061
CLTRACE 00062 | TRACE SVC PROGRAM SIO PRINTER |
CLTRACE 00063
```

CLTRACE 00069 SVC traces user SVC interrupts.

CLTRACE 00071 I/O traces user I/O interrupts.

CLTRACE 00073 PROGRAM traces user program interrupts.
CLTRACE 00075 PRO

```
CLTRACE  00077    EXTERNAL  traces user external interrupts.
CLTRACE  00079    EXT

CLTRACE  00082    PRIV      traces all user privileged instructions.

CLTRACE  00085    SIO       traces all user I/O operations including SIO,
CLTRACE  00087              FSIO, TIO, HIO, HDV, and TCH to virtual devices.

CLTRACE  00089    CCW       traces virtual and real CCW's for
CLTRACE  00091              non-spool/non-console device I/O operations. SIO
CLTRACE  00092              tracing must be specified in order to get CCW
CLTRACE  00093              tracing.

CLTRACE  00096    BRANCH    traces all user interrupts and all successful
CLTRACE  00098    BR        branches.

CLTRACE  00100    INSTRUCT  traces all instructions, user interrupts and
CLTRACE  00102    INST      successful branches.

CLTRACE  00105    ALL       traces all instructions, interrupts, successful
CLTRACE  00105              branches, privilege instructions, and user I/O
CLTRACE  00106              operations.

CLTRACE  00109    CSW       provides contents of virtual and real channel
CLTRACE  00109              status words at I/O interrupt. I/O must have been
CLTRACE  00111              specified if CSW is required.

CLTRACE  00114    END       terminates all tracing activity and print a
CLTRACE  00115              termination message.

CLTRACE  00118    TERMINAL  specifies that the tracing is to be done on the
CLTRACE  00120    TERM      terminal (virtual console).

CLTRACE  00122    PRINTER   specifies that the tracing is to be done on a
CLTRACE  00124    P         virtual spooled printer.
CLTRACE  00126    PRT

CLTRACE  00129    BOTH      specifies that tracing is to be done both at the
CLTRACE  00130              terminal and on a virtual spooled printer.

CLTRACE  00133    OFF       halts tracing of the specified activities on both
CLTRACE  00135    OF        terminal and printer.

CLTRACE  00137    NORUN     specifies that for terminal output, program
CLTRACE  00138    NOR       execution is to stop after the trace output has
CLTRACE  00139              completed and that CP console function mode is to
CLTRACE  00140              be entered.

CLTRACE  00142    RUN       specifies that for terminal output, the program is
CLTRACE  00143              to continue execution after the trace output has
CLTRACE  00144              completed and that CP console function mode is not
CLTRACE  00145              to be entered.
```

```
CLTRACE  00148    Responses

CLTRACE  00149    The following  symbols are  used in  the responses  received
CLTRACE  00150    from TRACE:

CLTRACE  00153         vvvvvv   = virtual storage address
CLTRACE  00154         tttttt   = virtual transfer address or new PSW address
CLTRACE  00155         rrrrrr   = real storage address
CLTRACE  00156         xxxxxxxx = virtual instruction, command, etc.
CLTRACE  00157         yyyyyyyy = real instruction, command, etc.
CLTRACE  00158         type     = virtual device name (DASD, TAPE, etc.)
CLTRACE  00159         vaddr    = virtual device address
CLTRACE  00160         raddr    = real device address
CLTRACE  00161         mnem     = mnemonic for instruction, etc.
CLTRACE  00162         int      = interrupt type (SVC, PROG, EXT, I/O)
CLTRACE  00163         code     = interrupt code number (in hexadecimal)
CLTRACE  00164         n        = condition-code number (0, 1, 2, or 3)

CLTRACE  00168    TRACE STARTED

CLTRACE  00170    tracing has been initiated

CLTRACE  00172    TRACE ENDED

CLTRACE  00174    tracing has been suspended


CLTRACE  00179    SIO, TIO, HIO, HDV, FSIO, or TCH


CLTRACE  00181    TCH:

CLTRACE  00183    TCH vvvvvv type vaddr xxxxxxxx CC n


CLTRACE  00185    TIO, HIO or HDV

CLTRACE  00192    r   ¬
CLTRACE  00193    |TIO|
CLTRACE  00194    |HIO| vvvvvv type vadd xxxxxxxx CC n RDEV radd CSW xxxx
CLTRACE  00195    L   ┘


CLTRACE  00197    SIO or FSIO

CLTRACE  00199    SIO vvvvvv type vadd xxxxxxxx CC n RDEV radd CAW vvvvvvvv


CLTRACE  00201    HDV
```

```
CLTRACE  00208    CCW ("IDAL" or "SEEK" line included only if applicable):

CLTRACE  00210    CCW vvvvvv xxxxxxxx xxxxxxxx rrrrrr yyyyyyyy yyyyyyyy
CLTRACE  00212    CCW  IDAL  vvvvvvvv vvvvvvvv  IDAL  00rrrrrr C0rrrrrr
CLTRACE  00214    CCW  SEEK  xxxxxxxx xxxxxx    SEEK  yyyyyyyy yyyy

CLTRACE  00217    INSTRUCTION TRACING:


CLTRACE  00219    Privileged Instruction:

CLTRACE  00221    ::: vvvvvv mnem xxxxxxxx


CLTRACE  00223    Executed Instructions:

CLTRACE  00226    ... vvvvvv EX xxxxxxxx vvvvvv mnem xxxx xxxxxxxx


CLTRACE  00228    All Other Instructions:

CLTRACE  00231    ... vvvvvv mnem xxxxxxxx xxxx


CLTRACE  00233    SUCCESSFUL BRANCH:

CLTRACE  00235    ==> vvvvvv mnem xxxxxxxx tttttt


CLTRACE  00240    INTERRUPT (SVC, PROGRAM, or EXTERNAL)

CLTRACE  00242    *** vvvvvv int code tttttt


CLTRACE  00244    I/O INTERRUPT   (First line  given  only  if  "CSW"  was
CLTRACE  00245         specified):

CLTRACE  00248    CSW R radd xxxxxxxx xxxxxxxx V vadd yyyyyyyy yyyyyyyy
CLTRANSF 00001    *** vvvvvv I/O vadd tttttt CSW xxxx
```

CLTRANSF 00003 **TRANSFER**

CLTRANSF 00006 **Privilege Class:** D, G

CLTRANSF 00008 The TRANSFER command directs a spool file to a specified
CLTRANSF 00009 user's card reader. The file is deleted from the sender's
CLTRANSF 00010 virtual device and is available in the recipient's virtual
CLTRANSF 00011 reader.

CLTRANSF 00020
CLTRANSF 00023 TRANSFER [r userid] {spoolid} [TO] userid
CLTRANSF 00024 TRAN [SYSTEM] {CLASS a}
CLTRANSF 00025 [ALL }
CLTRANSF 00026 [ ]
CLTRANSF 00027

CLTRANSF 00034 userid (for use by Class D users only) this allows the
CLTRANSF 00035 spooling operator to manipulate the spool files
CLTRANSF 00036 for a particular user.

CLTRANSF 00038 SYSTEM (for use by Class D users only) this allows the
CLTRANSF 00039 spooling operator to manipulate all spool files in
CLTRANSF 00040 the system regardless of userid.

CLTRANSF 00042 spoolid specifies the file to be directed to the named
CLTRANSF 00043 userid.

CLTRANSF 00046 CLASS a specifies that all files of the given class are to
CLTRANSF 00049 CL to be transferred. c is a one-character alphameric
CLTRANSF 00050 field with values from A to Z and from 0 to 9.

CLTRANSF 00052 ALL specifies that all spool files are to be
CLTRANSF 00053 transferred

CLTRANSF 00057 TO is an optional reserved keyword. If this key word
CLTRANSF 00061 T is omitted, userid may not be T or TO.

CLTRANSF 00063 userid specifies the user to whom the files are to be
CLTRANSF 00064 directed.

CLTRANSF 00067 **Responses**

CLTRANSF 00069 RDR FILE spoolid TO userid

CLTRANSF 00070 This is the response to the user issuing the TRANSFER
CLTRANSF 00071 command. One such response is given for each file
CLTRANSF 00074 transferred.

```
CLTRANSF 00076     (nnnn) FILES TRANSFERRED
CLTRANSF 00078     (NO   )

CLTRANSF 00080     is the response to the user who issued the TRANSFER command.

CLTRANSF 00082     RDR FILE spoolid FROM userid

CLTRANSF 00085     This is the response to the user receiving the file.
```

CLUNLOCK 00005 **UNLOCK**

CLUNLOCK 00008 **Privilege Class:** A

CLUNLOCK 00009 The UNLOCK command unlocks page frames previously locked by
CLUNLOCK 00010 a LOCK command.

CLUNLOCK 00019
CLUNLOCK 00020 | UNLOCK | {userid fpage lpage} |
CLUNLOCK 00021 | | {VIRT=REAL } |
CLUNLOCK 00022

CLUNLOCK 00027 userid is the user identification of the virtual user
CLUNLOCK 00029 previously specified in a LOCK command.

CLUNLOCK 00031 fpage is the hexadecimal value of the first page to be
CLUNLOCK 00032 unlocked.

CLUNLOCK 00034 lpage is the hexadecimal value of the last page to be
CLUNLOCK 00035 unlocked.

CLUNLOCK 00038 **Note:** for fpage and lpage, only the page numbers are
CLUNLOCK 00038 specified. For example, to unlock USERA's virtual storage
CLUNLOCK 00039 locations X'12000' to X'2C000', the following command should
CLUNLOCK 00040 be given:

CLUNLOCK 00042
CLUNLOCK 00043 | UNLOCK USERA 12 2C |
CLUNLOCK 00044

CLUNLOCK 00051 VIRT=REAL specifies that the page frames normally reserved
CLUNLOCK 00052 for the virtual=real machine are to be released
CLUNLOCK 00052 for normal paging use if the area is not
CLUNLOCK 00054 occupied by a virt=real virtual machine.

CLUNLOCK 00057 **Response**

CLVARY 00001 COMMAND COMPLETE

CLVARY 00005 **VARY**

CLVARY 00008 **Privilege Class:** B

CLVARY 00009 The VARY command allows the operator to mark a device
CLVARY 00011 unavailable for use by a user or the control program.
CLVARY 00012 Devices marked offline are remembered by the VM/370 system
CLVARY 00012 through any virtual machine system failures. A shutdown and
CLVARY 00013 subsequent cold start will assume that all devices are
CLVARY 00014 available.

CLVARY 00023 ┌──────────────────────────────────┐
CLVARY 00024 | VARY | DEV raddr {ONLINE } |
CLVARY 00025 | | {OFFLINE} |
CLVARY 00026 └──────────────────────────────────┘

CLVARY 00032 DEV — is a required reserved word.

CLVARY 00034 raddr — specifies the real address (ccu) of the device.

CLVARY 00036 ONLINE — specifies that a device previously offline is to
CLVARY 00038 ON — be made available.

CLVARY 00040 OFFLINE — specifies that the device is to be made unavailable.
CLVARY 00041 OF

CLVARY 00045 **Response**

CLWNG 00001 COMMAND COMPLETE

CLWNG 00005 **WNG**

CLWNG 00008 **Privilege Class:** A, B

CLWNG 00009 The WNG command is used to transmit high priority messages
CLWNG 00010 to a specified user or to all users. WNG does not wait for
CLWNG 00011 input operations at the terminal to complete, but breaks in
CLWNG 00012 on the line currently being typed and sends the message
CLWNG 00013 immediately.

CLWNG 00022
CLWNG 00023 | WNG | {userid } msg-text |
CLWNG 00024 | W | {OPERATOR{ |
CLWNG 00025 | | {ALL { |
CLWNG 00026

CLWNG 00033 userid specifies the user who is to receive the message.

CLWNG 00036 OPERATOR specifies that the message is to be sent to the
CLWNG 00039 OP primary system operator regardless of his userid.

CLWNG 00042 ALL is for use by the system operator to allow him to
CLWNG 00044 broadcast the message to all users receiving
CLWNG 00045 warning messages.

CLWNG 00048 msg-text specifies the message to be sent.

CLWNG 00051 **Responses**

CLWNG 00053 WNG FROM userid : message-text

CLWNG 00055 This is the response to the users who are to receive the
CLWNG 00056 warning message.

CLSTAR 00008 *

CLSTAR 00011 Privilege Class: ALL

CLSTAR 00013 The * is used to annotate the console sheet with a comment.

CLSTAR 00015
CLSTAR 00016
CLSTAR 00017

*	any-comment

CLSTAR 00021 Responses

CLBATCH1 00001 None.

CLBATCH1 00004

APPENDIXES

CLBATCH1 00011 **APPENDIX A: CMS BATCH FACILITY**

CLBATCH1 00013 The CMS Batch Facility is a variation of the CMS system
CLBATCH1 00014 which is controlled by an established input stream rather
CLBATCH1 00015 than interactive communication with a remote terminal. The
CLBATCH1 00015 remote terminal is required to initiate the CMS Batch
CLBATCH1 00017 virtual machine. Any user may submit a job stream to the CMS
CLBATCH1 00018 Batch machine, and, while the job stream is executing,
CLATTKEY 00001 continue regular CMS processing.

CLATTKEY 00011 # APPENDIX B: TERMINAL CONSIDERATIONS

CLATTKEY 00015 ## ATTENTION KEY HANDLING

CLATTKEY 00016 To simulate the functions of the central operator's console
CLATTKEY 00017 REQUEST and CANCEL keys, the terminal user may use the ATTN
CLATTKEY 00018 key (or its equivalent). Refer to Figures 31-37 for the
CLATTKEY 00019 positions of the keys used to perform the ATTN function on
CLATTKEY 00020 the various terminal types. Each time the user presses the
CLATTKEY 00021 ATTN key, CP types an exclamation point (!)s, some "idle" or
CLATTKEY 00022 nonprintable character, and a carriage return to indicate
CLATTKEY 00023 that the attention has been recognized. In the following
CLATTKEY 00024 chart, "one attention" means that the user presses the ATTN
CLATTKEY 00025 key and waits for the carriage to return; "two attentions"
CLATTKEY 00026 means that the user presses the ATTN key and while the
CLATTKEY 00027 exclamation point and the subsequent unprintable characters
CLATTKEY 00029 are being received at the terminal, presses ATTN again. More
CLATTKEY 00029 than two attentions is always the same as two except as
CLATTKEY 00030 noted below.

CLATTKEY 00032 The action taken by CP when an attention is recognized
CLATTKEY 00033 depends upon the user's operating mode and the state of his
CLATTKEY 00034 terminal. CP mode indicates that the user is conversing with
CLATTKEY 00034 the Control Program. VM mode indicates that the user is
CLATTKEY 00035 conversing with the operating system that is running in his
CLATTKEY 00036 virtual machine. The user's terminal is always considered to
CLATTKEY 00038 be in one of four states:

CLATTKEY 00043 - S1 the terminal is idle and its keyboard is locked.
CLATTKEY 00046 - S2 the terminal is receiving output from the computer.
CLATTKEY 00048 - S3 the terminal keyboard is unlocked for input but the
CLATTKEY 00049 user has not typed any characters.
CLATTKEY 00051 - S4 the terminal keyboard is unlocked for input and one or
CLATTKEY 00053 more characters have been typed.

CLATTKEY 00056 Figure 30 shows the action taken when a user presses the
CLATTKEY 00057 ATTN key or its equivalent.

MODE	STATE	ATTN	ACTION
CP	S1- terminal idle,keyboard locked	Any*	keyboard unlocked for input to CP.
	S2- terminal receiving output	Any*	active CE command terminated. keyboard unlocked for input to CP.
	S3- keyboard unlocked, no characters typed	Any*	attention interrupt made pending. virtual machine started at address in current PSW.
	S4- keyboard unlocked,one or more characters typed	Any*	input line cancelled. keyboard unlocked for input to CP.
VM	S1- terminal idle,keyboard locked	1	attention interrupt made pending.**
		2	keyboard unlocked for input to CP. any output being printed at the terminal is deleted.
	S2- terminal receiving output	1	attention interrupt made pending. any output being printed at the terminal is deleted.
		2	keyboard unlocked for input to CP.

* When the terminal is in CP mode, two or more attentions are treated the same as one attention.

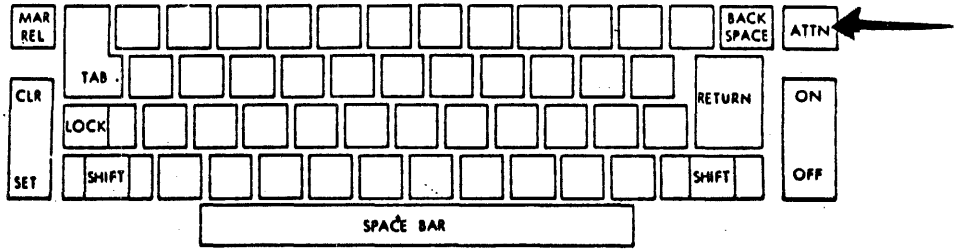
** If the user is running CMS in his virtual machine, he may use this as a means of stacking CMS commands. If a CMS command is executing and he wishes to enter a command to execute when the first one completes, he may enter the second one when the Attention interrupt is recognized.

Figure 30. Result of Attention Key Activation (Part 1 of 2)

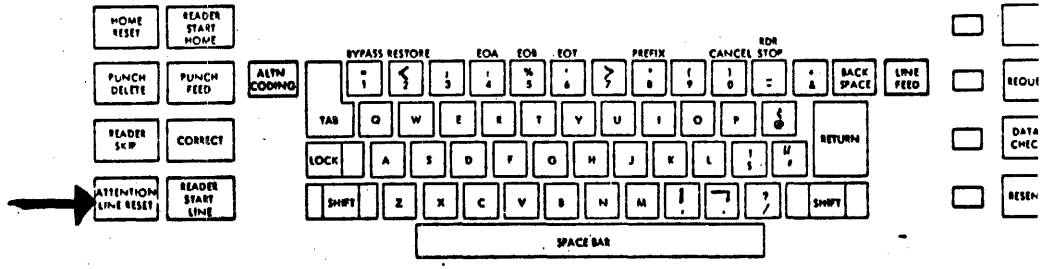
MODE	STATE	ATTN	ACTION
	S3- keyboard unlocked, no characters typed	1	unit exception status. control returned to virtual machine.
		2	device end status. keyboard unlocked for input to CP.
	S4- keyboard unlocked,one or more characters typed	1	unit exception status control returned to virtual machine. Any characters typed in are ignored.
		2	device end status any characters typed are saved in the virtual machine's memory. control returned to virtual machine.

Figure 30. Result of Attention Key Activation (Part 2 of 2)

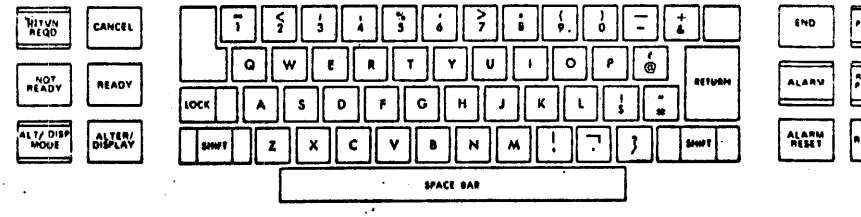
CLATKEQV 00003 ATTENTION KEY EQUIVALENTS



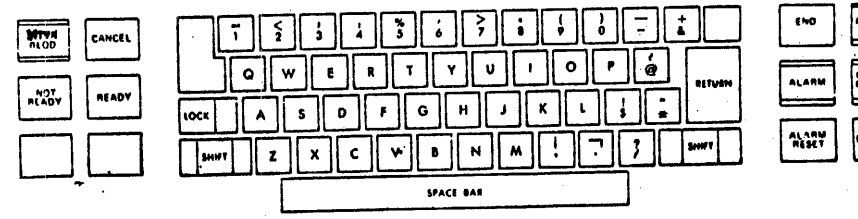
CLATKEQV 00008 Figure 31. IBM 2741 and IBM 1050 Keytcard ATTN key



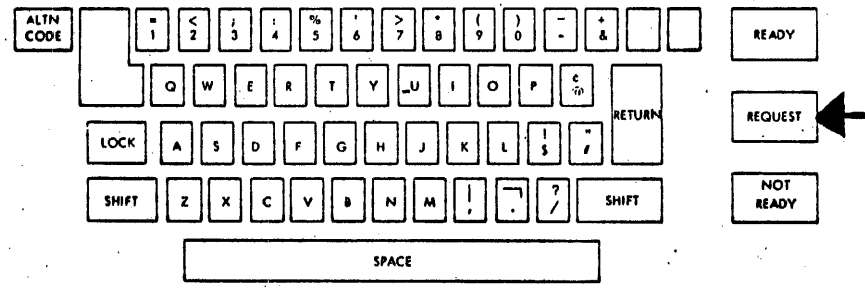
CLATKEQV 00012 Figure 32. IBM 1052 Keyboard ATTENTION LINE RESET key



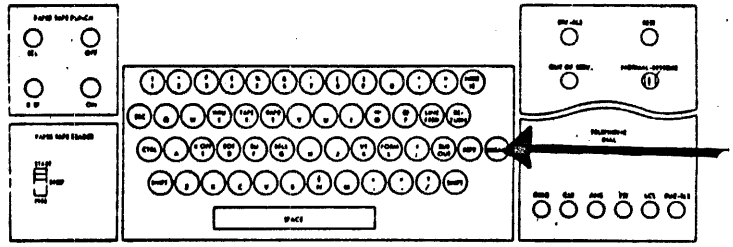
CLATKEQV 00017 Figure 33. IBM 3330 Model 1 and IBM 3335 Keyboard REQUEST key



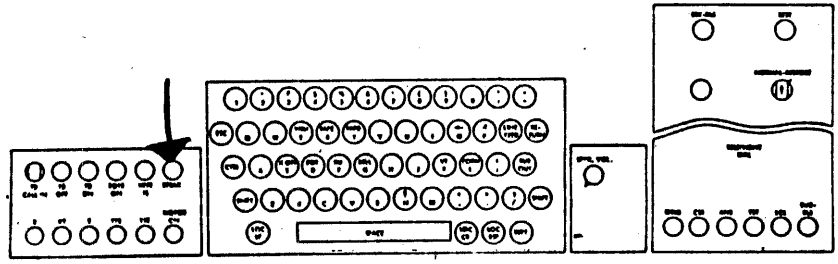
CLATKEQV 00021 Figure 34. IBM 3210 Model 2 Keybcard REQUEST key



CLATKEQV 00025 Figure 35. IBM 2150 Keyboard REQUEST key



CLATKEQV 00030 Figure 36. Teletype* Model 33 Keyboard BREAK key



CLATKEQV 00034 Figure 37. Teletype* Model 35 Keyboard BREAK key

CLATKEQV 00037 ---
CLATKEQV 00039 * Trademark of Teletype Corporation, Skokie, Illinois

CLLNEDIT 00010 SPECIAL VM/370 LOGICAL LINE EDITING CHARACTERS

CLLNEDIT 00012 VM/370 normally edits all input lines entered in either CP
CLLNEDIT 00013 mode or lines entered to a system in a virtual machine. The
CLLNEDIT 00014 editing of input for the virtual machine may be turned off
CLLNEDIT 00015 by the CP command SET LINEDIT OFF. All of the editing
CLLNEDIT 00016 characters are removed from the input line before it is
CLLNEDIT 00017 passed to the user's virtual machine. The four editing
CLLNEDIT 00018 functions defined in CP are described below.

CLLNEDIT 00021 Logical Character Delete - The logical Character Delete
CLLNEDIT 00021 character causes the preceding character to be deleted
CLLNEDIT 00023 from the input line. The Character Delete character
CLLNEDIT 00023 can be used to delete a line end character and a line
CLLNEDIT 00025 delete character.

CLLNEDIT 00028 Logical Line End - The logical Line End character causes a
CLLNEDIT 00028 new line character, hexadecimal '15', to be inserted
CLLNEDIT 00029 into the input line. The new line character is treated
CLLNEDIT 00030 as the end of a logical line by both the CP and CMS
CLLNEDIT 00031 input processors.

CLLNEDIT 00034 Logical Line Delete - The logical Line Delete character
CLLNEDIT 00035 causes the current logical line to be deleted from the
CLLNEDIT 00036 input line. If the Logical Line Delete character
CLLNEDIT 00036 immediately follows a new line character, both the
CLLNEDIT 00037 current and previous logical lines are deleted. The
CLLNEDIT 00038 current logical line begins immediately following the
CLLNEDIT 00039 previous new line character or at the beginning of the
CLLNEDIT 00039 physical line if no new line character is present in
CLLNEDIT 00040 the line.

CLLNEDIT 00042 Logical Escape - The logical Escape character causes the
CLLNEDIT 00043 character following it to be ignored as an input line
CLLNEDIT 00043 editing character. This allows any of the above
CLLNEDIT 00044 editing characters to be interpreted literally instead
CLLNEDIT 00045 of as an editing character without redefining that
CLLNEDIT 00047 editing character. An ESCAPE character appearing as the
CLLNEDIT 00047 last character of a line is removed from the line and
CLLNEDIT 00048 ignored. Any of the logical line editing characters
CLLNEDIT 00049 including the logical escape character may be entered
CLLNEDIT 00050 as data if preceded by an escape character.

CLLNEDIT 00053 The default input line editing characters for the system are
CLLNEDIT 00053 listed in Figure 38. An installation may redefine these
CLLNEDIT 00055 default characters for an individual user in the user
CLLNEDIT 00056 directory. A user may alter the character associated with
CLLNEDIT 00057 each function, or delete the function entirely for the
CLLNEDIT 00058 duration of the current terminal session by using the CP
CLLNEDIT 00059 TERMINAL Command.

Function	Character
Logical Character Delete	@
Logical Line End	#
Logical Line Delete	¢
Logical Escape	"

Figure 38. Logical Line Editing Characters

Examples of the Use of Line Editing Characters

Typed: REED@@ADCARD

Result: READCARD

Typed: QUERY YS@@ux@@USERS#¢QUERY NAMES

Result: QUERY NAMES

Typed: MSG ALL FILE ""MACLIB"" IS BEING UPIATED

Result: MSG ALL FILE "MACLIB" IS BEING UPDATED

Typed: QUERY NAMES # QUERY FILES # QUERY USERS # ¢¢¢

Result: null line

Typed: Q NAMES # Q FILES #@Q USERS ¢

Result: Q NAMES.

Typed: SET LINEDIT OFF # REED@@ADCARD

Result: SET LINEDIT OFF
READCARD

Note: The entire physical line is scanned before any command is executed; therefore, the second logical line would have been scanned before the command in the first logical line was executed.

Typed: SET LINEDIT OFF (carriage return)
REED@@ADCARD

Result: REED@@ADCARD

```
CLLNEDIT 00133              Line editing  has been  turned off  by the  command
CLLNEDIT 00135              SET LINEDIT OFF

CLLNEDIT 00138     Typed:   01632  This line of texasaxt is to te inserted
CLLNEDIT 00139              in a file at seq no 015a62a3a32

CLLNEDIT 00141     Result:  01632  This line of text is to be inserted in a
CLLNEDIT 00142              file at seq no 01632
```

APPENDIX C: DISK DETERMINATION (Filemode Management)

The table in Figure 39 relates CMS commands, method of specifying filemode, and the criteria used in choosing a disk directory for reading and writing. Symbols used in the table are:

Symbol	Meaning
command	CMS command name
fm	explicit mode letter can be specified
=	Write disk to Read disk
*	refer to all disks in a set search order
d	default mode: let system determine where-to-go
-	null mode; unable to specify mode letter in this command
Reading	the criteria for choosing the disk to be read from
N/A	not applicable, command does not cause any reading to be done
fm	read from the specified disk
*R	refer to all disks in the standard search order
1R	read only from the primary disk
Writing	indicates the criteria for selecting the disk to write to
N/A	not applicable, command does not cause any writing to be done
fm	write to the specified disk
R	write to the disk from where a file was read, or its parent
*W	choose any R/W disk in the standard search pattern
1W	attempt to write onto the primary disk

Command	Filemode	Reading	Writing
ACCESS	fm	fm	N/A
ASSEMBLE	-	*R	R,1W,*W
BASIC	-	*R	R,1W
*COBOL	-	*R	R,1W,*W
COMPARE	fm	fm	N/A
	*	*R	N/A
	d	1R	N/A
*CONVERT	fm	fm	fm
COPYFILE	fm	fm	fm
	=	N/A	R
CP	-	N/A	N/A
DEBUG	-	N/A	N/A
DISK DUMP	fm	fm	N/A
	*	*R	N/A
	d	1R	N/A
LOAD	-	N/A	1W
EDIT	fm	fm	R,1W
	*	*R	R,1W
	d	1R	R,1W
ERASE	fm	N/A	fm
	*	N/A	*W
	d	N/A	1W
EXEC	-	*R	N/A
FILEDEF	fm	fm	fm
	d	1R	1W
	*	*R	R,1W
*FORTGI	-	*R	R,1W
*FORTHX	-	*R	R,1W

* IBM Program Products

Figure 39. Disk Determination (Part 1 of 3)

Command	Filemode	Reading	Writing
GENDIRT	–	N/A	N/A
GENMOD	fm	N/A	fm
	*	N/A	1W
	d	N/A	1W
GLOBAL	–	N/A	N/A
*GOFORT	–	*R	R,1W
INCLUDE	–	*R	1W
LISTFILE	fm	fm	1W
	*	*R	1W
	d	1R	1W
LOAD	–	*R	1W
LOADMOD	–	*R	N/A
	fm	fm	N/A
MACLIB	–	1R	1W
MODMAP	–	*R	N/A
MOVEFILE	fm	fm	fm
*PLIOPT	–	*R	R,1W,*W
PRINT	fm	fm	N/A
	d	1R	N/A
	*	*R	N/A
PUNCH	fm	fm	N/A
	d	1R	N/A
	*	*R	N/A
QUERY	–	N/A	N/A
READCARD	fm	N/A	(fm)
	–	N/A	fm
	d	N/A	1W
	*	N/A	1W
RELEASE	–	cuu	N/A
	fm	fm	N/A

* IBM Program Products

Figure 39. Disk Determination (Part 2 of 3)

Command	Filemode	Reading	Writing
RENAME	fm	fm	fm
	*	*R	R
	=	N/A	R
RUN	fm	fm	N/A
	*	*R	N/A
	–	*R	N/A
SET	–	N/A	N/A
SORT	fm	fm	fm
	*	*R	R,1W
START	–	N/A	N/A
STATE	fm	fm	N/A
	*	*R	N/A
	d	*R	N/A
SVCTRACE	–	N/A	N/A
SYNONYM	fm	fm	N/A
	*	*R	N/A
TAPE DUMP	fm	fm	N/A
	*	*R	N/A
	d	*R	N/A
LOAD	fm	N/A	fm
	d	N/A	1W
	*	N/A	1W
SCAN	–	N/A	N/A
SKIP	–	N/A	N/A
TAPPDS	fm	N/A	fm
	d	N/A	1W
TXTLIB	–	*R	1W
TYPE	fm	fm	N/A
	*	*R	N/A
	d	1R	N/A
UPDATE	–	*R	R,1W

Figure 39. Disk Determination (Part 3 of 3)

APPENDIX D: RESERVED FILETYPE DESCRIPTIONS

The following table shows filetypes that have special uses in CMS.

FILETYPE	COMMAND	USAGE	FILENAME	FORMAT RECFM	FORMAT LRECL	CONTENTS
ASSEMBLE	ASSEMBLE	input	fn	F	80	Assembler lan Source statem
BASIC	BASIC	input	fn	F	80	BASIC languag source statem
BASDATA	BASIC execution	execution-time files	fn	U	≤3440	user input an output files
CMSUT1	READCARD	inter-mediate work file	READ	F	80	
	COPYFILE		COPYFILE			
	DISK		DISKLOAD			
	TAPE		TAPELOAD			
	UPDATE		fn			
	INCLUDE		DMSLDR			
	LOAD		DMSLDR			
	MACLIB		DMSLBM			
COBOL	COBOL*	input	fn	F	80	COBOL source ments
COPY	MACLIB	input	fn	F	80	*COPY control and macro definitions
EXEC	EXEC	input	fn	F	80	EXEC statemen
	LISTFILE	output	CMS			
FORTRAN	FORTG1*	input	fn	F	80	FORTRAN sourc statements
	FORTHX*					
	GOFORT*					
FTnnF001	FORTRAN execution	input/ output	fn			user input an cutput files
LISTING	ASSEMBLE	output	fn	F	121	processor pri output
	FORTRAN* compilers	output	fn			
	COBOL*	output	fn			
	PLIOPT*	output	fn			

* IBM Program Products

Figure 40. Reserved Filetypes (Part 1 of 2)

FILETYPE	COMMAND	USAGE	FILENAME	FORMAT	CONTENTS
MACLIB	GLOBAL MACLIB	library MACLIB	fn fn	library contains dictionary and members	macro definitions macro definitions
MACRO	MACLIB	input	fn	F 80	macro definitions
MAP	INCLUDE LOAD MACLIB TXTLIB	output output output output	LOAD LOAD fn fn		module map module map library map library map
MEMO			fn	F 80	
MODULE	GENMOD LOADMOD MODMAP	output input input	fn fn fn	V	non-relocatable object file
PLI	PLIOPT*	input	fn	F	PL/I source statements
SYNONYM	SYNONYM	reference	fn	F 80	command name synonyms
SYSUT1,2,3	ASSEMBLE COBOL* PLIOPT*	work work work	fn fn fn	V	
SYSUT4	COBOL*	work	fn	F 80	
SYSUT5	COBOL*	output	fn	F	relocatable object code
TEXT	ASSEMBLE COBOL* INCLUDE LOAD PLIOPT* TXTLIB FORTRAN* compilers	output output input input output input output	fn fn fn fn fn fn fn	F 80	object code object code object code object code object code object code object file
TXTLIB	GLOBAL TXTLIB	library output	fn fn	library contains dictionary and members	object decks
UPDATE	UPDATE	input	fn	F 80	UPDATE control cards
UPDLOG	UPDATE	output	fn	F	UPDATE log

* IBM Program Products

Figure 40. Reserved Filetypes (Part 2 of 2)

```
CLSAMPTM 00006    APPENDIX E: SAMPLE TERMINAL SESSION



CLSAMPTM 00009    #
CLSAMPTM 00011    vm/370 online xquvtz noxudj
CLSAMPTM 00012                                  -Attention key hit
CLSAMPTM 00014    login smith mask
CLSAMPTM 00016    ENTER PASSWORD:
CLSAMPTM 00018    *********                     -The password is typed in over
CLSAMPTM 00019                                   the mask
CLSAMPTM 00021    LOGMSG - 12:37:20 04/13/72
CLSAMPTM 00022    *  VM/370 WILL BE UP 24 HOURS A DAY
CLSAMPTM 00023    *  MAINTENANCE PERIODS WILL BE ANNOUNCED
CLSAMPTM 00024    *  48 HOURS IN ADVANCE

CLSAMPTM 00026    LOGON AT 14:26:31 EST THURSDAY 04/13/72

CLSAMPTM 00028    ipl cms

CLSAMPTM 00030    CMS VERSION 1.0 4/13/72 12:38:10

CLSAMPTM 00034    access 191 a (noprof          -The user wishes to gain
CLSAMPTM 00035                                   access to the CMS system but
CLSAMPTM 00035                                   does not want his file,
CLSAMPTM 00037                                   PROFILE EXEC to be executed.

CLSAMPTM 00040    R; T= 0.09/1.25 12:38:40
CLSAMPTM 00042    set rdymsg smsg               -The user doesn't want to
CLSAMPTM 00042                                   receive the long READY
CLSAMPTM 00043                                   message after each command
CLSAMPTM 00045    R;                            -short ready message
CLSAMPTM 00047    set blip blip                 -The user wishes to have a
CLSAMPTM 00047                                   printed record of the CPU
CLSAMPTM 00048                                   time used.
CLSAMPTM 00050    R;

CLSAMPTM 00053    edit testprog assemble        -A new file is to be created
CLSAMPTM 00053                                   with a file identifier of
CLSAMPTM 00054                                   TESTPROG ASSEMBLE A1, it will
CLSAMPTM 00054                                   have the default
CLSAMPTM 00054                                   characteristics for filetype
CLSAMPTM 00055                                   ASSEMBLE

CLSAMPTM 00059    NEW FILE:                     -The file does not currently
CLSAMPTM 00060                                   exist and will be created

CLSAMPTM 00063    EDIT:                         -The EDIT environment is
CLSAMPTM 00064                                   entered

CLSAMPTM 00068    input                         -The user wishes to enter the
CLSAMPTM 00068                                   INPUT environment to type in
```

```
CLSAMPTM 00069                                    the program

CLSAMPTM 00072
CLSAMPTM 00074    INPUT:                          -INPUT environment is entered

CLSAMPTM 00077    testprog csect                  -Text of program
CLSAMPTM 00078             using *,15
CLSAMPTM 00079             save  (14,12)
CLSAMPTM 00080             st    13,savearea+4
CLSAMPTM 00081             bal   12,around
CLSAMPTM 00082             using savearea,13
CLSAMPTM 00083    savearea ds    18f
CLSAMPTM 00084    around   equ   *
CLSAMPTM 00085             st    12,8(13)
CLSAMPTM 00086             lr    13,12
CLSAMPTM 00087             wto   'this is output of a sample'
CLSAMPTM 00088             wto   '*******assembler**********'
CLSAMPTM 00089             wto   '*******language**********'
CLSAMPTM 00090             wto   '*******program***********'
CLSAMPTM 00091             l     13,4(13)
CLSAMPTM 00092             xr    15,15
CLSAMPTM 00093             return (14,12),rc=(15)
CLSAMPTM 00094             end

CLSAMPTM 00099                                    -Carriage return hit to
CLSAMPTM 00103                                     return user to EDIT
CLSAMPTM 00105                                     environment

CLSAMPTM 00109    EDIT:                           -EDIT environment entered

CLSAMPTM 00113    file                            -The file is saved on disk

CLSAMPTM 00116    R;                              -The   EDIT   command   has
CLSAMPTM 00117                                     completed.    Control    is
CLSAMPTM 00118                                     returned to CMS

CLSAMPTM 00122    global maclib osmacro           -The program to be assembled
CLSAMPTM 00122                                     contains macros which are
CLSAMPTM 00122                                     included   in   the   macro
CLSAMPTM 00123                                     library,  OSMACRO  MACLIB.
CLSAMPTM 00123                                     This command specifies that
CLSAMPTM 00124                                     the library is to be searched
CLSAMPTM 00125                                     for all macros

CLSAMPTM 00128    R;

CLSAMPTM 00131    assemble testprog               -The   System   Assembler   is
CLSAMPTM 00131                                     invoked   to   assemble   the
CLSAMPTM 00131                                     source statements included in
CLSAMPTM 00133                                     the file, TESTPROG ASSEMBLE

CLSAMPTM 00135    BLIPBLIPBLIP                    -This shows that  6 seconds of
CLSAMPTM 00136                                     CPU time have  elapsed during
CLSAMPTM 00137                                     the assembly.
```

```
CLSAMPTM 00141    #ASSEMBLER DONE

CLSAMPTM 00145    #NO STATEMENTS FLAGGED IN THIS ASSEMBLY

CLSAMPTM 00147    R;

CLSAMPTM 00150    listfile  testprog * *          -Information on all files with
CLSAMPTM 00150                                     a filename of  TESTPROG is to
CLSAMPTM 00151                                     be typed

CLSAMPTM 00154    FILENAME   FILETYPE  FMODE
CLSAMPTM 00155    TESTPROG   ASSEMBLE   A1      -source statements
CLSAMPTM 00156    TESTPROG   LISTING    A1      -listing of program
CLSAMPTM 00157    TESTPROG   TEXT       A1      -object code

CLSAMPTM 00159    R;

CLSAMPTM 00163    print testprog listing          -The    listing    of    source
CLSAMPTM 00164                                     statements, generated object
CLSAMPTM 00164                                     code and diagnostic messages
CLSAMPTM 00164                                     is   to   be   spooled   to   the
CLSAMPTM 00165                                     printer

CLSAMPTM 00168    R;
CLSAMPTM 00170    load testprog                   -The file TESTPROG  TEXT which
CLSAMPTM 00170                                     contains   the   object   code
CLSAMPTM 00170                                     created   by   the   ASSEMBLE
CLSAMPTM 00171                                     command is to  be loaded into
CLSAMPTM 00172                                     storage

CLSAMPTM 00175    R;

CLSAMPTM 00177    genmod testprog                 -An absolute image  file is to
CLSAMPTM 00178                                     be    created    using    all
CLSAMPTM 00178                                     currently  loaded code.  The
CLSAMPTM 00179                                     filename  will   be  TESTPROG
CLSAMPTM 00180                                     MODULE A1

CLSAMPTM 00183    R;

CLSAMPTM 00185    listfile testprog * *           -Information on all files with
CLSAMPTM 00186                                     a filename of  TESTPROG is to
CLSAMPTM 00187                                     be typed at the terminal

CLSAMPTM 00190    FILENAME   FILETYPE  FMODE
CLSAMPTM 00191    TESTPROG   ASSEMBLE   A1      -source statements
CLSAMPTM 00192    TESTPROG   LISTING    A1      -listing of program
CLSAMPTM 00193    TESTPROG   TEXT       A1      -object code
CLSAMPTM 00194    TESTPROG   MODULE     A1      -load module file

CLSAMPTM 00199    R;

CLSAMPTM 00202    modmap testprog                 -The user  wishes to  type the
```

```
CLSAMPTM 00203                                  contents of the LOADMAP file

CLSAMPTM 00207    TESTPROG   20000
CLSAMPTM 00208    SUSREF     00600
CLSAMPTM 00209    NUCON      00000

CLSAMPTM 00213    R;

CLSAMPTM 00216    loadmod   testprog            -The absolute image module is
CLSAMPTM 00217                                  to be brought into storage

CLSAMPTM 00220    R;

CLSAMPTM 00223    start                         -The loaded program is to
CLSAMPTM 00224                                  begin execution

CLSAMPTM 00228    EXECUTION BEGINS...           -The program has begun
CLSAMPTM 00229                                  execution

CLSAMPTM 00232    *THIS IS OUTPUT OF A SAMPLE   -output created by the program
CLSAMPTM 00233    ********ASSEMBLER**********    which is executing
CLSAMPTM 00234    ********LANGUAGE**********
CLSAMPTM 00235    ********PROGRAM***********

CLSAMPTM 00239    R;

CLSAMPTM 00241    type sortd exec               -The specified file is to be
CLSAMPTM 00242                                  printed at the terminal
CLSAMPTM 00243                                  typewriter.

CLSAMPTM 00246    *EXAMPLE OF THE USE OF EXEC PROCEDURES

CLSAMPTM 00248    *THE FOLLOWING EXEC PROCEDURE WILL SORT A GIVEN FILE IN
CLSAMPTM 00249    *DESCENDING ORDER AND EITHER CREATE A NEW FILE CONTAINING
CLSAMPTM 00250    *THE SORTED OUTPUT OR REPLACE THE INPUT FILE WITH THE
CLSAMPTM 00251    *ONE CONTAINING THE OSRTED OUTPUT. FORMAT OF INPUT COMMAND IS:
CLSAMPTM 00254    *SORTD FN1 FT1 FM1 (*R*       )
CLSAMPTM 00255    *                  (FN2 FT2 FM2)
CLSAMPTM 00256    **R* INDICATES THAT THE INPUT FILE IS TO BE REPLACED.  IF
CLSAMPTM 00257    *FN2 FT2 FM2 IS SPECIFIED, THE SORTED OUTPUT IS PLACED INTO
CLSAMPTM 00258    *A FILE OF THAT NAME.


CLSAMPTM 00265    &CONTROL OFF                  -Suppress typing of CMS
CLSAMPTM 00267                                  commands as they are executed

CLSAMPTM 00270    RENAME &1 &2 &3 SORTDOWN SYSUT1 A1

CLSAMPTM 00272                                  -Rename the input file to the
CLSAMPTM 00273                                  file identifier expected in
CLSAMPTM 00274                                  the conversion program

CLSAMPTM 00278    LOAD SORTDOWN (START          -SORTDOWN is a program which
CLSAMPTM 00278                                  converts every character in
```

```
CLSAMPTM 00280                                  the file to its complement.
CLSAMPTM 00280                                  It uses a file named SORTDOWN
CLSAMPTM 00280                                  SYSUT1 as input and creates
CLSAMPTM 00280                                  an output file named SORTDOWN
CLSAMPTM 00281                                  SYSUT2 A1 containing the
CLSAMPTM 00282                                  converted records.

CLSAMPTM 00285    RENAME SORTDOWN SYSUT1 A1 &1 &2 &3

CLSAMPTM 00287    SORT SORTDOWN SYSUT2 A1 SORTDOWN SYSUT1 A1

CLSAMPTM 00289                                 -The standard SORT supplied
CLSAMPTM 00290                                  with CMS sorts a file into
CLSAMPTM 00291                                  ascending order. Since every
CLSAMPTM 00291                                  character in the file has
CLSAMPTM 00292                                  been converted to its
CLSAMPTM 00292                                  complement, the file will be
CLSAMPTM 00294                                  sorted into descending order

CLSAMPTM 00297    ERASE SORTDOWN SYSUT2 A1

CLSAMPTM 00300    LOAD SORTDOWN (START          -Each record in the sorted
CLSAMPTM 00300                                  file is converted back to its
CLSAMPTM 00301                                  original form

CLSAMPTM 00304    ERASE SORTDOWN SYSUT1 A1      -Erase work file

CLSAMPTM 00307    &IF &4 NE *R* &GOTO -A        -Determine if existing file is
CLSAMPTM 00308                                  to be replaced

CLSAMPTM 00311    ERASE &1 &2 &3                -If existing file is to be
CLSAMPTM 00312                                  replaced, erase existing file
CLSAMPTM 00312                                  and rename the work file
CLSAMPTM 00313                                  containing the sorted records
CLSAMPTM 00313                                  to the file identifier of the
CLSAMPTM 00314                                  original file

CLSAMPTM 00317    RENAME SORTDOWN SYSUT2 A1 &1 &2 &3

CLSAMPTM 00319                                 -Rename the workfile
CLSAMPTM 00319                                  containing the sorted file to
CLSAMPTM 00320                                  the original file identifier.

CLSAMPTM 00323    &EXIT                         -Return to CMS

CLSAMPTM 00326    -A RENAME SORTDOWN SYSUT2 A1 &5 &6

CLSAMPTM 00328                                 -Rename the workfile
CLSAMPTM 00329                                  containing the sorted records
CLSAMPTM 00329                                  to the specified new file
CLSAMPTM 00330                                  identifier

CLSAMPTM 00333    &EXIT
```

```
CLSAMPTM 00335    R;

CLSAMPTM 00337    Example of Use Of The Preceding EXEC Procedure

CLSAMPTM 00339    SORTD INPUTTO SORT A1 OUTPUTFR SORT A1

CLSAMPTM 00342                                   -The command name EXEC need
CLSAMPTM 00343                                    not be entered. File INPUTTO
CLSAMPTM 00343                                    SORT A1 is to be sorted into
CLSAMPTM 00344                                    descending order and the
CLSAMPTM 00344                                    sorted output is to be placed
CLSAMPTM 00344                                    in a new file named OUTPUTFR
CLSAMPTM 00345                                    SORT A1.

CLSAMPTM 00348    EXECUTION BEGINS...

CLSAMPTM 00350    DMSSRT604R ENTER SORT FIELDS:

CLSAMPTM 00352    30 34

CLSAMPTM 00354    EXECUTION BEGINS...

CLSAMPTM 00356    R;

CLSAMPTM 00359    SORTD INPUTTO SORT A1 *R*      -File INPUTTO SORT A1 is to be
CLSAMPTM 00359                                    sorted into descending order
CLSAMPTM 00359                                    and the sorted output is to
CLSAMPTM 00361                                    replace the original file.

CLSAMPTM 00364    EXECUTION BEGINS...

CLSAMPTM 00366    DMSSRT604R ENTER SORT FIELDS:

CLSAMPTM 00368    30 34

CLSAMPTM 00370    EXECUTION BEGINS...

CLSAMPTM 00372    R;


CLSAMPTM 00375    *This is an example of the use of the LISTFILE command to
CLSAMPTM 00376    *create an EXEC procedure.  If the user has many files
CLSAMPTM 00377    *with a filetype of ASSEMBLE which he wants punched so
CLSAMPTM 00378    *that he may transfer them to another user, he could do
CLSAMPTM 00380    *this in two ways:

CLSAMPTM 00382    *1. He could enter the command:

CLSAMPTM 00384    *        DISK DUMP filename ASSEMBLE

CLSAMPTM 00387    *   for each file.  This could be a very time consuming
CLSAMPTM 00389    *   procedure if many files are involved.

CLSAMPTM 00391    *2. He could enter the command:
```

```
CLSAMPTM 00393              LISTFILE * ASSEMBLE * (EXEC)

CLSAMPTM 00395     R;


CLSAMPTM 00398     *This would create  a file  named CMS  EXEC with  records of
CLSAMPTM 00400     *the following format:

CLSAMPTM 00402     *    &1 &2 filename ASSEMBLE filemode

CLSAMPTM 00405     *One record will  be created for  each file with  a filetype
CLSAMPTM 00406     *of ASSEMBLE.  He could then  execute the following command:

CLSAMPTM 00409     *CMS DISK DUMP
CLSAMPTM 00410     *which will substitute DISK for &1 and DUMP for &2
CLSAMPTM 00411     *which creates a DISK DUMP command for each file.
CLSAMPTM 00413     *Before doing  this he  issues a  command  tc indicate  that
CLSAMPTM 00414     *all files in  his virtual  punch are  to be  transferred to
CLSAMPTM 00415     *another user.
CLSAMPTM 00417     spool punch to jones
CLSAMPTM 00418     cms disk dump

CLSAMPTM 00421     DISK DUMP PROG1 ASSEMBLE A1    -each command  in  the  file
CLSAMPTM 00422     PCH FILE 0001 to JONES           CMS EXEC  is  printed  as  it
CLSAMPTM 00423     DISK DUMP PROG2 ASSEMBLE A1      is executed.  The line  after
CLSAMPTM 00424     PCH FILE 0002 TO JONES           the command  indicates  that
CLSAMPTM 00425     DISK DUMP PROG3 ASSEMBLE A1      the file  is  now  in  the
CLSAMPTM 00426     PCH FILE 0003 TO JONES           virtual punch  of user  jones
CLSAMPTM 00427     DISK DUMP PROG4 ASSEMBLE A1      who may  issue  a  DISK  LOAD
CLSAMPTM 00428     PCH FILE 0004 TO JONES           command to  restore it  to  a
CLSAMPTM 00429     DISK DUMP PROG5 ASSEMBLE A1      disk file.
CLSAMPTM 00431     PCH FILE 0005 TO JONES
CLSAMPTM 00434     R;

CLSAMPTM 00437     CLOSE PUNCH                    -the files are  now queued for
CLSAMPTM 00438                                     input for user jones.

CLSAMPTM 00440     logout                         -the user logs out from the CP
CLSAMPTM 00441                                     system.

CLSAMPTM 00444     CONNECT=00:20:15 VIRTCPU=03:56:20 TOTCPU=05:20:30
CLCOMPAT 00001     LOGOFF AT 13:47:11 EST FRIDAY 04/13/72
```

CLCOMPAT 00005 # APPENDIX F: COMPATIBILITY OF VM/370 WITH CP-67/CMS

CLCOMPAT 00007 Certain commands familiar to CP-67 users are not supported
CLCOMPAT 00008 in VM/370. The functions available under CP-67 do, however,
CLCOMPAT 00009 have equivalents in VM/370 where appropriate. The following
CLCOMPAT 00010 is a list of all CP-67 console functions, with operand
CLCOMPAT 00011 variations, showing the incompatibilities with VM/370.
CLCOMPAT 00012 Functions listed under CP-67 3.1 and not under VM/370
CLCOMPAT 00013 indicated that the syntax and function are identical. There
CLCOMPAT 00014 are, of course, additional functions available with the
CLCOMPAT 00015 VM/370 commands, as shown in the "Commands" section of this
CLCOMPAT 00016 book.

CLCOMPAT 00019 Status: A - CP/67 Syntax accepted in VM/370
CLCOMPAT 00022 R - CP/67 Syntax not accepted - Functionally
CLCOMPAT 00022 supported by another command
CLCOMPAT 00023 format.
CLCOMPAT 00025 N - CP/67 Syntax not accepted - function not
CLCOMPAT 00026 supported in VM/370.

CLCOMPAT 00039
CLCOMPAT 00040
CLCOMPAT 00041
CLCOMPAT 00042
CLCOMPAT 00043
CLCOMPAT 00044
CLCOMPAT 00045
CLCOMPAT 00046
CLCOMPAT 00047
CLCOMPAT 00048
CLCOMPAT 00049
CLCOMPAT 00050
CLCOMPAT 00051
CLCOMPAT 00052
CLCOMPAT 00053
CLCOMPAT 00054
CLCOMPAT 00055
CLCOMPAT 00056
CLCOMPAT 00057
CLCOMPAT 00058
CLCOMPAT 00059
CLCOMPAT 00060

Status	CP/67 Command	VM/370 Equivalent
R	ACNT	ACNT All
A	ATTACH cuu TO {SYSTEM} AS {volid} {userid} {vaddr}	
A	BEGIN	
A	BEGIN hexloc	
A	DCP hexloc1[-hexloc2]	
A	DCP Lhexloc1[-hexloc2]	
A	DCP Thexloc1[-hexloc2]	
A	DETACH vaddr	
R	DETACH raddr	DETACH raddr FROM {SYSTEM} {userid}
A	DIAL system	

CLCOMPAT 00064 Figure 41. VM/370 Compatibility with CP-67 (Part 1 of 5)

Status	CP/67 Command	VM/370 Equivalent
N	DIRECT LOCK	
N	DIRECT UNLOCK	
A	DISABLE line	
A	DISABLE ALL	
A	DISCONN	
R	DISCONN xxx	DISCONN HOLD
A	DISPLAY hexloc1 [-hexloc2]	
R	DISPLAY L[-hexloc2]	DISPLAY IO[-hexloc2]
A	DISPLAY Lhexloc1[-hexloc2]	[-END]
R	DISPLAY T[-hexloc2]	DISPLAY TO[-hexloc2]
A	DISPLAY Thexloc1[-hexloc2]	[-END]
R	DISPLAY K[-hexloc2]	DISPLAY KO[-hexloc2]
A	DISPLAY Khexloc1[-hexloc2]	[-END]
A	DISPLAY G[-reg2]	
A	DISPLAY Greg[-reg2]	
A	DISPLAY Y[-reg2]	
A	DISPLAY Yreg[-reg2]	
A	DISPLAY X[-reg2]	
A	DISPLAY Xreg[-reg2]	
A	DISPLAY PSW	
A	DMCP hexloc1[-hexloc2]	
R	DMCP L[-hexloc2]	DMCP IO[-hexloc2]
A	DMCP Lhexloc1[-hexloc2]	[-END]
R	DMCP T[-hexloc2]	DMCP TO[-hexloc2]
A	DMCP Thexloc1[-hexloc2]	[-END]
A	DRAIN	
N	D_U_M_P	

Figure 41. Compatibiltiy of VM/370 with CP-67 (Part 2 of 5)

Status	CP/67 Command	VM/370 Equivalent
A	ENABLE line	
A	ENABLE ALL	
A	EXTERNAL	
A	IPL CMS	
A	IPL devadd	
R	IPLSAVE CCW	IPL vaddr NOCLEAR
R	KILL userid	FORCE userid
N	KILL	
A	LINK userid xxx yyy [W|R] [PASS=pwd]	
R	LINK userid xxx yyy [W|R] (NOPASS)	LINK tserid xxx yyy [W|R]
A	LOCK userid fpage lpage	
A	LOGIN userid	
R	LOGIN userid xxx	LOGIN userid MASK
A	LOGOUT	
R	LOGOUT xxx	LOGOUT HOLD
A	MSG userid line	
R	MSG CP line	MSG OPERATOR line
A	MSG ALL line	
R	PSWRESTART	SYSTEM RESTART
A	PURGE READER	
A	PURGE PRINTER	
A	PURGE PUNCH	
R	QUERY DEVICE ALL	QUERY ALL
R	QUERY DEVICE xxx	QUERY xxx
A	QUERY DUMP	
A	QUERY FILES	

Figure 41. Compatibility of VM/370 with CP-67 (Part 3 of 5)

Status	CP/67 Command	VM/370 Equivalent
A	QUERY LOGMSG	
N	QUERY MAX	
A	QUERY NAMES	
n	QUERY PORTS	
R	QUERY PORTS ALL	QUERY LINES
N	QUERY PORTS FREE	
R	QUERY PORTS xxx	QUERY xxx
A	QUERY PRIORITY userid	
R	QUERY Q2	QUERY PAGING
A	QUERY TIME	
A	QUERY userid	
A	QUERY USERS	
A	QUERY VIRTUAL xxx	
A	QUERY VIRTUAL CORE	QUERY VIRTUAL STORAGE
A	QUERY VIRTUAL ALL	
A	READY xxx	
A	REPEAT xxx y	
R	RESET	SYSTEM RESET
R	SET ADSTOP xxxxxx	ALSTOP xxxxxx
R	SET ADSTOP OFF	ALSTOP OFF
R	SET APLBALL {ON }	TERMINAL APL {ON }
	{OFF}	{OFF}
R	SET ATTN {ON }	TERMINAL ATTN {ON }
	{OFF}	{OFF}
R	SET CARDSAVE {ON }	SECCL READER { HOLD }
	{OFF}	{NOHOLD}
A	SET DUMP xxx	
A	SET {LINEDIT} {ON }	
	{RUN } {OFF}	
A	SET LOGMSG	
A	SET LOGMSG NULL	
A	SET LOGMSG n	

Figure 41. Compatibility of VM/370 with CP-67 (Part 4 of 5)

Status	CP/67 Command	VM/370 Equivalent
N	SET MAX	
A	SET MSG {ON }	
	{OFF}	
R	SET Q2 nn	SET PAGING nn
R	SET TRACE devtype	TRACE type dev
R	SET TRACE OFF	TRACE type OFF
R	SET TRACE END	TRACE END
A	SET WNG {ON }.	
	{OFF}	
A	SHUTDOWN	
A	SLEEP	
A	SPACE xxx	
R	SPOOL xxx ON yyy	SECCL yyy CLASS x
A	SPOOL xxx CONT	
R	SPOOL xxx OFF	SECCL xxx NOCONT
A	START xxx	
A	STCP hexloc	
A	STCP Shexloc	
A	STORE Lhexloc hexinfo...	
A	STORE Shexloc hexstring	
A	STORE Greg hexinfo...	
A	STORE Yreg hexinfo...	
A	STORE Xreg hexinfo...	
A	STORE PSW [hexinfo1] hexinfo2	
R	TERM xxx	FLUSH xxx
A	UNLOCK userid fpage lpage	
A	WNG userid text	
A	WNG ALL text	
R	XFER xxx {TO userid}	SECCL xxx {TO userid}
	{OFF }	{OFF }

Figure 41. Compatibility of VM/370 with CP-67 (Part 5 of 5)

CLCOMPAT 00293 **Incompatibility Statement to CP-67/CMS Users**

CLCOMPAT 00294 Although the CMS in VM/370 is built upon CMS Version 3.1 in
CLCOMPAT 00295 CP-67/CMS, there are five types of modifications that were
CLCOMPAT 00297 made to 3.1 that affect the relationship between versions:

CLCOMPAT 00301 1. Unchanged: some commands and system functions remain
CLCOMPAT 00302 unchanged; therefore, complete compatibility exists.

CLCOMPAT 00306 2. Additional Functional Capability: functional and
CLCOMPAT 00306 syntactical enhancements have been effected; but, in
CLCOMPAT 00307 some cases, old keywords and functions will be
CLCOMPAT 00308 supported.

CLCOMPAT 00311 3. Command Name Alterations: where commands have
CLCOMPAT 00312 undergone name changes, a SYNONYM file may be included
CLCOMPAT 00313 during nucleus system generation.

CLCOMPAT 00316 4. Keyword Changes: some keywords within a command have
CLCOMPAT 00317 been modified, deleted or added.

CLCOMPAT 00320 5. Major Modifications: improvements to commands and
CLCOMPAT 00320 system functions have caused complete incompatibilities
CLCOMPAT 00321 in the following areas:

CLCOMPAT 00325 - Because the CMS nucleus is significantly larger,
CLCOMPAT 00326 all MODULES must be recreated from their object
CLCOMPAT 00327 (TEXT) files using the GENMOD command.

CLCOMPAT 00329 - Because the user may now have up to 10 disks, the
CLCOMPAT 00330 logical directory identifications (filemode
CLCOMPAT 00331 letters) have been changed to reflect a more
CLCOMPAT 00332 "natural", easy-to-remember, search order: P, T,
CLCOMPAT 00332 A, B, S, C becomes A, B, C, D, E, F, G, S, Y, Z -
CLCOMPAT 00333 with the system disk being the S disk, and the
CLCOMPAT 00334 primary disk becoming the A disk.

CLCOMPAT 00336 - The following global changes of filetypes must be
CLCOMPAT 00337 made:

CLCOMPAT 00339 SYSIN to ASSEMBLE
CLCOMPAT 00341 ASP360 to MACRO

CLCOMPAT 00343 - For language processors, the DECK/NODECK has a new
CLCOMPAT 00345 meaning, that is, route the object (TEXT) file to
CLCOMPAT 00346 the spooled card punch; the LOAD/NOLOAD option now
CLCOMPAT 00347 invokes the function formerly performed by
CLCOMPAT 00348 DECK/NODECK, that is, the writing of the TEXT file
CLCOMPAT 00349 to a CMS disk.

CLCOMPAT 00351 - No 2311 disk support is provided for CP or CMS
CLCOMPAT 00352 files.

- In Version 1.0 the tape designations are as follows:

      TAP1 is for 181
      TAP2 is for 182

  The default for tape commands is TAP1.

- CMS does not function on a real CPU without the Control Program.

- TXTLIB files must be recreated.

- Because many fields have been changed in the CMS nucleus or rearranged, many user programs which refer to these fields have to be reassembled with the new CMS macro libraries.

- Modules which refer to fields containing sizes, limits, and quantities within the CMS nucleus might have to be reassembled and then regenerated.

- SYSLIB MACLIB has been renamed to CMSLIB MACLIB.

- All EXEC files should be checked for command name and operand changes, filemode usage, etc., and changed to conform to VM/370 CMS. Major changes are:

      &TYPEOUT to &CONTROL
      &PRINT   to &TYPE
      &INDEX0 to &RETCODE

- Options specified for a LOAD command do not remain in effect for subsequent INCLUDE commands; options are reset to default settings unless the SAME option is specified.

- Filenames and filetypes must be composed entirely of alphameric characters.

- If the CMS V1.0 system does not recognize a command name, the command line is automatically passed to CP. If the CMS SET and QUERY commands do not recognize a parameter, the command line is passed to CP. This is not true for EXEC files; the CP command must be explicitly stated. The feature may be negated by entering the command:

      SET IMPCP OFF

CLCMSPAT 00004 Comments of each CP-67/CMS command for VM/370 CMS:

CLCMSPAT 00015 **ALTER** command name changed to RENAME.
CLCMSPAT 00016 Components of the new file identifier may not be
CLCMSPAT 00017 specified as asterisk(*). An equal sign(=)
CLCMSPAT 00018 performs same function.
CLCMSPAT 00019 NOUP option keyword changed to NOUPDIRT; NOUP
CLCMSPAT 00020 the abbreviation.
CLCMSPAT 00022 Default options added: NOTYPE, UPDIRT.

CLCMSPAT 00024 **ASSEMBLE** only one file may be assembled per ASSEMBLE
CLCMSPAT 00025 command.

CLCMSPAT 00027 Options Changed:
CLCMSPAT 00030 NODECK is the default
CLCMSPAT 00032 DIAG|NODIAG changed to TERM|NOTERM
CLCMSPAT 00034 LTAPn not supported
CLCMSPAT 00036 LDISK option name changed to DISK

CLCMSPAT 00039 Options Added:
CLCMSPAT 00041 LOAD|NOLOAD, ALGN|NOALGN
CLCMSPAT 00044 OS|DOS, TEST|NOTEST, LINECT nn|55,
CLCMSPAT 00046 NUM|NONUM, STMT|NOSTMT

CLCMSPAT 00052 **BLIP** functionally supported by SET BLIP.

CLCMSPAT 00055 **BRUIN** not implemented.

CLCMSPAT 00058 **CEDIT** functionally supported by EDIT.

CLCMSPAT 00060 **CHARDEF** functionally supported by CP TERMINAL CHARDEF
CLCMSPAT 00061 command.

CLCMSPAT 00063 **CLOSIO** functionally supported by the CP commands, SPOOL
CLCMSPAT 00064 and CLOSE.

CLCMSPAT 00067 **CLROVER** functionally supported by SVCTRACE OFF command.

CLCMSPAT 00069 **CNVT26** functionally supported by COPYFILE command with
CLCMSPAT 00070 EBCDIC option.

CLCMSPAT 00073 **COMBINE** functionally supported by COPYFILE.

CLCMSPAT 00076 CLCMSPAT 00077 CLCMSPAT 00078	COMPARE	filemode required. NOSEQ option functionally supported by Col option.
CLCMSPAT 00081 CLCMSPAT 00083	CPFUNCTN	command name changed to CP. NOMSG option not supported.
CLCMSPAT 00086	CYTFY	functionally supported by COPYFILE.
CLCMSPAT 00089	DEBUG	no change.
CLCMSPAT 00092	DEBUG SUBCOMMANDS	
CLCMSPAT 00095	BREAK	no change.
CLCMSPAT 00097	CAW	no change.
CLCMSPAT 00099	CSW	no change.
CLCMSPAT 00101 CLCMSPAT 00103	DEF	subcommand name changed tc DEFINE; DEF minimum truncation; no change in format.
CLCMSPAT 00106	DUMP	no change.
CLCMSPAT 00109	GO	no change.
CLCMSPAT 00112	GPR	no change.
CLCMSPAT 00115	IPL	no supported from DEBUG.
CLCMSPAT 00118	KX	supported by CMS command HX.
CLCMSPAT 00121	ORIGIN	no change.
CLCMSPAT 00124	PSW	no change.
CLCMSPAT 00127	RESTART	not supported.
CLCMSPAT 00130	RETURN	no change.
CLCMSPAT 00132	SET	no change.
CLCMSPAT 00134	STORE	no change.
CLCMSPAT 00136	TIN	no change.
CLCMSPAT 00138 CLCMSPAT 00139	X	If symbol specified, length of field defaults tc 4.

```
CLCMSPAT 00142   DISK          no change.

CLCMSPAT 00144   DUMPD         function not supported.

CLCMSPAT 00146   DUMPF         functionally supported by TYPE  command with HEX
CLCMSPAT 00147                 option.

CLCMSPAT 00150   DUMPREST      functionally supported by DDR command.

CLCMSPAT 00153   ECHO          functionally supported by CP command ECHO.

CLCMSPAT 00156   EDIT          filename must be specified.
CLCMSPAT 00158                 filemode may be specified.
CLCMSPAT 00160                 LRECL may be specified for a new file.

CLCMSPAT 00163   EDIT SUBCOMMANDS

CLCMSPAT 00165   BACKSPACE     functionally  supported  by  CMS  command,  SET
CLCMSPAT 00166                 INPUT.

CLCMSPAT 00169   BLANK         functionally supported by OVERLAY.

CLCMSPAT 00171   BOTTOM        no change.

CLCMSPAT 00174   BRIEF         function accomplished by VERIFY OFF request.

CLCMSPAT 00177   CHANGE        no change.

CLCMSPAT 00180   DELETE        /string/ is not valid as an operand.
CLCMSPAT 00182                 * may be used to delete to end of file.

CLCMSPAT 00185   FILE          filetype and filemode may be specified.

CLCMSPAT 00187   FIND          no change.

CLCMSPAT 00189   INPUT         no change.

CLCMSPAT 00191   INSERT        functionally supported by INPUT request.

CLCMSPAT 00193   LOCATE        no change.

CLCMSPAT 00195   NEXT          no change.

CLCMSPAT 00197   OVERLAY       no change.

CLCMSPAT 00200   PRINT         request name changed to TYPE.
CLCMSPAT 00203                 *  may be  specified to indicate that  typing is
CLCMSPAT 00204                    to continue until EOF.
```

```
CLCMSPAT 00206                instead of L in second field, * is specified.

CLCMSPAT 00210    QUIT        no change.

CLCMSPAT 00213    REPEAT      valid only for subsequent OVERLAY request.
CLCMSPAT 00216                *  may be specified as first operand to indicate
CLCMSPAT 00217                   that the  OVERLAY request  is to  be repeated
CLCMSPAT 00218                   for the remainder of the file.

CLCMSPAT 00221    RETYPE      request name changed to REPLACE.

CLCMSPAT 00223    SAVE        filetype and filemode may be specified.

CLCMSPAT 00226    SERIAL      first operand changed to
CLCMSPAT 00228                (OFF|'seq'(same as ID)|ON|ALL).

CLCMSPAT 00231    TABDEF      functionally supported by CMS command SET INPUT.

CLCMSPAT 00234    TABSET      no change.

CLCMSPAT 00236    TOP         no change.

CLCMSPAT 00238    UP          no change.

CLCMSPAT 00240    VERIFY      operand format changed; function added.

CLCMSPAT 00242    X           no change.

CLCMSPAT 00244    Y           no change.

CLCMSPAT 00246    ZONE        no change.

CLCMSPAT 00249    ERASE       default option added; old format accepted.
CLCMSPAT 00251                * * * not supported.

CLCMSPAT 00254    EXEC        no change.

CLCMSPAT 00256    EXEC Control Words

CLCMSPAT 00258    &ERROR      action does not default to &CONTINUE.

CLCMSPAT 00260    &IF         no change.

CLCMSPAT 00262    &EXIT       no change.

CLCMSPAT 00264    &QUIT       functionally supported by &EXIT 0.

CLCMSPAT 00266    &SKIP       no change.

CLCMSPAT 00269    &GOTO       EXIT not supported as operand.
CLCMSPAT 00271                line-number now a valid operand.
```

CLCMSPAT 00274 &LOOP no change.

CLCMSPAT 00276 &CONTINUE no change.

CLCMSPAT 00279 &TYPEOUT functionally supported by &CONTROL control word;
CLCMSPAT 00280 ON, NOEXEC, RESUME, KILL not valid.
CLCMSPAT 00282 CMS operand added.

CLCMSPAT 00285
CLCMSPAT 00287
CLCMSPAT 00289 &TIME operands changed to [ON|OFF] [RESET|TYPE].
CLCMSPAT 00291

CLCMSPAT 00293 &SPACE no change.

CLCMSPAT 00295 &PRINT functionally supported by &TYPE.

CLCMSPAT 00297 &UPRINT functionally supported by &BEGTYPE.

CLCMSPAT 00299 &PUNCH no change.

CLCMSPAT 00301 &UPUNCH functionally supported by &BEGPUNCH.

CLCMSPAT 00303 &COMMENT functionally supported by * card.

CLCMSPAT 00305 &ARGS no change.

CLCMSPAT 00307 &READ VARS operand added.

CLCMSPAT 00309 &STACK no change.

CLCMSPAT 00311 &BEGSTACK ALL operand added.

CLCMSPAT 00313 &ENDSTACK functionally supported by &END.

CLCMSPAT 00315 &SET not implemented.

CLCMSPAT 00319 FILEDEF device names changed: CON-TERMINAL
CLCMSPAT 00321 DSK-DISK
CLCMSPAT 00323 DSK-nn-not supported
CLCMSPAT 00325 DUMMY-no change.
CLCMSPAT 00327 PRT-PRINTER
CLCMSPAT 00329 PUN-PUNCH
CLCMSPAT 00331 RDR-READER
CLCMSPAT 00333 TAPEn-unchanged
CLCMSPAT 00335 BAT-not supported

CLCMSPAT 00340 FINIS not supported from terminal.

CLCMSPAT 00343 FORMAT all functions supported; all operands changed.

CLCMSPAT 00346 FORTRAN This command is now supported by the commands

```
CLCMSPAT 00346                 FORTG1, FORTHX, GOFORT, and CONVERT which invoke
CLCMSPAT 00347                 IBM Program Products.

CLCMSPAT 00349  GENMOD         incompatible; positional parameters and options
CLCMSPAT 00351                 changed; equivalent function performed.

CLCMSPAT 00354  GLOBAL         PRINT function removed; others compatible.

CLCMSPAT 00356  IPL            not explicitly supported in CMS; however, the
CLCMSPAT 00357                 command may be issued and will be passed to the
CLCMSPAT 00358                 Control Program for processing.
CLCMSPAT 00359                 either a device address or system name must be
CLCMSPAT 00360                 specified.
CLCMSPAT 00362                 cyl-no may be specified.
CLCMSPAT 00364                 Options added: CLEAR|NOCLEAR.

CLCMSPAT 00366  KE             functionally supported by the CP command
CLCMSPAT 00367                 TERMINAL LINESIZE nn.

CLCMSPAT 00370  KO             command name changed to HO.

CLCMSPAT 00372  KT             command name changed to HT.

CLCMSPAT 00374  KX             command name changed to HX.

CLCMSPAT 00376  LINEND         functionally supported by the CP command
CLCMSPAT 00377                 TERMINAL LINEND.

CLCMSPAT 00379  LISTF          command name changed to LISTFILE; LISTF
CLCMSPAT 00380                 accepted.
CLCMSPAT 00385                 Option Changes:
CLCMSPAT 00387                    SORT-   option not implemented.
CLCMSPAT 00388                    ITEM-   option not implemented; ALLOC option
CLCMSPAT 00389                            produces both logical records and
CLCMSPAT 00390                            blocks.
CLCMSPAT 00392                    NAME-   option name changed to FNAME.
CLCMSPAT 00394                    TYPE-   option name changed to FTYPE.
CLCMSPAT 00396                    MODE-   option name changed to FMODE.
CLCMSPAT 00398                    REC-    supported by ALLOC option.
CLCMSPAT 00400                    DATE-   produces mm/dd/yy hh:mm.
CLCMSPAT 00401                    YEAR-   not implemented; functionally
CLCMSPAT 00402                            supported by DATE option.
CLCMSPAT 00403                    TIME-   not implemented; functionally
CLCMSPAT 00404                            supported by DATE option.
CLCMSPAT 00406                    LABEL and FORMAT options added.
CLCMSPAT 00408                    APPEND option added.
CLCMSPAT 00410                    HEADER|NOHEADER option added.


CLCMSPAT 00415  LOAD           Option Changes:
CLCMSPAT 00419                    SLCxxxxxx-option changed to ORIGIN xxxxxx.
CLCMSPAT 00420                    SLC12000-default changed to first available
CLCMSPAT 00421                       location.
CLCMSPAT 00423                    SINV-option name changed to NOINV.
```

```
CLCMSPAT 00425                  PINV-option name changed to INV.
CLCMSPAT 00427                  SREP-option name changed to NOREP.
CLCMSPAT 00429                  PREP-option name changed to REP.
CLCMSPAT 00431                  SLIBE-option name changed to NOLIBE.
CLCMSPAT 00433                  SAUTO-option name changed to NOAUTO.
CLCMSPAT 00435                  XEQ-option name changed to START.
CLCMSPAT 00437                  NOXEQ-option not supported.
CLCMSPAT 00439              Options added: RESET.

CLCMSPAT 00441              TXTLIB files may no longer be specified in a
CLCMSPAT 00442              LOAD command, but must have been previously
CLCMSPAT 00443              specified by a GLOBAL command.


CLCMSPAT 00447  LOADMOD     filetype must be specified if filemode is given.


CLCMSPAT 00450  LOGIN       functionally supported by ACCESS command.
CLCMSPAT 00451              comma between mode and extdisk replaced with
CLCMSPAT 00453              slash (/); extdisk optional.
CLCMSPAT 00455              Options:
CLCMSPAT 00456                  NOPROF-option supported.
CLCMSPAT 00458                  NOTYPE-option not supported.
CLCMSPAT 00460                  NO-UFD-option name changed to ERASE.


CLCMSPAT 00463  LOGOUT      functionally supported by CP command LOGOUT.


CLCMSPAT 00465  MAPPRT      functionally supported by CMS commands TYPE or
CLCMSPAT 00466              PRINT.


CLCMSPAT 00469  MODMAP      no change.


CLCMSPAT 00472  OFFLINE READ     functionally supported by READCARD command.

CLCMSPAT 00474  OFFLINE PRINT    functionally supported by PRINT command.

CLCMSPAT 00476  OFFLINE PRINTCC  functionally supported by PRINT command.

CLCMSPAT 00478  OFFLINE PRINTUPC functionally supported by PRINT command.

CLCMSPAT 00480  OFFLINE PUNCH    functionally supported by PUNCH command.

CLCMSPAT 00482  OFFLINE PUNCHCC  functionally supported by PUNCH command.

CLCMSPAT 00484  OFFLINE PUNCHDT  functionally supported by PUNCH command.


CLCMSPAT 00487  OSTAPE      command name changed to TAPPDS.
```

CLCMSPAT 00489 **PLI** This command is now supported by the command
CLCMSPAT 00491 PLIOPT which invokes an IBM Program Product.

CLCMSPAT 00494 **PRINTF** command name changed to TYPE.
CLCMSPAT 00496 filemode may be specified.
CLCMSPAT 00498 n3 functionally supported by COL option.
CLCMSPAT 00500 Options added: HEX, MEMBER.

CLCMSPAT 00503 **RELEASE** DET option not implemented although functionally
CLCMSPAT 00504 supported by the CP command DETACH.

CLCMSPAT 00507 **REUSE** command name changed to INCLUDE.
CLCMSPAT 00509 option differences are the same as for LOAD.
CLCMSPAT 00510 TXTLIB files may no longer be specified in the
CLCMSPAT 00511 command; but must have been previously specified
CLCMSPAT 00512 by a GLOBAL command.

CLCMSPAT 00515 **RT** no change.

CLCMSPAT 00518 **SCRIPT** not implemented.

CLCMSPAT 00520 **SETERR** functionally supported by SVCTRACE ON.

CLCMSPAT 00522 **SETOVER** functionally supported by SVCTRACE ON.

CLCMSPAT 00524 **SNOBOL** not implemented.

CLCMSPAT 00527 **SORT** filemode must be specified for both input and
CLCMSPAT 00528 output files.

CLCMSPAT 00531 **SPLIT** functionally supported by COPYFILE command.

CLCMSPAT 00533 **START** (NO) operand not valid.

CLCMSPAT 00535 **STAT** functionally supported by QUERY command.

CLCMSPAT 00537 **STATE** no change.

CLCMSPAT 00539 **SYN** command name changed to SYNONYM; SYN minimum
CLCMSPAT 00540 truncation.
CLCMSPAT 00543 filetype must be SYNONYM.

```
CLCMSPAT 00545                  Options:
CLCMSPAT 00547                     SYNONYM command with P and PLSER options is
CLCMSPAT 00549                     functionally supported by QUERY SYNONYM.


CLCMSPAT 00553   TAPE           TAPn may also be specified by a virtual address.
CLCMSPAT 00554                  "n" options after SCAN, SKIP, SLOAD, replaced by
CLCMSPAT 00555                  'EOF n'.
CLCMSPAT 00559                  Options added: 7TRACK|9TRACK, DEN, TRTCH.
CLCMSPAT 00562                  Functions added: MODESET, BSF, BSR, ERG, FSF,
CLCMSPAT 00563                     FSR, RUN, REW.

CLCMSPAT 00567   TAPE DUMP      Options added: WTM|NOWTM, NOPRINT|PRINT|TERM|DISK.

CLCMSPAT 00572   TAPE LOAD      functionally supported by TAPE LOAD ECFn.
CLCMSPAT 00574                  file identifiers may be specified.
CLCMSPAT 00577                  Options added: NOPRINT|PRINT|TERM|DISK,
CLCMSPAT 00579                     EOFn|EOT|EOF1.

CLCMSPAT 00583   TAPE SCAN      filename and filetype may be specified.
CLCMSPAT 00585                  functionally supported by TAPE SCAN (EOF n).
CLCMSPAT 00587                  Options added: NOPRINT|PRINT|TERM|DISK,
CLCMSPAT 00589                     EOFn|EOT|EOF1.

CLCMSPAT 00592   TAPE SKIP      comments same as for TAPE SCAN.

CLCMSPAT 00595   TAPE SLOAD     functionally supported by TAPE LOAD fn ft.
CLCMSPAT 00597                  filemode may be specified.

CLCMSPAT 00600   TAPEIO         functionally supported by TAPE.

CLCMSPAT 00603   TAPPDS         default filename is TAPPDS for NOPDS option.
CLCMSPAT 00605                  default filetype is CMSUT1.
CLCMSPAT 00607                  Option Changes:
CLCMSPAT 00609                     NPDS - option name changed to NOPDS.
CLCMSPAT 00611                     NCOLI - option name changed to NOCOLI.
CLCMSPAT 00613                     TAPx - default is TAP1.
CLCMSPAT 00615                     NEND - option name changed to NOEND.
CLCMSPAT 00617                     NMAXTEN - option name changed to NOMAXTEN.

CLCMSPAT 00620   TAPRINT        functionally supported by MOVEFILE command.


CLCMSPAT 00624   TPCOPY         functionally supported by MOVEFILE command.


CLCMSPAT 00627   TXTLIB         PRINT and LIST functions supported by MAP
CLCMSPAT 00628                  function.


CLCMSPAT 00632   UPDATE         Option Changes:
CLCMSPAT 00634                     P - option name changed to REP.
CLCMSPAT 00636                     default options added NOREP, NOSEQ8, NOINC.
```

CLCMSPAT 00639 **USP** functionally supported by INCLUDE command with
CLCMSPAT 00640 the SAME option. See discussion of REUSE
CLCMSPAT 00641 compatibility.

CLCMSPAT 00644 **VSET** command name changed to SET.
CLCMSPAT 00650 Functions:
CLCMSPAT 00652 BLIP ON may be specified to return to
CLCMSPAT 00653 default
CLCMSPAT 00655 CHARDEF functionally supported by CP
CLCMSPAT 00657 command TERMINAL|(CHARDEL|LINEDEL|ESCAPE)
CLCMSPAT 00661 IMPEX no change.
CLCMSPAT 00663 LDRTBLS no change.
CLCMSPAT 00664 LINEND functionally supported by CP
CLCMSPAT 00666 command TERMINAL LINEND.
CLCMSPAT 00668 RDYMSG ON|OFF changed to LMSG|SMSG.
CLCMSPAT 00670 REDTYPE no change.
CLCMSPAT 00672 RELPAGE no change.
CLCMSPAT 00673 Functions added: INPUT, OUTPUT, INOUT, ABBREV,
CLCMSPAT 00674 IMPCP.

CLCMSPAT 00678 **WRTAPE** functionally supported by TAPE command or
CLCMSPAT 00680 MOVEFILE command.

CLCMSPAT 00683 **$** command name changed to RUN.
CLCMSPAT 00685 filetype and filemode may be specified.

CLCMSPAT 00686 files may also have filetypes in addition to
CLCMSPAT 00687 EXEC, MODULE, and TEXT of those used by the
CLCMSPAT 00688 language processors for input.

CLPROGPR 00007 **APPENDIX G: PROGRAM PRODUCTS INFORMATION**

CLPROGPR 00009 Certain functions referred to in this publication are
CLPROGPR 00010 provided through IBM Program Products, which are available
CLPROGPR 00011 from IBM for a license fee. Program Products referred to in
CLPROGPR 00012 this manual are:

CLPROGPR 00017 • Full American National Standard COBOL Version 4. See the
CLPROGPR 00018 Publication: IBM Virtual Machine Facility/370, COBOL
CLPROGPR 00020 Programmer's Guide (Order No. GC20-18xx-x).

CLPROGPR 00023 • PL/I Optimizing Compiler. See the publication: IBM
CLPROGPR 00024 Virtual Machine Facility/370 PL/I Programmer's Guide
CLPROGPR 00025 (Order No. GC20-18xx-x).

CLPROGPR 00028 • FORTRAN (G1). A version of the FORTRAN (G) compiler
CLPROGPR 00028 suitable for the terminal environment. See the
CLPROGPR 00030 publication: IBM Virtual Machine Facility/370
CLPROGPR 00031 FORTRAN Programmer's Guide (Order No. GC20-18xx-xxx-x).

CLPROGPR 00034 • FORTRAN (HX). A version of the FORTRAN (G) compiler
CLPROGPR 00034 suitable for the terminal environment. See the
CLPROGPR 00036 publication: IBM Virtual Machine Facility/370 3_7_0
CLPROGPR 00037 FORTRAN Programmer's Guide (Order No. Go. GC20-18xx-x).

CLPROGPR 00040 • Code and GO FORTRAN. A FORTRAN compiler designed for
CLPROGPR 00041 very fast compile-execute sequence. See the Publication:
CLPROGPR 00043 IBM Virtual Machine Facility/370 FORTRAN Programmer's
CLPROGPR 00044 Guide (Order No. GC20-18xx-x).

CLMINTRU 00006 APPENDIX H: MINIMUM TRUNCATIONS FOR CMS COMMAND NAMES

CLMINTRU 00008 SYSTEM SHORTEST
CLMINTRU 00010 COMMAND FORM____

CLMINTRU 00012	ASSEMBLE	A
CLMINTRU 00013	ACCESS	AC
CLMINTRU 00014	BASIC	BA
CLMINTRU 00015	COMPARE	COM
CLMINTRU 00016	COPYFILE	COPY
CLMINTRU 00017	EDIT	E
CLMINTRU 00018	EXEC	EX
CLMINTRU 00019	FILEDEF	FI
CLMINTRU 00020	GENMOD	G
CLMINTRU 00021	GLOBAL	GL
CLMINTRU 00022	INCLUDE	IN
CLMINTRU 00023	LISTFILE	L
CLMINTRU 00024	LOADMOD	LOADM
CLMINTRU 00025	MACLIB	MAC
CLMINTRU 00026	MODMAP	MOD
CLMINTRU 00027	MOVEFILE	MOVE
CLMINTRU 00028	PRINT	PR
CLMINTRU 00029	PUNCH	PU
CLMINTRU 00030	QUERY	Q
CLMINTRU 00031	RENAME	R
CLMINTRU 00032	RELEASE	REL
CLMINTRU 00033	READCARD	READ
CLMINTRU 00034	SET	S
CLMINTRU 00035	SVCTRACE	SVC
CLMINTRU 00036	SYNONYM	SYN
CLMINTRU 00037	TYPE	T
CLMINTRU 00038	TXTLIB	TXT
CLMINTRU 00039	UPDATE	U

CLGLOSS 00007 GLOSSARY

CLGLOSS 00011 For a more complete list of data processing terms, see the
CLGLOSS 00013 publication Data Processing Glossary, GC20-1699. Where a
CLGLOSS 00014 term has an asterisk (*) preceding the listed meaning, it is
CLGLOSS 00015 an American National Standard definition.

CLGLOSS 00017 active disk table: A table residing in the user's copy of
CLGLOSS 00018 the CMS nucleus that contains an entry for each of the CMS
CLGLOSS 00020 disks that can be accessed.

CLGLOSS 00021 active file table: A table residing in the user's copy of
CLGLOSS 00022 the CMS nucleus that contains an entry for each user file
CLGLOSS 00024 that is open.

CLGLOSS 00026 active page: A page in real storage that can be addressed.

CLGLOSS 00028 address stop: A capability to specify at the system console
CLGLOSS 00028 an address which, when encountered, causes a halt in
CLGLOSS 00029 processing.

CLGLOSS 00031 address translation: The process of changing the address of
CLGLOSS 00031 a data item or an instruction from its virtual address to
CLGLOSS 00032 its real storage address. See also dynamic address
CLGLOSS 00033 translation.

CLGLOSS 00037 alphameric characters: In software, usually the characters
CLGLOSS 00037 A through Z, digits 0 through 9, and the special characters
CLGLOSS 00038 #, $, and @.

CLGLOSS 00041 attention interruption: An interruption of instruction
CLGLOSS 00041 execution caused by a terminal user pressing the attention
CLGLOSS 00043 key. See also simulated attention.

CLGLOSS 00044 attention key: A function key on some terminals that causes
CLGLOSS 00046 an interruption of execution by the central processing unit.

CLGLOSS 00047 available page queue: A queue of the pages whose real
CLGLOSS 00048 storage is currently available for allocation to any task.
CLGLOSS 00050 See also active page queue, hold page queue.

CLGLOSS 00051 basic control mode: A mode in which the features of an IBM
CLGLOSS 00052 System/360 computing system and additional System/370
CLGLOSS 00053 features, such as new machine instructions, are operational
CLGLOSS 00054 on a System/370 computing system. See also extended control
CLGLOSS 00055 (EC) mode.

CLGLOSS 00057 BC mode: Basic control mode.

CLGLOSS 00059 channel program translation: In a channel program,
CLGLOSS 00060 replacement by software of virtual addresses with real
CLGLOSS 00061 addresses.

CLGLOSS 00063 character-deletion character: A character within a line of
CLGLOSS 00064 terminal input specifying that it and the immediately
CLGLOSS 00065 preceding character are to be removed from the line.

CLGLOSS 00067 CMS: Conversational Monitor System.

CLGLOSS 00069 CMS function: See CMS service routines.

CLGLOSS 00070 CMS nucleus: A portion of CMS that is resident in the
CLGLOSS 00071 user's virtual storage whenever CMS is executing. Each CMS
CLGLOSS 00072 user receives a copy of the CMS nucleus when he performs an
CLGLOSS 00073 IPL of CMS.

CLGLOSS 00074 CMS primary disk: In VM/370, an area on a direct access
CLGLOSS 00075 storage device allocated to each CMS user or accessed as the
CLGLOSS 00076 A-disk on which his stored files and/or data sets are
CLGLOSS 00077 retained until he requests that they be deleted.

CLGLOSS 00078 CMS service routines: Routines that CMS commands use for
CLGLOSS 00079 internal processing, such as addressing and updating
CLGLOSS 00081 directories or performing disk or terminal input/output.

CLGLOSS 00082 CMS system disk: In VM/370, an area on direct access
CLGLOSS 00083 storage containing the CMS nucleus, and the disk-resident
CLGLOSS 00084 portion of CMS. The CMS system disk can be accessed as the
CLGLOSS 00086 S-disk in read-only mode.

CLGLOSS 00087 CMS system file: In VM/370, any CMS file residing on the
CLGLOSS 00089 CMS system disk as opposed to the user's disks.

CLGLOSS 00090 CMS user disk: In VM/370, CMS disk area allocated to each
CLGLOSS 00092 user (at the time he is authorized to use the VM/370 system)
CLGLOSS 00092 on which stored files are retained until the user causes
CLGLOSS 00094 them to be deleted.

CLGLOSS 00095 cold start: (1) Usually synonymous with initial program
CLGLOSS 00096 loader. (2) In VM/370, a system restart that ignores
CLGLOSS 00098 previously initialized data areas.

CLGLOSS 00099 command: (1) * A control signal. (2) * Loosely, an
CLGLOSS 00100 instruction in machine language. (3) * Loosely, a
CLGLOSS 00101 mathematical or logic operator. (4) A request from a
CLGLOSS 00102 terminal for the execution of a particular program, called a
CLGLOSS 00103 command processor. (5) See also operator command, channel
CLGLOSS 00104 command, subcommand.

CLGLOSS 00105 command privilege class: In VM/370, classes assigned to a
CLGLOSS 00106 VM/370 user that allow him to access logical subsets of the
CLGLOSS 00107 VM/370 control program commands.

CLGLOSS 00108 command processor: A problem program executed as a result
CLGLOSS 00110 of entering a command at the terminal.

CLGLOSS 00112 command scan: A service routine that checks the syntax of
CLGLOSS 00113 commands and subcommands.

CLGLOSS 00115 context editing: In systems with time sharing, a method of
CLGLOSS 00116 editing a line data set or file without using line numbers.
CLGLOSS 00116 To refer to a particular line, all or part of the contents
CLGLOSS 00118 of that line are specified.

CLGLOSS 00120 control terminal: In systems with time sharing, any active
CLGLOSS 00121 terminal at which the user is authorized to enter commands
CLGLOSS 00122 affecting system operation.

CLGLOSS 00124 Conversational Monitor System (CMS): A virtual machine
CLGLOSS 00125 operating system that operates only under the control of the
CLGLOSS 00126 VM/370 control program.

CLGLOSS 00128 core table: See page table.

CLGLOSS 00130 current line pointer: (1) In systems with time sharing, a

pointer that indicates the line of a line data set or file with which the user is currently working. (2) In VM/370, a pointer that indicates the record number of a CMS file with which the user is currently working.

DAT: Dynamic address translation.

dedicated channel option: A VM/370 virtual machine option which improves performance of a virtual machine by bypassing address translation of virtual devices and VM/370 control program channel scheduling.

directory: An index that is used by the control program to locate one or more blocks of data that are stored in separate areas of a data set or files in direct access storage.

directory file: A VM/370 disk file that defines the virtual machine configuration of each user.

disconnect mode: In VM/370, the state of a virtual machine that does not have a physical line or terminal connected as an operator console.

dispatching: (1) The act of scheduling a task for execution. (2) In VM/370, the starting of virtual machine execution.

dynamic address translation (DAT): (1) The change of a virtual storage address to a real storage address during execution of an instruction. See also address translation. (2) A hardware feature that performs the translation.

EC mode: See extended control mode.

edit mode: In systems with time sharing, an entry mode under the EDIT command that accepts successive subcommands suitable for modifying an existing line data set or file.

extended control (EC) mode: A mode in which all the features of an IBM System/370 computing system, including dynamic address translation, are operational. See also basic control (BC) mode.

CLGLOSS 00174 IBM Virtual Machine Facility/370 (VM/370): A time-sharing
CLGLOSS 00174 System Control Program that consists of: (1) a control
CLGLOSS 00175 program that manages the resources of an IBM System/370
CLGLOSS 00176 computing system in such a way that multiple remote terminal
CLGLOSS 00178 users have a functional simulation of an IBM System/370 (a
CLGLOSS 00178 virtual machine) at their disposal and (2) the
CLGLOSS 00179 Conversational Monitor System.

CLGLOSS 00181 input line: In VM/370, information typed by a user between
CLGLOSS 00182 the time the typing element of his terminal comes to rest--
CLGLOSS 00182 following a carriage return--until another carriage return
CLGLOSS 00184 is typed.

CLGLOSS 00185 interaction: In systems with time sharing, a basic unit
CLGLOSS 00186 used to record system activity, consisting of acceptance of
CLGLOSS 00187 a line of terminal input, processing of the line, and a
CLGLOSS 00188 response, if any. An interaction is recorded when a user
CLGLOSS 00190 task starts its wait for a line of terminal input.

CLGLOSS 00192 interaction time: In systems with time sharing, the time
CLGLOSS 00193 between requests for successive lines of terminal input.

CLGLOSS 00195 interactive user: In VM/370, a user (a virtual machine) is
CLGLOSS 00195 dynamically classed by the control program as interactive if
CLGLOSS 00196 he executes fewer than approximately 250,000 instructions
CLGLOSS 00197 between terminal interrupts.

CLGLOSS 00198 job: (1) * A specified group of tasks prescribed as a unit
CLGLOSS 00199 of work for a computer. By extension, a job usually includes
CLGLOSS 00200 all necessary programs, linkages, files, and instructions to
CLGLOSS 00201 the operating system. (2) In IBM System/360 operating
CLGLOSS 00202 systems, a collection of related problem programs,
CLGLOSS 00203 identified in the input stream by a JOB statement followed
CLGLOSS 00204 by one or more EXEC and DD statements. (3) In systems with
CLGLOSS 00205 time sharing, the processing done on behalf of one user from
CLGLOSS 00207 logon to logoff. Synonymous with terminal job.

CLGLOSS 00208 line-deletion character: A terminal character that
CLGLOSS 00209 specifies that all characters are to be deleted from a
CLGLOSS 00211 logical line of terminal input.

CLGLOSS 00213 line number: A number associated with a line that can be
CLGLOSS 00214 used to refer to the line.

CLGLOSS 00215 line number editing: In systems with time sharing, a mode
CLGLOSS 00216 of operation under the EDIT command in which lines or
CLGLOSS 00217 records to be modified are referred to by line or record
CLGLOSS 00218 number.

CLGLOSS 00219 load map: A map containing the main/real storage locations
CLGLOSS 00220 of control sections and entry points of a program loaded
CLGLOSS 00222 into main/real storage.

CLGLOSS 00223 main storage: (1) * The general purpose storage of a
CLGLOSS 00224 computer. Usually, main storage can be accessed directly by
CLGLOSS 00225 the operating registers. Contrasts with auxiliary storage.
CLGLOSS 00226 (2) All program-addressable storage from which instructions
CLGLOSS 00227 may be executed and from which data can be loaded directly
CLGLOSS 00229 into registers. See also real storage and virtual storage.

CLGLOSS 00231 MCR: See multiple control registers.

CLGLOSS 00233 multiple control registers (MCR): A set of registers used
CLGLOSS 00234 for operating system control of relocation, priority
CLGLOSS 00235 interruption, program event recording, error recovery, and
CLGLOSS 00236 masking operations.

CLGLOSS 00238 null line: In VM/370, a terminal input line consisting of a
CLGLOSS 00238 carriage return or line-end character issued as the first
CLGLOSS 00239 and only information after the typing element of the
CLGLOSS 00240 terminal has come to rest following a previous carriage
CLGLOSS 00241 return or line-end character. Usually used to terminate
CLGLOSS 00243 input mode and enter EDIT mode.

CLGLOSS 00245 page: (1) A fixed-length block of instructions, data, or
CLGLOSS 00246 both, that can be transferred between real storage and
CLGLOSS 00247 external page storage. (2) To transfer instructions, data,
CLGLOSS 00248 or both between real storage and external page storage.

CLGLOSS 00250 page-in: The process of transferring a page from external
CLGLOSS 00251 page storage to real storage.

CLGLOSS 00253 page number: The part of a virtual storage address needed
CLGLOSS 00254 to refer to a page.

CLGLOSS 00256 page-out: The process of transferring a page from real
CLGLOSS 00257 storage to external page storage.

CLGLOSS 00259 paging: The process of transferring pages between real
CLGLOSS 00260 storage and external page storage.

CLGLOSS 00262 paging area: In VM/370, an area in a direct access storage
CLGLOSS 00263 device that is used by the control program for the temporary
CLGLOSS 00264 storage of pages when paging occurs.

CLGLOSS 00266 paging device: A direct access storage device on which
CLGLOSS 00267 pages (and possibly other data) are stored.

CLGLOSS 00269 paging supervisor: A part of the supervisor that allocates
CLGLOSS 00270 and releases real storage space (page frames) for pages, and
CLGLOSS 00272 initiates page-in and page-out operations.

CLGLOSS 00273 password: In systems with time sharing, a one- to
CLGLOSS 00274 eight-character symbol that the user may be required to
CLGLOSS 00275 supply at the time he logs on the system. The password is
CLGLOSS 00276 confidential as opposed to user identification. Users can
CLGLOSS 00278 also assign passwords to data sets or files.

CLGLOSS 00281 preferred virtual machine: In VM/370, a generic name
CLGLOSS 00282 describing a particular virtual machine to which one or more
CLGLOSS 00284 of the performance options have been assigned.

CLGLOSS 00285 priority: A virtual machine parameter that influences the
CLGLOSS 00287 internal scheduling algorithm of the VM/370 control program.

CLGLOSS 00288 prompting: A function that helps a terminal user by
CLGLOSS 00289 requesting him to supply operands necessary to continue
CLGLOSS 00290 processing.

CLGLOSS 00292 real address: The address of a location in real storage.

CLGLOSS 00294 real storage: The storage of System/370 from which the
CLGLOSS 00295 central processing unit can directly obtain instructions and
CLGLOSS 00296 data, and to which it can directly return results. See also
CLGLOSS 00297 main storage.

CLGLOSS 00300 relocate hardware: See dynamic address translation.

CLGLOSS 00302 response time: (1) The time between the submission of an
CLGLOSS 00303 item of work to a computing system and the return of
CLGLOSS 00304 results. (2) In systems with time sharing, the time between
CLGLOSS 00305 the end of a block or line-end character of terminal input
CLGLOSS 00306 and the display of the first character of system response at
CLGLOSS 00308 the terminal.

CLGLOSS 00310 segment: (1) A continuous 64K area of virtual storage.
CLGLOSS 00311 that is allocated to a job or system task. (2) In VM/370, a
CLGLOSS 00312 segment can also be 1024K.

CLGLOSS 00313 separator: (1) * See file separator, group separator,
CLGLOSS 00314 information separator, record separator, unit separator. (2)
CLGLOSS 00316 * Same as delimiter.

CLGLOSS 00317 simulated attention: A function that allows terminals
CLGLOSS 00318 without attention keys to interrupt processing. The
CLGLOSS 00319 terminal is queried periodically for a specified character
CLGLOSS 00321 string. See also attention interruption.

CLGLOSS 00322 spooling area: Any direct access storage area temporarily
CLGLOSS 00323 used by VM/370 to store input for a virtual card reader or
CLGLOSS 00325 output from a virtual printer or punch.

CLGLOSS 00327 static CP area: Those portions of virtual storage that are
CLGLOSS 00328 allocated, during system generation and IPL, to the control
CLGLOSS 00329 program functions.

CLGLOSS 00331 storage block: A 2K block of real storage to which a
CLGLOSS 00332 storage key can be assigned.

CLGLOSS 00334 System restart: The restart that allows reuse of previously
CLGLOSS 00334 initialized areas. System restart usually requires less
CLGLOSS 00336 time than IPL. Synonymous with warm start.

CLGLOSS 00337 temporary disk: In VM/370, an area on a direct access
CLGLOSS 00338 storage device allocated to the user at logon time on which
CLGLOSS 00339 his stored files and/or data sets are retained until he logs
CLGLOSS 00340 off.

CLGLOSS 00341 terminal I/O wait: The condition of a task that cannot
CLGLOSS 00342 continue processing until a message is received from a
CLGLOSS 00343 terminal.

CLGLOSS 00345 terminal job: See job (3).

CLGLOSS 00347 terminal user: In systems with time-sharing, anyone who is
CLGLOSS 00348 eligible to log on.

CLGLOSS 00350 text library: In VM/370, a CMS user or system file that
CLGLOSS 00351 contains relocatable object modules and a directory
CLGLOSS 00352 indicating the location of each of these modules within the
CLGLOSS 00353 library.

CLGLOSS 00355 time share: To use a device for two or more interleaved
CLGLOSS 00356 purposes.

CLGLOSS 00358 time sharing: (1) * Pertaining to the interleaved use of
CLGLOSS 00359 the time of a device. (2) A method of using a computing
CLGLOSS 00360 system that allows a number of users to execute programs
CLGLOSS 00361 concurrently and to interact with the programs during
CLGLOSS 00362 execution.

CLGLOSS 00364 time slice: (1) A uniform interval of time on the central
CLGLOSS 00365 processing unit allocated for use in performing a task. Once
CLGLOSS 00366 the interval has expired, CPU time is allocated to another
CLGLOSS 00367 task; thus, a task cannot monopolize CPU time beyond a
CLGLOSS 00368 fixed limit. (2) In systems with time sharing, a segment of
CLGLOSS 00369 time allocated to a terminal job. See also major time slice,
CLGLOSS 00370 minor time slice.

CLGLOSS 00372 time slicing: (1) An optional feature of the MFT and MVT
CLGLOSS 00373 control program configurations that can be used to prevent a
CLGLOSS 00374 task from monopolizing the central processing unit and,
CLGLOSS 00375 thereby, delaying the assignment of CPU time to other tasks.
CLGLOSS 00376 (2) In systems with time sharing, the allocation of time
CLGLOSS 00377 slices to terminal jobs.

CLGLOSS 00380 user disk: See CMS user disk.

CLGLOSS 00382 user identification: A one- to eight character symbol
CLGLOSS 00383 identifying each system user. Abbreviated USERID.

CLGLOSS 00384 user profile table: In systems with time sharing, a table
CLGLOSS 00385 of user attributes kept for each active user, built from
CLGLOSS 00386 information gathered during the logon.

CLGLOSS 00388 verification mode: In systems with time-sharing, a mode of
CLGLOSS 00389 operation under the EDIT command in which all subcommands
CLGLOSS 00390 are acknowledged and any textual changes are displayed as
CLGLOSS 00391 they are made.

CLGLOSS 00393 virtual address: An address that refers to virtual storage
CLGLOSS 00394 and must, therefore, be translated into a real storage
CLGLOSS 00395 address when it is used.

CLGLOSS 00397 virtual computing system: Same as virtual machine.

CLGLOSS 00399 virtual CPU time: In VM/370, the time required to execute
CLGLOSS 00400 only the instructions of the virtual machine.

CLGLOSS 00402 virtual equals real (V=R) storage: An area of virtual
CLGLOSS 00402 storage that has the same range or address as real storage
CLGLOSS 00403 and is used for a program or, in OS/VS, part of a program
CLGLOSS 00405 that cannot be paged during execution.

CLGLOSS 00407 virtual machine: A functional simulation of an IBM
CLGLOSS 00408 System/370 computing system. Each virtual machine usually is
CLGLOSS 00409 controlled by a suitable operating system. VM/370 controls
CLGLOSS 00411 the concurrent execution of multiple virtual machines.

CLGLOSS 00412 virtual = real option: In VM/370, a virtual machine option
CLGLOSS 00413 that permits the entire storage of a virtual machine to have
CLGLOSS 00414 the same range of addresses as real storage, except page
CLGLOSS 00415 zero which must be relocated. It is used for programs that
CLGLOSS 00415 dynamically modify channel programs and for performance
CLGLOSS 00416 gains.

CLGLOSS 00418 virtual storage: (1) Addressable space that appears to the
CLGLOSS 00419 user as real storage, from which instructions and data are
CLGLOSS 00420 mapped into real storage locations. The size of virtual
CLGLOSS 00421 storage is limited by the addressing scheme of the computing
CLGLOSS 00422 system and by the amount of auxiliary storage available,
CLGLOSS 00423 rather than by the actual number of real storage locations.
CLGLOSS 00424 (2) Addressable space that appears to the user as real
CLGLOSS 00424 storage, from which instructions and data are mapped into
CLGLOSS 00425 real storage locations. The size of the virtual storage is
CLGLOSS 00427 limited be the addressing scheme of the virtual machine and
CLGLOSS 00428 the aggregated amount of external page storage available.

CLGLOSS 00430 VM/370: IBM Virtual Machine Facility/370.

CLGLOSS 00431 VM/370 console function: A facility that allows the
CLGLOSS 00432 terminal user to simulate the functions available at a CPU
CLGLOSS 00433 operator's console of the real machine. The command
CLGLOSS 00434 facilities of the VM/370 control program (excluding its
CLGLOSS 00434 subsystem CMS) are referred to collectively as VM/370
CLGLOSS 00435 commands and the ones which affect simulation are referred
CLGLOSS 00436 to as console functions.

CLGLOSS 00438 warm start: (1) Same as system restart. (2) The automatic
CLGLOSS 00439 reinitialization of the VM/370 control program that occurs
CLGLOSS 00440 if the control program can not continue processing.

-- END--