

# Honeywell



**LEVEL 66**  
SOFTWARE  
**TSS GENERAL**  
**INFORMATION**

SERIES 60 (LEVEL 66)/6000

SOFTWARE

# TSS GENERAL INFORMATION MANUAL

## SUBJECT

General Description of the Time Sharing System including the Command Language, Files, Terminal Usage, and Service Subsystems

## SPECIAL INSTRUCTIONS

This revision replaces Revision 0, dated April 1974, and its addenda A and B. Change bars in the margins indicate changes and additions; asterisks indicate deletions.

## SOFTWARE SUPPORTED

Series 60 (Level 66) Software Release 3  
Series 6000 Software Release I

## ORDER NUMBER

DD22, Rev. 1

September 1976

**Honeywell**

## PREFACE

This manual provides overall information on the Honeywell Series 60 Level 66 and Series 6000 Time Sharing System. The Series 60 Level 66 is hereafter referred to as Series 60. The information in this manual refers to both Series 6000 and the Series 60, unless otherwise specifically stated. Included in this document are command definitions, time sharing file descriptions, terminal operations, error message definitions, and service/utility subsystem descriptions.

# CONTENTS

		Page
Section I	Time Sharing System . . . . .	1-1
	System Description . . . . .	1-1
Section II	Command Language and File Usage . . . . .	2-1
	Definitions . . . . .	2-1
	File Designation . . . . .	2-3
	File Description Subcatalog Level . . . . .	2-4.1
	File Names, Catalog Names, and Passwords . . . . .	2-5
	System Selection . . . . .	2-5
	Commands . . . . .	2-5
Section III	Time Sharing Error Messages Explanation . . . . .	3-1
Section IV	Terminal Usage . . . . .	4-1
	General . . . . .	4-1
	Teleprinter Operation . . . . .	4-1
	Terminal Applications . . . . .	4-1
	Editing . . . . .	4-2
	Log-On Procedure . . . . .	4-3
	Entering Build Mode Input . . . . .	4-7
	Correction or Modification of Line-Numbered Files . . . . .	4-7
	Automatic Terminal Disconnections . . . . .	4-8
	Log-Off Procedure . . . . .	4-8
	Terminating an Output Process . . . . .	4-9
	Paper Tape Input in Build Mode . . . . .	4-9
	Building File from Non-ASCII Paper Tape . . . . .	4-10
	Automatic Paper Tape Input . . . . .	4-10
	Keyboard/Display Terminal . . . . .	4-11
	General Characteristics . . . . .	4-12
	Data Display and Transmission . . . . .	4-13
	Log-On . . . . .	4-14
	Log-Off . . . . .	4-15
	Unique Features . . . . .	4-15
	7700 Series VIP Tape Cassette and Print Page Adapter Operations . . . . .	4-15
	Output to Cassette . . . . .	4-15
	Input from Cassette . . . . .	4-16
	Echo Back . . . . .	4-16
	Backspace Cassette . . . . .	4-16
	Rewind Cassette . . . . .	4-16
	Output to Printer . . . . .	4-17
	Continuous Output Mode . . . . .	4-17
	Summary of 7700 Cassette/Printer Commands . . . . .	4-17
	Commands for VIP Terminals . . . . .	4-18
	Form Feed Commands . . . . .	4-18
	Case Commands . . . . .	4-18
Section V	Service Subsystems and Programs . . . . .	5-1
	ABACUS Subsystem . . . . .	5-1
	Use of ABACUS . . . . .	5-1
	Numbers . . . . .	5-2
	Variables . . . . .	5-2

CONTENTS (cont)

	Page
Constants and Functions . . . . .	5-3
Summation Operator and FOR Variables. . . . .	5-3
Continuation Lines. . . . .	5-5
Order of Evaluation and Use of Parentheses. . . . .	5-5
Mode and Precision of Calculations. . . . .	5-5
ASCII-TO-ASCII Conversion Subsystem. . . . .	5-6
ASCASC Subsystem/Command. . . . .	5-6
Execution . . . . .	5-6
File System. . . . .	5-7
File System Structure . . . . .	5-7
Catalogs and Files. . . . .	5-7
Passwords . . . . .	5-7
Permissions . . . . .	5-8
General, Specific, and EXCLUDE Permission . . . . .	5-9
ACCESS Subsystem . . . . .	5-11
Capabilities. . . . .	5-11
Use of ACCESS . . . . .	5-12
Identifiers and Delimiters in User Responses. . . . .	5-14
Word Delimiters. . . . .	5-15
Line Delimiters. . . . .	5-15
ACCESS Functions. . . . .	5-16
Short Form Usage of ACCESS Functions . . . . .	5-17
Questions and Responses. . . . .	5-18
CREATE CATALOG . . . . .	5-18
CREATE FILE. . . . .	5-21
ACCESS FILE. . . . .	5-23
DEACCESS FILE. . . . .	5-25
PURGE CATALOG. . . . .	5-25
PURGE FILE . . . . .	5-26
RELEASE CATALOG. . . . .	5-27
RELEASE FILE . . . . .	5-27
MODIFY CATALOG . . . . .	5-27
MODIFY FILE. . . . .	5-29
LIST CATALOG . . . . .	5-32
LIST SPECIFIC. . . . .	5-33
Example of Line Delimiter Use. . . . .	5-33
Special Features . . . . .	5-34
RECOVERY Subsystem . . . . .	5-36
Recovery Operation. . . . .	5-37
Questions and Responses. . . . .	5-38
Time Sharing Media Conversion Program. . . . .	5-39
Operational Description . . . . .	5-39
Definitions . . . . .	5-41
Errors. . . . .	5-42
Binary Card Format. . . . .	5-42
Sample Deck Setups. . . . .	5-43
Section VI	
Deferred Processing . . . . .	6-1
Introduction . . . . .	6-1
User Interface . . . . .	6-1
Functional Summary . . . . .	6-9
Section VII	
Command File Processing . . . . .	7-1
Introduction . . . . .	7-1
User Interface . . . . .	7-1
Command Loader Interface . . . . .	7-7
Supporting Commands. . . . .	7-9
Appendix A	
Executive Error Messages. . . . .	A-1

CONTENTS (cont)

	Page
Index . . . . .	i-1

ILLUSTRATIONS

Figure 2-1	Time Sharing System Subsystem Commands. . . . .	2-6
Figure 4-1	Keyboard for 775/785 Series VIP . . . . .	4-11
Figure 4-2	Display Devices . . . . .	4-12
Figure 5-1	Logical Structure of the File System. . . . .	5-10

## SECTION I

### TIME SHARING SYSTEM

#### SYSTEM DESCRIPTION

The Time Sharing System operates under the direction of the General Comprehensive Operating Supervisor (GCOS), and constitutes one dimension of an integrated, multidimension information processing system. Under GCOS, the multiprocessing dimensions carry on their activities simultaneously, with intercommunication existing between all processing dimensions. This intercommunication feature has considerable significance for the user of a time sharing terminal.

The Time Sharing System (TSS) consists of a Time Sharing Executive, a number of independent processing subsystems which operate under the Executive, and a common command language. The major subsystems of the Time Sharing System include the following:

- ABACUS -- A desk calculator facility featuring complex algebraic capabilities such as functions, summation operations, and remembered variables. For further information see Section V.
- BASIC -- An algebraic-language compiler/executor designed for the user with numerical calculations involving relatively small quantities of data.
- CARDIN -- A facility for submitting a punch card format job at a time sharing terminal for processing as a batch job. Job status is available on request. The JOUT subsystem complements CARDIN by providing the capability to scan the job output.
- dataBASIC -- Honeywell's dataBASIC subsystem provides for data base management and inquiry by combining data base manipulation capabilities with a BASIC type language. It permits a file to be constructed, maintained, retrieved, and deleted on a content-addressable basis.
- TEXT EDITOR (and RUNOFF) -- A facility for building, maintaining, and reformatting text files.
- TSS ALGOL -- An ALGOL subsystem that gives the time sharing user the capabilities of the ALGOL language.
- TSS JOVIAL -- A JOVIAL subsystem that provides the time sharing user with the capabilities of the JOVIAL language processor.
- TSS FORT -- A time sharing based FORTRAN subsystem. Refer to the FORTRAN manual.
- TSS YFORT -- A batch-based FORTRAN subsystem. Refer to the FORTRAN manual.

The following subsystems and subroutines provide service and utility functions for the Time Sharing System:

- ACCESS -- is a file system manipulation subsystem that allows the user to create, delete, and modify file system catalogs, subcatalogs, and named files. The file space, not file content, is manipulated with ACCESS. See Section V.
- Command Loader -- is a default subsystem which will be invoked whenever an unrecognized command is given, either at system-selection level or in line numbered build mode. The input is assumed to be the catalog/file description of an H\* file to be loaded and executed or a series of responses for a command file application. See TSS System Programmer's Reference Manual.
- FDUMP -- is a remote-terminal, word-oriented file inspection and maintenance facility for files, regardless of their format. The files may have been generated in either batch, remote batch, or time sharing environments. See the TSS Terminal/Batch Interface manual for details of the FDUMP subsystem.
- File and Record Control Subroutines (TSS) -- provides File and Record Control subroutines needed for FORTRAN, ALGOL, and JOVIAL. These subroutines may also be used in COBOL or may be called directly by programs written in GMAP. These subroutines also provide automatic functions for dealing with the variety of file and device types available on the system. See the File and Record Control manual.
- HELP -- permits a terminal user to obtain a detailed explanation of any system error message. See Section III.
- JOUT -- provides a means for inspecting output from batch jobs. The batch job could be a CARDIN job with a disposition code of J or JOUT, a remote terminal batch job (GRTS), or a job submitted at the central site. See the TSS Terminal/Batch Interface manual for details of the JOUT subsystem.
- TRACE -- a powerful, conversational debug tool that permits a time sharing program to be executed in a controlled environment. See the Debug and Trace Routines manual for details of the TRACE subsystem.
- LODS -- provides a debugging environment for a specified Time Sharing subsystem by loading the Debug Trace Package with the subsystem.
- LODT -- similar to LODS, except the debugging environment is provided for a user program resident on an H\* file.
- LODX -- allows the user to load and execute a program resident on an H\* file.
- Media Conversion Program -- is a batch-world program that may be run either at the central computer site or entered through a remote batch terminal. It generates a standard format, time sharing text file from a suitable card deck, or conversely, produces a card deck from such a file.
- CONVERT -- provides for the conversion of textual information between physical file formats, the reformatting of files, and the initiation of batch jobs. See the TSS Terminal/Batch Interface manual for details of the CONVERT subsystem.



- RBUG -- is a conversational debug routine that can be used in conjunction with CARDIN. RBUG has all of the capabilities of the DEBUG routine of the batch world, permitting the user to monitor execution of his program, insert and remove breakpoints, and alter contents of memory locations and registers dynamically, all in an interactive manner. See TSS Terminal/Batch manual.
- SABL -- retrieves specific locations of the ABRT file for printing at the user's terminal or optionally on the central site printer. The file named ABRT must have been created by the user and entered into his Available File Table (AFT). When the system aborts the user's program, the memory storage area containing the program is written to the ABRT file.
- SCAN -- provides a means of examining output of a batch job from a time sharing terminal; the batch job may have been submitted through CARDIN, remote batch, or as a central site job with the output placed into the file system. See TSS Terminal/Batch manual.

The primary functions of the time sharing command language are as follows:

- Initiation of processing within a subsystem (e.g., LIST and RUN commands)
- Storage, retrieval, and purge of permanent files (e.g., SAVE and OLD commands)
- Request for operations on temporary time sharing files (e.g., NEW and RESEQUENCE commands)
- Request for pertinent operating information (e.g., HELP and STATUS commands)
- Direction of flow of control within the subsystem (e.g., DONE and BYE commands)

The command language is described in Section II along with an explanation of time sharing file usage.

In addition to the usual time sharing facilities at his disposal, the Time Sharing System user also has access to remote batch facilities. This capability is provided by a group of functionally interrelated subsystems called the Terminal/Batch Interface Facility. The time sharing terminal user can perform the following operations:

- Access and modify a file of information created in the batch or remote batch dimension.
- Submit a job, such as a GMAP assembly and execution, to the batch dimension and inspect the output directly from a terminal.
- Establish conversational communication between a batch program and the user's terminal.
- Use an adjacent remote batch terminal as a high volume, hard copy output device, and, indirectly, as a high volume input device.

The basis for this communication between the several processing dimensions is the GCOS File System, which provides a common data base for all users of the system, and the common interface provided by GCOS. The file system provides automatic storage and retrieval of symbolically named permanent files on high capacity storage devices. These files are readily accessible in any processing mode. As a byproduct, the use of physical file volumes, such as card decks and tape reels, actually handled and stored by the user is considerably de-emphasized.

Considerable effort has been made to standardize error messages and comments for all subsystems in the Time Sharing System, and to have error message explanations immediately available at the terminal. Identical error or exception conditions arising in different subsystems are identified by the same error message text. Those messages that are not fully self-explanatory are prefixed with a message number enclosed by carets (i.e., <nn>), in almost all cases. This message number relates to a message explanation as given by the HELP subsystem. Upon encountering an error message that he does not fully understand, the user can call the HELP subsystem and give the error message number when the number is requested. He will then receive an explanation of the error condition and suggestions as to possible courses of remedial action.

The Time Sharing System is completely modular and open-ended in that it is explicitly designed to allow user implemented subsystems, tailored for a specific application, to be added to the Honeywell-supplied subsystems. This implementation of subsystems can be done readily, with no disturbance to the system. Specialized debugging facilities are provided for the checkout of new subsystems simultaneous with normal time sharing operation.

## SECTION II

### COMMAND LANGUAGE AND FILE USAGE

The command language controls the operation of various time sharing subsystems. By using the command language, the user communicates with the computer throughout his session at the terminal. \*

Some of the commands are recognized by all of the time sharing subsystems. Conversely, some subsystems have a unique command language which is recognized only after the subsystem has been called into operation. The relationship between subsystems and commands is summarized in Figure 2-1.

The user's response to subsystem selection (see Section IV) calls the specified subsystems into operation. At this level, the user can be in the build mode or the Direct Access (DAC) mode depending upon the subsystem and the command. In the build mode, the user is building or adding to a file. In the DAC mode, the command either activates specialized subsystems or initiates a conversational question/answer sequence (as in CARDIN). The RUN command, where applicable, always implies a change to the DAC mode. A number of minor subsystems have no build mode phase and go into DAC immediately upon selection. herefore they recognize no command language.

NOTE: The CONVERT subsystem (see the TSS Terminal/Batch Interface manual) eliminates the question/answer sequence.

#### DEFINITIONS

- Line Numbers

Line numbers are required by the BASIC, ALGOL, JOVIAL, FORTRAN, and CARDIN subsystems for line sequencing purposes. In the case of BASIC, line numbers are also used as statement numbers. A line number consists of one to eight numeric characters (including leading blanks) appearing within the first eight character positions of the line. A line number may be terminated by any nonnumeric character (including a blank). The pound sign (#) is not part of the line number, although like other nonnumeric characters, it can be used to terminate a line number. It also has a variety of other uses associated with line numbers.

- Manual Mode

In manual mode, the user must provide (type) the line numbers for each line.

- Automatic Mode

In automatic mode, the system provides the line numbers. They are printed as the build mode request for input (asterisk) is issued. The number is written onto the collector file as a part of the statement.

- New File

A new file is a temporary file created for the user when the NEW command is given. It is assumed the user will build a file which then may be saved, thus creating an old file. A new file is created by a (destructive) reinitialization of the current file.

- Old File

An old file is a previously built and saved file which the user selects with the OLD command or response, naming the desired file. The old file is copied onto the current file where it is available to the user for processing or modification.

- Current File

The current file (\*SRC, referenced by \*) is a temporary file assigned to the user, on which a new file is built or on which the selected old file is copied. Regardless of the intervening commands or subsystem selections, the current file contains the last NEW or OLD selection, with whatever modifications that may have been entered. The modifications are, therefore, temporary until the file is saved by means of the command SAVE, or RESAVE.

- Collector File

The collector file (SY\*\*) is a temporary file assigned to each user when he logs on. All input which is not a recognizable command is gathered onto this file -- for example, numbered statements. Then, when the file becomes full or a command is typed, depending upon the subsystem, the collector file is merged with the current file and the entire current file is edited and sorted if necessary. For example, when the commands RUN, LIST, or SAVE are encountered in the BASIC subsystem, and data exists in the collector file, it is merged with the current file in sort order. (The collector file is normally transparent to the user.)

- Available File Table

An Available File Table (AFT) is provided for each Time Sharing System user. This table holds a finite number of file names (currently set at 20 including SY\*\* which always remains in the AFT) which are entered in the AFT when the files are initially accessed (opened). The advantages of the AFT are:

1. Files requiring passwords or long catalog/file descriptions may be referenced by file name alone, once they have been entered in the table.
2. Files used repeatedly remain readily available, thus reducing the overhead time and cost of accessing the file each time.

The following commands cause the named permanent files to be placed in the AFT.

RUN	<u>filename</u> (s)
OLD	<u>filename</u> (s)
SAVE/RESAVE	<u>filename</u> (s)
GET	<u>filename</u> (s)
PRINT	<u>filename</u> (s)
PERM	tempfile, <u>filename</u>
ERASE	<u>filename</u> (s)
CODE/DECODE	<u>filename</u> ; <u>filename</u>
LENGTH	<u>filename</u> (s)
LODX/LODT	<u>filename</u>

Because the AFT is of finite length, it can become full. If this happens and a command is given which requires a new filename to be placed in the AFT, the command subsystem will print an error message indicating that the AFT is full. At this point, the user must remove any unneeded files from the AFT in order to continue. The STATUS FILES command produces a listing of all of the user's files in the AFT. The REMOVE command can be used to remove specified files from the AFT. The files are not purged or altered in any way; only the name is removed from the AFT and the file is set "not busy". All files (except SY\*\* and \*SRC) may be removed by a REMOVE CLEARFILES command.

#### FILE DESIGNATION

The designation of permanent files in the following discussion of commands is specified in the following formats:

1. filename where the filename only is required.
2. filedescr where the full file description may be used, in any of the following formats:
  - a. filename
  - b. filename\$password
  - c. userid/catalog\$password...  
/catalog\$password/filename\$password

If a required password is stored incorrectly or not given, the system will explicitly ask for the proper password.

If the file was previously opened (e.g., with a GET), only the filename need be given regardless of its full description. If the requested file is not already open, it must emanate directly from the user's master catalog (quick-access type file) in order for format a to be applicable.

As a general rule, use of formats b or c will result in removing the file from the Available File Table (AFT) before attempting to access it, if necessary.

Where desired permissions and/or alternate name are applicable, they are specified in the following format:

filedescr,permissions

or

filedescr"altname",permissions

Where: Altname may be a valid file name (one to eight characters), enclosed in double quote signs.

permissions may be any one of the following:

READ (R)

WRITE (W)

EXECUTE (E)

APPEND (A)

READ,WRITE (R,W)

READ,APPEND (R,A)

Where a desired permissions specification is applicable, a null permissions field implies READ and WRITE permissions; i.e., the default interpretation for desired permissions is R,W.

If a file segment specification, of the form (i,j) where i and j are line numbers, is given in addition to desired permissions and/or alternate name, it must appear last in the specification string; e.g.:

filedescr,permissions(i,j)

or

filedescr"altname",permissions(i,j)

Examples:

OLD FIL1\$GOGO,R

SAVE /CAT1/CAT2\$MAYI/FILO\$HERE

LIST FILE2\$HOHO(1,100)

PURGE FIL3\$ARIZ;FIL4;FIL5\$SUN

GET JJONES/DATACAT/BATCHWRLDFIL"INFILE"

FILE DESCRIPTION SUBCATALOG LEVEL

The maximum number of subcatalogs permitted for file descriptions is eight for all TSS subsystems, with exceptions only for DRUN, CRUN, and COUT wherein only three levels are allowed.

## FILE NAMES, CATALOG NAMES, AND PASSWORDS

File names for time sharing usage are normally eight characters or less in length, and must be composed of alphanumerics, periods, and minus signs. Catalog names and passwords may be up to 12 characters in length, and composed of the same characters as file names.

If a batch file with a name longer than eight characters (12 characters maximum) is to be accessed, it must be given an alternate name (altname) from one to eight characters in length. The renaming is local and temporary. An altname may also be used to temporarily rename one or more of several duplicately named time sharing files the user wishes to have accessed concurrently. Permanent files may be duplicately named so long as they emanate from different catalogs or subcatalogs, but if they are used, they must be appended with an altname.

## SYSTEM SELECTION

Although a system selection query is never issued, it is possible to select a system by simply entering the desired selection at the build mode level. The selection can optionally be accompanied by an OLD or NEW request; if an OLD or NEW request is not made, the current file is retained, provided one exists. Once a system selection has been made, that system remains in effect until explicitly changed (or cancelled by way of the break key).

Conversely, system selection need never be made directly. The use of a particular execution command informs the Time Sharing System as to what system the user wishes to employ. The execution command and its called system are as follows:

<u>Execution Command</u>	<u>System</u>
BRN	BASIC
FRN	FORTRAN
JRN	CARDIN

Refer to "Commands" below for descriptions of these execution commands. Note that JRN cannot be used as an execution command for a CARDIN program unless that program follows CONVERT subsystem syntax.

## COMMANDS

Following is a description of the Time Sharing System subsystem commands. Although the command words are spelled out completely in the following descriptions, in general usage those exceeding four characters may be shortened to the first four characters (e.g., RESEQUENCE or RESE).

NOTE: Commands ASCBCD, BCDASC, ASCASC, BPUNCH, BPRINT, RUN(CARDIN) and PRINT are superseded by formats of the CONVERT subsystem but will remain available for use until some future software release.



Figure 2-1 gives a complete list of Time Sharing System subsystem commands. The commands are applicable to all major subsystems described in Section I except where noted in Figure 2-1. The Text Editor must be in the direct-mode to use most of the commands. The exceptions are: #AUTO, #LUCID, #RECOVER, #ROLLBACK, and #TAPE.

ABC	DSTS	NOPARITY
ACCESS	EDIT	OLD
APRINT	ERASE	PARITY
ASCASC	FDUMP	PERM
ASCBCD	FRN	PRINT
AUTOMATIC	GET	PURGE
#AUTO <sup>a</sup>	HELP	RECOVER
BCDASC	HOLD	#RECOVER <sup>a</sup>
BPRINT	JABT	RELEASE
BPUNCH	JDAC	REMOVE
BRN	JOUT	RESAVE
BSEQUENCE	JPRINT <sup>b</sup>	RESEQUENCE
BYE	JPUNCH <sup>b</sup>	ROLLBACK
CATALOG	JRN <sup>b</sup>	#ROLLBACK <sup>a</sup>
CMOD	JSTS	RUN
CODE	LEADER	RUNOFF
CONNECT	LENGTH	SABT
CONVERT <sup>b</sup>	LIB	SAVE
COUT	LINE	SCAN
CPY	LIST	SEND
CPOS	LODT	SEQUENCE
CRUN	LODS	SMCL
DABT	LODX	STATUS
DECODE	LOGOFF	STRIP
DELETE	LUCID	SYSTEM
DISPLAY <sup>b</sup>	#LUCID <sup>a</sup>	TAPE
DONE	NEW	#TAPE <sup>a</sup>
DRUN	NEWUSER	

<sup>a</sup> Text Editor build mode only and not applicable at the subsystem selection level.

<sup>b</sup> CONVERT subsystem commands.

Figure 2-1. Time Sharing System Subsystem Commands

*System?*  
*Edit*  
*FORT*  
*B/...*

- ABC

Calls the ABACUS subsystem for algebraic-expression evaluation.

- ACCESS

Calls the ACCESS subsystem for time sharing interface with the file system.

- APRINT

Calls the APRINT subsystem for printing either standard time sharing ASCII files or files resulting from a RUNOFF/REFORM process on printers equipped with an ASCII print train.

- ASCASC filedescr-1; filedescr-2

This command, issued under Series 6000 FORTRAN and other time sharing language systems, causes the translation of a time sharing format ASCII file to a standard-system-format ASCII file or vice versa. In both translations, file 1 is converted to the format required in file 2. If file 1 is in time sharing ASCII format (record media code 5), the characters of the file are read and converted to the word-oriented standard system ASCII format for file 2. File 2 may then be used as input data for the language system. If file 1 is in standard system ASCII format (record media code 6), the words in the file are read and converted to the character-oriented time sharing ASCII format for file 2. File 2 may then be listed at a terminal. The question and answer sequences for this command depend on the format of the file to be converted. For information about record media codes see the TSS System Programmer's Reference Manual, Section VI.

- ASCBCD ascfil; bcdfil

Under CARDIN, the ASCII time sharing file specified by ascfil is converted to a standard system format BCD file on the permanent file specified by bcdfil, following the question/answer sequence that is initiated by this command if the former file does not contain first line reformatting information. Both ascfil and bcdfil may be simply a file name or a full file description, as required. The ascfil field may specify also the current file by an asterisk.

- AUTOMATIC

1. AUTOMATIC

Causes the automatic creation of line numbers, by the system, at the point at which the automatic mode is entered (or re-entered), with line numbers initially starting at 010 and incrementing by 10 (or, on re-entry, resuming where the previous automatic numbering left off). These line numbers appear in the terminal copy, and are written in the file, just as though the user had typed them.

2. AUTOMATIC n,m

Causes the automatic creation of line numbers, as above, but starting with line number n and incrementing by m.

3. AUTOMATIC ,m  
AUTOMATIC n,

Causes automatic creation of line numbers beginning at 10 and incrementing by m, or beginning at n and incrementing by 10 (on re-entry, the line numbering resumes where it left off).

Normally the line number will be followed by a blank. Any nonblank, nonnumeric character affixed to the end of the command AUTOMATIC will cause the blank to be suppressed. For example: AUTONB or AUTOMATICX.

No commands are recognized while in the automatic mode. The automatic mode is cancelled by giving a carriage return immediately following the issuance of an asterisk and line number by the system. The user may not use character delete (@) or line delete (CTRL X) to delete characters associated with the generated line number or its associated blank.

• BCDASC bcdfil; ascfil

Under CARDIN, the standard-system-format BCD file (permanent) specified by bcdfil is converted to an ASCII time sharing file on the permanent file specified by ascfil, following the question/answer sequence that is initiated by this command. Both bcdfil and ascfil may be simply a file name or a full file description, as required. The ascfil field may also specify the current file by an asterisk.

• BPUNCH  
• BPRINT

1. BPUNCH ascfil  
BPRINT ascfil

Under CARDIN, the contents of the ASCII time sharing file specified by ascfil are converted to BCD and are punched or printed, respectively, at the central computer site, following a question/answer sequence initiated by these commands if the file does not contain first line reformatting information. These commands allow the user to create hard copy backup (cards) for his TSS files, and to list long files on a high speed printer. Ascfil may be simply a file name or a full file description, as required. The ascfil field may also specify the current file by an asterisk.

Since a batch BMC job is spawned by these commands, the batch \$ IDENT card information is requested by the subsystem.

2. BPUNCH F1;\*;F2;F3;.....;Fn  
BPRINT F1;\*;F2;F3;.....;Fn

Concatenates the named files (\* is used to denote the current file) and punches or prints them as one file. For maximum efficiency all the files should contain a first-line format record.

- BRN

Identical to RUN when operating under BASIC, but is independent of the system selection that has been previously made, if any.

- BSEQUENCE

Identical to RESEQUENCE when operating under BASIC, but is independent of the system selection that has been previously made, if any.

● BYE

Causes the computation of the user's system usage charges during the session and disconnects the terminal.

Depending upon the last selected subsystem, the AFT may first be scanned for user's temporary files. A message is issued as to the number of temporary files, then the user is queried as to the disposition. Each filename is printed followed by a question mark. The user may respond as follows:

1. carriage return - implies the file is to be released; pass to next file.
2. NONE - implies all of the succeeding files are to be released.
3. SAVE filedescr - specifies that the file is to be saved on the permanent file described by filedescr. (Refer to the PERM command.) The user is not queried for the disposition of a temporary ABRT file in his AFT.

● CATALOG

1. CATALOG

Lists all catalog and file names which emanate from the user's own master catalog.

2. CATALOG #LIB

Lists all file names in the library.

3. CATALOG #CMD

Lists catalog and file names emanating from the command library (CMDLIB).

4. CATALOG filename

Prints a list of the attributes of the file specified. The file must emanate from the user's catalog.

5. CATALOG, n, x (mm-dd-yy), S, A, R, FIRST/name/

Allows limited listing of catalog and file names. Optional fields are:

x, to specify whether the date is date created (C), date of last access (A), or last date the file was changed (L)  
date, in the form mm-dd-yy is required when C, A, or L is requested.

n, number of files to be listed, starting from the most recent to oldest

R, reverse the list before printing

S, sort the names

A, list in abbreviated fashion (eight per line)

Any option may be omitted and the order given is immaterial. The FIRST/name/ allows the cat/file list to start at the specified name.

6. CATALOG /catalog1/catalog2

Prints a list of all catalog and file names which emanate from the specified catalog (catalog2 in this case).

7. CATALOG catadescr,n,x(mm-dd-yy),S,A,R,FIRST/name/

Allows limited listing of catalog and file names emanating from the catalog implied by catadescr. Options are as described in (5) above.

8. CATALOG /catalog1/catalog2\*

Prints a detailed list of catalog2's attributes.

9. CATALOG userid\*\*

Lists catalog and file names emanating from the UMC of the specified USERID. The requesting user must have modify permission at the UMC level of the specified USERID.

A user who is the originator of (or has modify permission for) a catalog belonging to another user may list, purge, or release the catalog. This also applies to a specific list request for a catalog or file. Permission is not required to list strings which originate from the user's own catalog, the library (#LIB), or the command library (#CMD). Passwords need not be given in the catalog structure unless the specified file or catalog was created by another user.

• CMOD operation-1;operation-2;...;operation-n

Allows modification of the user's Program Switch Word (PSW). Each operation is an arithmetic or Boolean operation and affects only the lower half (bits 18-35) of the PSW. An operation may include any of the following forms:

<u>FORM</u>	<u>MEANING</u>	<u>RESTRICTIONS</u>
n	Set C(PSW) bits 18-35 to n	$n < 262144$
+n	Add n to C(PSW) bits 18-35	$n < 262144$
-n	Subtract n from C(PSW) bits 18-35	$n < 262144$
Si	Set bit i	$18 \leq i < 36$
Si-j	Set bits i through j	$18 \leq i \leq j < 36$
Ri	Reset bit i	$18 \leq i < 36$
Ri-j	Reset bits i through j	$18 \leq i \leq j < 36$
Ni	Negate bit i	$18 \leq i < 36$
Ni-j	Negate bits i through j	$18 \leq i \leq j < 36$
D	Display C(PSW) bits 0-35	
E	Store error code in PSW bits 18-35 (See Appendix A)	

• CODE

CODE is used to encrypt a file, making its content unintelligible to others. Input and output file descriptions may accompany the command, separated from one another by a semicolon. The input file may be random or sequential (\* designates the current file) and either description may include passwords, permissions, an alternate name and up to eight levels of subcatalogs. The output file will be created for the user if it does not already exist.

\*  
Upon receipt of the CODE command, the user is asked to enter (and verify) a key to be used for encrypting the file. This is simply a one- to 11-character password which may include any combination of uppercase and lowercase alphabets and nonprinting characters such as BELL, TAB, SPACE, etc. The encryption key is known only to the user; i.e., it is neither resident on mass storage nor retained anywhere in the system.

DECODE is used to obtain the original version of an encrypted file. Input and output file descriptions are required as for CODE and are managed in a similar manner. The output file may be \* to designate the current file. Upon receipt of the DECODE command, the user is asked to enter the encryption key. This is the same key originally used with the CODE command to encrypt the file.

File encryption offers a level of data base security unsurpassed by other methods presently available for privacy protection. Sensitive information can be safely stored in the system without fear of penetration by unauthorized individuals. Even if a penetrator obtains the coded file and knows the encryption algorithm used to produce it, the file cannot be decoded except by trial and error -- a process obviously prohibitive by the 11-character key length. Note that it is, of course, necessary to destroy the original file once it has been encrypted. This may be conveniently accomplished with post use of the PURGE or ERASE command.

● CONNECT XXXX

Permits a user, who does not explicitly log-off the system with the "BYE" command (this disconnect may be caused by noise on the phone line, accidentally disrupting the data set, accidentally hitting the clear key, or by accidentally hitting control C instead of control X for line delete), to call back in and reconnect to the same User Status Table (UST) previously used. This would allow the user to continue, without losing data, at the same point as he was disconnected or a point prior to the disconnect. "XXXX" is the channel number from the previous sign-on period.

Example:

0110401

HIS SERIES 6000 ON 01/30/77 AT 5.241 CHANNEL 2140

USER ID- SMITH

PASSWORD--

XXXXXXXXXXXXXXXXXXXX

\*CARDIN OLD TEST

\*STAT

CHANNEL 2140

USER STATUS ON JAN 30, 1977 AT 5:14:51 LOG-ON AT 5:13:53

PROC TIME USED 0.00 SEC., 1 FILE I/O 80 CHAR KEY I/O

LIST OF OPEN FILES : TEST

\* LIST

10 THIS IS A TEST

(line drops)

user re-dials

0110401

HIS SERIES 6000 ON 01/30/77 AT 5.257 CHANNEL 2150

USER ID - SMITH\$JOHN

\*CONNECT 2140

10 THIS IS A TEST

20 OF THE RECONNECT            OUTPUT CONTINUES

30 FEATURE FOR 2/H            FROM PREVIOUS LIST

40 END

\* STATUS

CHANNEL 2150

USER STATUS ON JAN 30,1977 AT 5:15:51 LOG-ON AT 5:13:53

PROC TIME USED 0.05 SEC., 2 FILE I/O 264 CHAR KEY I/O

LIST OF OPEN FILES: TEST

\*

- CONVERT (input files) (=output files) (:options)

Is a command of the CONVERT subsystem. See the TSS Terminal/Batch Interface manual for details of the CONVERT subsystem.



• COUT

1. COUT

Permits all output accumulated since either the beginning of a command file application or previous COUT to be directed to the user's remote device. Used in conjunction with command file processing.

2. COUT filedescr

All accumulated output is directed to the designated file. Used in conjunction with command file or deferred processing.

3. COUT \*

Designated file is user's current file.

4. COUT;EXCLUDE (or) filedescr;EXCLUDE

This optional parameter used to eliminate user lines containing responses from the output.

• CPY filedescr-in;filedescr-ot [n]

This command may be used to copy all or part of the contents of one file to another. The first two parameter declarations are required and constitute file descriptions for the input and output files, respectively. Either description may include passwords, permissions, an alternate name and up to eight levels of subcatalogs. The third parameter is optional and, when present, specifies the (decimal) number of llinks to copy.

Either random or sequential files may be copied. Furthermore, a random file may be copied to a sequential file or vice versa. The output file will be created (in the same mode) for the user if it does not already exist. If this is necessary and the number of llinks to copy is specified, then the file is created with a corresponding current size. Otherwise, the size requirement is determined by examining the input file. If the input file is sequential and conforms to standard system format, then the output file is created large enough to accommodate the first file. If the input file is random and constitutes an H\* or Q\* in system loadable format, the output file is created only large enough to accommodate the information that is necessary. If the input file content is not recognized, the output file is created the same physical size as the input file and the entire file content is copied.

- CPOS expression; operation

Permits either conditional alteration of the normal serial processing of input file responses for a command file application or conditional execution of a single command, based on the contents of the PSW (Program Switch Word). The first parameter represents a Boolean expression which, when true, causes the operation implied by the second expression to be performed. Expression operands may include any of the following:

<u>FORM</u>	<u>IS TRUE WHEN:</u>	<u>RESTRICTIONS</u>
i	Bit i of the PSW is on	$0 \leq i < 36$
LTn	C(PSW) bits 18-35 is less than n	$n < 262144$
LEn	C(PSW) bits 18-35 is less than or equal to n	$n < 262144$
GTn	C(PSW) bits 18-35 is greater than n	$n < 262144$
GEN	C(PSW) bits 18-35 is greater than or equal to n	$n < 262144$
EQn	C(PSW) bits 18-35 is equal to n	$n < 262144$
NEN	C(PSW) bits 18-35 is not equal to n	$n < 262144$

Permissible operators include + (OR), \* (AND), - (EXCLUSIVE OR) and / (NOT). Although / is a unary operator involving only one operand, by convention A/B is taken to mean A\*/B.

- CRUN

1. CRUN filedescr-in

Initiates the processing of the input command file (filedescr-in) which must contain the user's terminal responses in the proper order. When only an input file description is specified, or is followed by two consecutive semicolons, all keyboard output generated for the command file application is directed to the user's remote device upon completion of the application.

2. CRUN filedescr-in;filedescr-ot;option-1;option-2;...;option-n

Optional CRUN parameters include the following:

<u>FORM</u>	<u>MEANING</u>
nn...n	Maximum processor time limit
(arg-1,arg-2,...,arg-n)	Substitutable arguments
ARG/c	Substitution-implying character
DEL/c	Delete character
INCLUDE or EXCLUDE	Output file user response disposition. INCLUDE is assumed by default.
UPPER or LOWER	Output file case disposition

3. CRUN \*;filedescr-ot or filedescr-in;\* or \*;\*

Asterisk (\*) is used to designate the user's current file to be the input file, output file, or both.

4. "response-1\response-2\...\response-n"

Information within the quoted string constitutes the input file.

• DABT

Permits deferred sessions originated by the user to be aborted or inactive entries on the deferred queue file to be removed.

1. DABT nnnnD<sub>1</sub>;nnnnD<sub>2</sub>;...;nnnnD<sub>n</sub>

Aborts the specified deferred jobs.

2. DABT ALL

Aborts all deferred jobs belonging to the requesting user.

3. DABT REMO

Remove all jobs from the deferred processing queue file which are no longer active.

• DECODE

The DECODE command is used to obtain the original version of an encrypted file. Input and output file descriptions are required as for CODE and are treated in a similar manner. The output file may be \* to designate the current file. Upon receipt of the DECODE command, the user is asked to enter the encryption key. This is the same key originally used with the CODE command to encrypt the file. Key verification is not required with DECODE.

• DELETE

1. DELETE a,b,c,d,...

Lines a, b, c, and d are deleted from the current file.

2. DELETE a-b,c-d

Lines numbered a through b and c through d are deleted from the current file.

3. DELETE a,b,c-d,e,f-g,...

Lines numbered a,b,c through d,e, and f through g are deleted from the current file.

4. DELETE -n

Acceptable only as first argument, since it implies deletion of lines from beginning of current file to line n.

5. DELETE n-

Acceptable only as last argument, since it implies deletion of lines n through end of current file.

6. DELETE;\*

Causes deletion of all lines in current file.

• DISPLAY (input files)(:options)

Is a command of the CONVERT subsystem. See the TSS Terminal/Batch Interface manual for details of the CONVERT Subsystem.

• DONE

Causes return from selected subsystem.

• DRUN filedescr-in;filedescr-ot;option-1;option-2;...option-n

Initiates deferred processing of filedescr-in optionally at a later time and date. Input and output file descriptions (filedescr-in and filedescr-ot) are required and must constitute the first two parameter declarations. Optional parameters may follow the output file description in any order. Since the user does not actively participate in the dialog exchange with the computer during the deferred session, the user's responses must be anticipated in advance and be provided in the input file in the proper order.

• DSTS

Permits status inquiries for deferred sessions originated by a user.

1. DSTS nnnnD ;nnnnD ;...;nnnnD

Prints the status of the requested job identifiers.

2. DSTS ALL

Prints the status of all deferred jobs belonging to the requesting user.

• EDITOR

Causes the Text Editor subsystem to be called into use. Whereupon the user may exercise any of the text editing capabilities available in the Text Editor subsystem. The current file is the recipient of any modification. An editor command preceded by a hyphen (-) can be entered while in the line number build mode.

• ERASE filedescr-1;filedescr-2;...;filedescr-n

Erases (overwrites with zeros) the file space associated with the specified file(s), but does not release the file(s) from the file system. (Refer to PURGE and RELEASE commands.) \* may be used to designate the current file.

- FDUMP

Calls the FDUMP subsystem for file dumping and correction. Refer to the TSS Terminal/Batch Interface manual for details of the FDUMP subsystem.

- FRN

Identical to RUN when operating under FORTRAN, but is independent of the system selection that has been made, if any.

- GET filedescr-1;filedescr-2;...;filedescr-n  
(permissions and altname applicable)

The permanent file(s) designated by filedescr-i will be accessed and the filename(s) placed in the AFT. This is a simple means by which common data files emanating from other users' master catalogs may be opened. A linked file can be accessed in random mode by specifying MODE/RANDOM/, or more simply, M/R/ following the file description.

- HELP

Calls the HELP subsystem to obtain an error message explanation. For example, if the error message

009-SYSTEM UNKNOWN

were issued, the user could call HELP and respond to the request

PLEASE ENTER MESSAGE NUMBER-

with 9, if he desired an error message explanation.

- HOLD

Prevents any console or master user issued warning or information message from appearing at the terminal, either in printer or paper tape output, until a subsequent \*SEND command is given. The user assumes responsibility for any warnings he may miss while the HOLD is in effect. This command is used primarily during output of listings for display or reproduction purposes. (Refer to the SEND command.)

- JABT snumb (Job Abort)

Under CARDIN, causes the batch job specified by snumb (and submitted from the same terminal) to be aborted, with an X1 abort code assigned. Only jobs containing a \$ USERID control card belonging to the requesting user can be aborted.

- JDAC name (Job Direct Access)

Allows a time sharing terminal user to establish Direct Access Communication (DAC) with a slave program running in the system. The DAC is initiated at the subsystem level by

\*JDAC name \*

Name refers to the name of a user-supplied DAC slave program (e.g., the Time Sharing System is a DAC slave program). If the program name is not provided in the initial call to JDAC, the system will request a program name. When the direct access program terminates, the return is to build mode level.

- JOUT snumb

Permits manipulation, from a time sharing terminal (via a call to JOUT subsystem), of the output of certain types of batch jobs. Refer to the TSS Terminal/Batch Interface manual for details of the JOUT subsystem.

- JPRINT } (input files)(:options)
- JPUNCH }

Are Commands of the CONVERT subsystem. See the TSS Terminal/Batch Interface manual for details of the CONVERT subsystem.

- JRN (input files) (:options)

Is a command of the CONVERT subsystem. See the TSS Terminal/Batch Interface manual for details of the CONVERT subsystem.

- JSTS snumb (Job Status)

Causes the current batch processing status of the job specified by snumb (e.g., 0005T) to be printed at the terminal, in plain text.

- LEADER  
LEADER title

Causes a title to be punched in bold, block letters in the paper tape and then list the current file. If the first form is used, the title will be requested. In either form, the date in the international standard format will be punched in block letters after the title has been punched. Although only upper case characters are punched, the title can be composed of upper or lower case alphabets, numerics and special characters except the commercial at sign. After the title and date has been punched, a carriage return, line feed and eight rubouts will be punched before the contents of current file (\*SRC) are listed. After listing the current file, another carriage return, line feed, an X-OFF character and eight rubouts will be punched.

- LENGTH

1. LENGTH

Generates a report of the content length of the current file, in terms of 320-word blocks (llinks).

2. LENGTH filedescr-1;filedescr-2;..filedescr-n

Generates a report of the type, current size, and content length of the permanent file specified by filedescr-i. Size and content length are given in units of 320-word blocks (llinks). If the file is sequential and not ASCII, the media code is given and will be that of the first record of the first llink.

- LIB filename

File filename from the library becomes the current file.

- LINELENGTH n

Used to increase the length of an input line from a terminal, where n may be 80 through 160. Cannot be used on VIP type terminals.

NOTE: LINE or LINELENGTH followed by a carriage return will set the line length to 80 characters.

- LIST

1. LIST

Lists the current file on the terminal.

2. LIST i,j,k...

Lists only the specified lines i, j, k, etc., from the current file.

3. LIST filedescr (permissions and altname applicable)

Lists the file specified by filedescr on the terminal, without altering the current file. Filedescr must include at least one alpha character if it consists of filename only. Otherwise, the filename may be prefixed with a semicolon.

4. LIST filedescr-1(i,j);...;filedescr-n(i,j)  
(permissions and altname applicable)

Adjoins and lists the specified files or file segments on the terminal. The current file is not altered. The current file may be included in the list by the use of an asterisk (\*). If the list is greater than one line in length, it may be continued on the next line provided the last nonblank character on the first line is a (leading) delimiter.

5. LISTH

Lists the file with a header (date and time) printed at the top of the listing. LIST formats (1), (2), (3), and (4) may all use the LISTH form instead of LIST.

6. LISTEnnn (no intervening blanks allowed)

Lists the file(s) as specified by the operand; but with all lines to be "broken" or "folded" at the character position (nnn) specified. Listing of the line will be continued on succeeding line(s). If nnn is omitted, the value 72 is assumed. LIST formats (2) through (4) may also use the LISTEnnn form in place of LIST. Files containing overlength lines (records) may be listed in this manner.

7. LIST i-j

Lists all lines of the current file whose line numbers are greater than or equal to i and less than or equal to j.

8. LIST i-j, k-m...

Lists all lines of the current file from i through j and k through m, etc.

9. LIST filedescr(operand list)

Lists the lines of the file filedescr specified by operand list. Operand list may be any type described in 2, 7 or 8.

10. LISTI n

Lists each nth line of the current file starting with the first line.

11. LIST 99999999

If LIST is given with a line number greater than the last line number on the current file, then the last line number of the current file will be printed.

12. LISTL

Lists the last line number of the current file. A short form of LIST 99999999. Does not require a line number.



- LODS
- LODT
- LODX

Provide a loader function for user subsystem programs. Refer to TSS System Programmer's Reference Manual for details of the commands.

- LOGOFF

Causes the computation of the user's system usage charges during the session and disconnects the terminal.

- LUCID

This command is used instead of the TAPE command for non-ASCII paper tape input. The input is stored on a temporary file (TAP\*) file as unaltered eight-bit codes. The TAP\* file is left open (unedited in the user's AFT). When a pause greater than one second stops the tape read, the system returns to the subsystem selection level. This command does not function when data communication is via a Low Speed Line Adapter (LSLA) or an Asynchronous Communications Base (ACB) on a DATANET 355/6600 Front-End Network Processor. In the EDITOR subsystem, this command takes the form #LUCID.

NOTE: TAP\* can be copied to a permanent file by the user via the PERM command, if desired.

- NEW

1. NEW

A new file (empty current file) is started. (The system will return to the build mode.) The current file is cleared of any prior content.

2. NEWP filedescr (permissions applicable)

The named file is created by the NEWP command with the attributes specified and is opened with an alternate name of \*SRC. If the named file already exists, an error message is sent to the user. This file remains the user's current file until another form of the OLD or the NEW command is given.

3. NEWP# filedescr (permissions applicable)

Execution is the same as for NEWP except that the created file remains the user's current file until log-off, or until another OLDP, OLDP#, NEWP, or NEWP# command is given. The normal OLD or NEW commands use this file, (i.e., the file, specified by OLDP# or NEWP#) as the current file.

- NEWUSER

1. NEWUSER

Causes the computation of the user's system usage charges during the session and initiates a new log-on sequence.

2. NEWUSER account number

Causes the computation of charges for user's previous account number, this account number to be closed, and the new account number specified to replace the old. Accounting data is reinitialized as for a new user but the log-on sequence is bypassed; i.e., the previous user-id and password are assumed.

OLD

1. OLD filedescr (permissions and altname applicable)

File filedescr becomes the current file.

2. OLD filedescr(i,j) (permissions and altname applicable)

Lines i through j of file filedescr become the current file. Filedescr must be a line-numbered file.

3. OLD f-1(i,j);...;f-n(i,j) (permissions and altname applicable)

The n files or file segments are adjoined in the order listed and become the current file, where f is a filedescr. Adjoining of BASIC files should be done with caution (sequence numbers are also statement numbers). The asterisk designating the contents of the current file (or segment thereof) may appear as a filedescr anywhere in the file list.

Note that these files or segments are concatenated on the current file and resequencing may be required for satisfactory operation in line-number dependent systems. Sorting or resequencing is not automatic.

4. OLD f-1(i,j):...:f-n(i,j) (permissions and altnames applicable)

The n files or file segments are merged by line numbers, and become the current file, where f is a filedescr (colon-separated). If duplicately numbered statements appear in two or more files, each such statement appears in the order specified by the file list. If it is desired to retain only the last duplicately numbered statement, the colons may be replaced by pound signs (#). The asterisk designating the contents of the current file (or segment thereof) may appear as a filedescr anywhere in the file list.

5. OLD f-1(i,j);f-2(i,j);f-3(i,j);...:f-n(i,j)  
(permissions and altname applicable)

A combination of forms (3) and (4). Concatenation or merging is performed in the order (from left to right) indicated by the file list.

If the file list is too long for one line, the OLD subsystem will request more input when a delimiter is the last nonblank character before the carriage return.

6. OLDP filedescr (permissions applicable)

The specified permanent file is accessed with an alternate name of \*SRC and becomes the current file. This file is the user's current file until another form of the OLD or NEW command is given. The contents of the file will always be checked or verified for Time Sharing System format.

7. OLDP# filedescr (permissions applicable)

Execution is the same as for the OLDP command, except that this file remains the user's current file until log-off, or until another OLDP, OLDP#, NEWP, or NEWP# command is given. The normal OLD or NEW commands use this file (i.e., the file specified by OLDP# or NEWP#) as the current file. OLDP# can be cancelled by REMOVE \*SRC.

NOTE: The OLDN subsystem is called in when the commands OLD, NEW or LIB (normal forms) are given by the user. If a NEWP or OLDP command was issued and then one of the normal forms was typed in, OLDN will deaccess the permanent \*SRC file and assign a new temporary \*SRC file to the user. The permanent file remains in the user's catalog until he releases it.

If a NEWP# or OLDP# command was issued and then one of the normal forms was typed in, OLDN will retain the permanent file as \*SRC. If a NEWP or OLDP was typed in instead of the normal form, the permanent \*SRC will be deaccessed, and a new permanent file with the alternate name \*SRC will be created and/or accessed.

If a NEWP# or OLDP# command was issued and then followed by another NEWP# or OLDP# command, the OLDN subsystem will deaccess the present \*SRC file and then create and/or access the newly specified \*SRC file.

Merging and concatenation are not allowed with OLDP, NEWP, OLDP#, and NEWP#.

- PARITY/NOPARITY

1. PARITY

The data sent from the computer system to a terminal in direct access mode is normally in seven-bit, even parity code. The PARITY command is only used to return to this mode of operation from a NOPARITY mode of operation.

2. When the NOPARITY (NOPA) command is given, subsystem output sent from the computer system to a terminal in direct access mode is in eight-bit, parity independent code. This command may be used at the system level or at the command level in BASIC, FORTRAN and CARDIN. The NOPARITY (NOPA) command can only be used with a Type 4 terminal (teleprinter).

NOTE: Error messages and messages sent by the TSS Executive will be sent with even-parity even though the command NOPARITY was issued by the terminal operator.

- PERM tempfile;filedescr

The temporary file tempfile is copied onto the permanent file described by filedescr. If the file does not already exist, it will be created with the permissions and/or password accompanying the file description. Both the temporary and permanent files will be removed from the AFT upon completion of the copy if the file names are identical. If the names are not identical, only the temporary file is removed from the AFT.

- PRINT

Under CARDIN, print at the terminal all or any part of a source file or concatenation of source files, reformatting the file by use of format options and/or tab characters, if desired.

1. PRINT

The entire current file is reformatted and printed.

3. RESEX n,m

Line numbers are inserted at the beginning of each line in the current file, regardless of whether or not line numbers already exist. The numbering begins with n and increments by m, or optionally, begins with 10 and increments by 10, if n,m are not specified. If the first character of the existing line is a numeric, a blank is inserted following the generated line number. If the first character of the existing line is not numeric, no blank is inserted.

4. RESE# n,m

Line numbers are inserted at the beginning of each line in the current file, even if line numbers already exist. This numbering begins with n and increments by m, or optionally begins with 10 and increments by 10 if n, m are not specified. If the first character of the existing line is a numeric, a pound sign (#) is inserted following the generated line number. If the first character of the existing line is not numeric, the pound sign is not inserted.

CAUTION: When resequencing, or performing a partial resequence, it is possible to produce files with line numbers out of order. This may be caused by incorrect parameters on partial resequence or when new line numbers exceed eight digits (in non BASIC files). When line numbers are too large, a warning is given. In either case, recovery may be made by resequencing the total file using a smaller beginning line number or a smaller increment.

• ROLLBACK filedescr

The named permanent file is accessed with read and write permission and becomes the input collector file for the user. Any data lines previously collected on the file will be merged with the current file at this time and the first and last such lines of recovered data will be shown to the user.

• #ROLLBACK filedescr

Same as ROLLBACK, but applicable only in Text Editor build mode. Note that the recovered data will be appended to the current file, instead of merged.

• RUN

1. RUN

When a subsystem selection has been previously made, the RUN command can be used to initiate processing on behalf of the subsystem selected.

2. FORTRAN RUN Command

Under the FORT system, the current file is compiled and executed by the time sharing based FORTRAN; under the YFORT system, it is the batch-based FORTRAN. Refer to the FORTRAN manual for further FORTRAN RUN options.

3. RUNH

Executes the selected subsystem and prints a header (date and time) at the top of the program execution report.

- SABB

The Scan Abort (SABB) subsystem allows the user to scan the ABRT file by snapping portions of the file at the terminal. This file is established either by the Time Sharing System or by the user so that the dump of an aborted subsystem may be copied on it. The abort may be caused by an uncorrectable fault in a subsystem or by execution of a DRL ABORT. Refer to the TSS System Programmer's Reference Manual for details of SABB.

- SAVE filedescr-1,permissions,size;  
filedescr-2,permissions,size;...;filedescr-n

The current file is saved on one or more new permanent file(s) defined by filedescr-i. Sorting by line number is or is not done, according to subsystem requirements. The file(s) specified are created with no general permissions or with the permissions specified in the SAVE command. An initial size can be specified in the command by typing in the word SIZE or the letter S followed by the numeric size value. If no size is specified, the subsystem will determine an initial size based on the size of the current file. Size and permissions may be interchanged.

- SCAN filedescr (permissions and altname applicable)

Under CARDIN, the SCAN subsystem -- batch-output scanner -- is initiated to scan the file described by filedescr. The desired functions are defined by the question/answer sequence that follows the use of this command. See the TSS Terminal/Batch Interface manual.

- SEND

Cancel the effect of a previous HOLD command, and causes the last message previously withheld to appear at the terminal. (Refer to the HOLD command.)

- SEQUENCE

Identical to RESEQUENCE for all major systems except BASIC, but operates independently of any system selection that may have been previously made.

- SMCL

A System Master Catalog entry is maintained for each Time Sharing user and includes such information as user ID, logon password, maximum number of llinks permitted for saving files, current number of llinks in use, maximum dollar resources permitted, resources used, and certain permissions assigned by site personnel. This information, except for the logon password, may be obtained by responding with the command "SMCL" at system selection level or while in line-numbered build mode. No parameters are required. The following example illustrates use of the command and the output which is generated:

\*SMCL

```
USER ID-HANSEN
MAX LLINKS-UNLMTD
LLINKS USED-908
MAX RESOURCES-$2000.00
RESOURCES USED-$1484.71
PERMISSIONS-X S T C(35)
```

Permission abbreviations and a brief explanation of each follows:

<u>ABBREV</u>	<u>NAME</u>	<u>MEANING</u>
X	LODX	User is permitted to load and execute bound programs residing on an H* or Q* file.
S	LODS	User is permitted use of the LODS subsystem for debugging resident TSS software.

- T TALK User is permitted use of the CARDIN TALK option.
- C CARDIN User is permitted to run batch jobs via CARDIN. The suffixed parenthesized quantity represents the highest job urgency that is allowed.

- STATUS

- 1. STATUS

- Gives station code followed by a list of the user's status as to processor time used, number of file I/O's, characters output to the terminal and lists the files that are open.

- 2. STATUS FILES

- Lists only the names of the user's open files.

- STRIP

- 1. STRIP

- Removes the line number and any spaces preceding it from each line of the current file. Only the first eight character positions of the line are examined.

- 2. STRIPB

- Strips trailing blanks from existing lines in the current file. If the line number is the only data in the line, STRIPB appends one blank character to that line number.

- 3. STRIPX

- Strips the line numbers from the current file and also the trailing blank if the next character of the existing line is numeric.

- 4. STRIP#

- Strips line numbers from the current file and also the trailing pound sign if the next character of existing line is numeric.

- SYSTEM name

- Cancels any previously made system selection and causes the named selection to qualify the meaning of any subsequent RUN or RESEQUENCE commands. If name is not specified, only the cancellation take place and subsequent RUN or RESEQUENCE commands result in unpredictable situations.

- TAPE

- Builds or extends a current file with input from paper tape. Neither line feeds nor rubouts are supplied by the Time Sharing System. (The command is #TAPE when given in the EDITOR subsystem.)

- #TAPE

- Builds or extends a current file with input from paper tape. Neither line feeds nor rubouts are supplied by the Time Sharing System. This command is only applicable to the EDITOR subsystem.

## SECTION III

### TIME SHARING ERROR MESSAGES EXPLANATION

Error messages generated by the various time sharing subsystems and by the Time Sharing System Executive program fall into two classes (from the viewpoint of explanations):

- Error messages that are considered self-explanatory.
- Error messages that, due to the need for reasonable conciseness in conversational messages, may require further explanation for a given user the first few times that the message is encountered.

All messages falling into the second class are prefixed by a message number, usually enclosed by carets (i.e., `nn`, or in some cases `nn`). Further explanation of these messages is immediately available at the terminal through the HELP subsystem. HELP may be called for either at the subsystem-selection level or at the command level under most major subsystems.

HELP message explanations are listed below, indexed under the associated error message(s). These error messages, in turn, fall into two categories from the viewpoint of origin and applicability.

- Error messages originating from the time sharing Executive, most of which can be received only by an implementor of a new, not fully debugged, time sharing subsystem during its checkout. These messages are numbered 1 through 49.
- Error messages originating from the various time sharing subsystems, which would be received by a user of the system. These messages indicate faulty system usage or system malfunction, and are numbered beginning with 50.

NOTE: On some types of terminals, the carets enclosing the error message number are reproduced as parentheses.

In the following descriptions, generated error messages and their associated HELP subsystem error message explanations are listed by message numbers.



001 - INCORRECT PRIMITIVE

AN ILLEGAL PRIMITIVE HAS OCCURRED IN A COMMAND LIST. CHECK THE COMMAND LIST POINTER IN THE PROGRAM DESCRIPTOR AND THE COMMAND LIST FORMAT AND PRIMITIVES.

002 - location INVALID FILE I/O COMMAND

IN THE CALLING SEQUENCE OF A DRL FOR DISK I/O, THE SEEK, READ or WRITE COMMAND IS INCORRECT. CHECK THE SUBSYSTEM CODE.

003 - location INVALID DCW

IN THE CALLING SEQUENCE OF A DRL FOR DISK I/O, A DCW IS INCORRECT. CHECK THE SUBSYSTEM CODE.

004 - location INVALID DRL ARGUMENT

THE ADDRESS OF A DRL ARGUMENT IS OUTSIDE THE RANGE OF THE PROGRAM. THE NUMBER GIVEN IN THE COMMENT IS THE RETURN FROM THE DERAIL. CHECK THE SUBSYSTEM CODE FOR IMPROPER INITIALIZATION.

005 - BAD DRL CODE

THE ADDRESS OF A DRL CODE IS OUT OF THE RANGE OF USABLE CODES OR ILLEGAL FOR THIS SUBSYSTEM. CHECK THE SUBSYSTEM CODE.

006 - LEVEL OF CONTROL TOO DEEP

THE MAXIMUM NUMBER OF CALLS IN THE PROGRAM STACK OR THE CALLSS STACK HAS BEEN EXCEEDED. IN THE CASE OF THE PROGRAM STACK, THIS MEANS THAT THE SELECTED SYSTEMS PRIMITIVE LIST CONTAINED A CALLP, AND IN TURN, THAT SUBSYSTEMS PRIMITIVE LIST CONTAINED A CALLP, ETC. UNTIL THE LENGTH OF THE PROGRAM STACK WAS EXCEEDED. LIKEWISE, IN THE CASE OF THE CALLSS STACK OF SUBSYSTEMS CALLING OTHER SUBSYSTEMS BY MEANS OF THE DRL CALLSS, THE TABLE LIMIT WAS EXCEEDED. REVIEW THE SUBSYSTEM AND DEPTH OF CALLS.

007 - BAD PROG. DESCRIPTION

IN THE PROGRAM DESCRIPTOR, THE POINTER TO THE COMMAND LIST IS ZERO OR POINTS TO NON-COMMAND LANGUAGE DATA. CHECK THE PROGRAM DESCRIPTOR AND COMMAND LANGUAGE LIST.

008 - LOOP IN PRIMITIVES

A NUMBER OF THE PRIMITIVES ARE EXECUTED ENTIRELY WITHIN THE TSS SCAN MODULE. A COUNTER IS INITIALIZED AT THE ENTRY TO SCAN AND A COUNT KEPT OF PRIMITIVES EXECUTED. WHEN THE COUNT EXCEEDS A GIVEN MAXIMUM, IT BECOMES OBVIOUS THERE IS A LOOP. CHECK THE SEQUENCE OF THE PRIMITIVES FOR THE SUBSYSTEM.

009 - SYSTEM UNKNOWN

THE REQUESTED SUBSYSTEM IS UNKNOWN TO TSS OR IS NOT INCLUDED IN THE SYSTEM FOR THIS INSTALLATION. CHECK THE NAME FOR SPELLING TOO.

010 - PROGRAM TOO LARGE TO SWAP

A SUBSYSTEM IS SO LARGE THAT THE NUMBER OF DCW'S REQUIRED TO LOAD OR SWAP THE PROGRAM EXCEED THE MAXIMUM NUMBER OF DCW'S WHICH CAN BE BUILT. CHECK THE SIZE OF THE SUBSYSTEM. PERHAPS THE SUBSYSTEM EXPANDS ITS CORE LIMITS WITH A DRL ADDMEM. CHECK ALL DRL ADDMEM REQUESTS. SEE .LADCW DEFINED IN COMMUNICATION REGION FOR MAXIMUM NUMBER OF DCW'S ALLOWED.

011 - location INCORRECT CORE FILE USAGE

A REQUEST TO MOVE CORE FILE SPECIFIES MORE THAN TEN WORDS TO BE MOVED. CHECK ALL DRL CORFIL REQUESTS.

012 - location PRIVILEGED I/O ATTEMPTED

PRIVILEGED DISK I/O IS RESERVED FOR SUBSYSTEMS WHICH SPECIFICALLY REQUIRE INFORMATION FROM FILES ALLOCATED TO THE TIME SHARING SYSTEM. PLEASE REVIEW THE NEED FOR PRIVILEGED DISK I/O AND JUSTIFY IT WITH THE COMPUTING CENTER.

013 - location USERID NOT PERMITTED

THE DRL USER ID CAN BE USED ONLY BY THE LOGON SUBSYSTEM. CHECK THE SUBSYSTEM CODE.

014 - NOT CURRENTLY ASSIGNED.

015 - location CANNOT RESET USERID

THE LOGON SUBSYSTEM IS EXECUTING A DRL USER ID, BUT THE ID OF THE SPECIFIED U.S.T. IS NON-ZERO. A TERMINATE MUST BE EXECUTED FOR THAT USER BEFORE THE U.S.T. CAN BE REUSED. TRY TO DETERMINE WHY THE TERMINATE WAS BYPASSED, OR WHY NEW SYSTEM WAS SELECTED AFTER LOGON.

016 - location OVERFLOW FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED AN OVERFLOW CONDITION AT THE DESIGNATED LOCATION AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR. THE LOCATION IS RELATIVE TO ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM. THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM. REVIEW YOUR PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

017 - location ILLEGAL OP CODE

THE SUBSYSTEM IN EXECUTION ENCOUNTERED AN ILLEGAL (OR ZERO) OP CODE OR A MME OPERATION AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW YOUR PROGRAM CODE AND INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

018 - location MEMORY FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED A MEMORY FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW THE PROGRAM CODE AND INITIALIZATION OF ADDRESS OR INDEX REGISTERS AS WELL AS THE PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

019 - location FAULT TAG FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED A FAULT TAG FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW THE PROGRAM CODE AND INITIALIZATION OF ADDRESS OR INDEX REGISTERS AS WELL AS THE PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

020 - location DIVIDE CHECK FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED A DIVIDE CHECK FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW YOUR PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

021 - (nnnnnn) BAD STATUS SWAP OUT #S

A BAD I/O STATUS HAS BEEN RECEIVED ON A WRITE DRUM FILE #S, THE SWAP FILE. TRY AGAIN. IF PROBLEM PERSISTS, ALERT OPERATIONS. THE PARENTHEZIZED NUMBER IS THE STATUS CODE.

022 - (nnnnnn) BAD STATUS SWAP IN #S

A BAD I/O STATUS HAS BEEN RECEIVED ON A READ DRUM FILE #S, THE SWAP FILE. TRY AGAIN. IF PROBLEM PERSISTS, ALERT OPERATIONS. THE PARENTHEZIZED NUMBER IS THE STATUS CODE.

023 - (nnnnnn) BAD STATUS LOAD #P

A BAD I/O STATUS HAS BEEN RECEIVED ON A READ DRUM FILE #P, THE TSS FILE. TRY AGAIN. IF PROBLEM PERSISTS, ALERT OPERATIONS. THE PARENTHEZIZED NUMBER IS THE STATUS CODE.

024 - location MME FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED A MME FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW THE PROGRAM CODE AND INITIALIZATION OF ADDRESS OR INDEX REGISTERS AS WELL AS THE PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

025 - location LOCKUP FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED A LOCKUP FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW THE PROGRAM CODE AND INITIALIZATION OF ADDRESS OR INDEX REGISTERS AS WELL AS THE PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

026 - location OP-NOT-COMPLETE FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED AN OP-NOT-COMPLETE FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW THE PROGRAM CODE AND INITIALIZATION OF ADDRESS OR INDEX REGISTERS AS WELL AS THE PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

027 - location COMMAND FAULT

THE SUBSYSTEM IN EXECUTION ENCOUNTERED A COMMAND FAULT AT THE DESIGNATED LOCATION, AND THE SUBSYSTEM DID NOT SPECIFY A FAULT VECTOR.

THE LOCATION IS RELATIVE TO SUBSYSTEM ZERO (SEE EDIT MAP) UNLESS IT IS A MASTER SUBSYSTEM, THEN THE LOCATION IS RELATIVE TO TSS ZERO AND ONE MUST DETERMINE THE LOAD ADDRESS OF THE SUBSYSTEM TO DETERMINE THE FAULT LOCATION IN THE MASTER SUBSYSTEM.

REVIEW THE PROGRAM CODE AND INITIALIZATION OF ADDRESS OR INDEX REGISTERS AS WELL AS THE PROGRAM INPUT FOR INCORRECT DATA BEFORE REQUESTING HELP FROM THE COMPUTING CENTER.

028 - location REWIND ATTEMPTED FOR RANDOM FILE - filename

A RANDOM FILE CANNOT BE SPACED IN THIS MANNER. USAGE OF THE RANDOM FILE IN THE CORRECT MANNER WILL CLEAR UP THE PROBLEM.

029 - ILLEGAL SYSTEM SELECTION

SOME SYSTEMS, NAMELY THE MASTER SUBSYSTEMS, HAVE RESTRICTED THEIR AVAILABILITY TO CERTAIN USERS. YOU DO NOT HAVE PERMISSION TO USE THE SELECTED SUBSYSTEM. SELECT ANOTHER.

30-49 - NOT CURRENTLY ASSIGNED.

<50> FILE filename -- reason text

<50< FILE filename -- reason text

(The two messages above refer to permanent files.)

<50> CURRENT FILE -- reason text

<50< COLLECTOR FILE -- reason text

(The two messages above refer to the temporary files \*SRC and SY\*\*, respectively.)

<50> WORK FILE -- reason-text

(The message above refers to all other temporary files.)

ERROR-MESSAGE 50 EXPLANATION: FILE-SYSTEM ERRORS.

THIS MESSAGE IS ISSUED FOR EITHER ONE OF TWO CASES: (1) THE NAMED PERMANENT FILE COULD NOT BE ACCESSED-- <50>, OR COULD NOT BE CREATED--<50< OR (2) A REQUIRED TEMPORARY FILE COULD NOT BE OBTAINED OR EXPANDED. REPLY TO THE QUESTION "GROUP?" AS FOLLOWS FOR A FURTHER EXPLANATION: IF YOUR MESSAGE STATES "NO PERMISSION, NONEXISTENT FILE" OR "INVALID PASSWORD," REPLY "1". IF "FILE BUSY, NO FILE SPACE" OR "ILLEGAL CHAR.," REPLY "2". IF "I/O ERROR, FILE TABLE FULL, DUPLICATE NAME" OR "SYSTEM LOADED," REPLY "3". IF IT STATES "STATUS NN" REPLY "4".

STATUS 01: THE SPECIFIED USER'S-MASTER-CATALOG DOES NOT EXIST. CHECK USER-ID.

STATUS 02: I/O ERROR. THE FILE SYSTEM HAS ENCOUNTERED AN UNRECOVERABLE INTERNAL I/O ERROR. (THIS DOES NOT IMPLY AN ERROR ON YOUR FILE SPACE.) REPORT THE STATUS TO THE CENTRAL COMPUTER SITE. ALSO RETRY.

STATUS 03: PERMISSION DENIED. THE NAMED FILE COULD NOT BE ACCESSED BECAUSE YOU HAVE NOT BEEN ALLOWED THE PERMISSION(S) REQUESTED. IF THE FILE IS ALREADY OPEN, THE PERMISSIONS REQUESTED DO NOT MATCH THE PERMISSIONS WITH WHICH THE FILE IS ALREADY OPENED. THIS STATUS IS ALSO RETURNED BY THE FILE SYSTEM WHEN AN ATTEMPT IS MADE TO OPEN A "NULL" FILE WITH "READ" PERMISSION ONLY.

STATUS 04: FILE BUSY. ANOTHER USER HAS ALREADY ACCESSED THIS FILE WITH AN ACCESS-MODE PERMISSION THAT LOGICALLY EXCLUDES YOUR REQUESTED PERMISSION; I.E., A GRANTED WRITE PERMISSION EXCLUDES ANY OTHER CONCURRENT ACCESSES AND A GRANTED READ PERMISSION EXCLUDES ANY OTHER ACCESS WITH WRITE PERMISSION. THE FILE, THEREFORE, IS TEMPORARILY BUSY TO SOME OR ALL OTHER USERS. (MULTIPLE CONCURRENT ACCESSES OF A FILE WITH READ PERMISSION, ONLY, IS ALLOWED.)

STATUS 05: NONEXISTENT FILE. EITHER THE NAMED FILE DOES NOT EXIST, AT THE CATALOG LEVEL IMPLIED OR SPECIFIED, OR ONE OR MORE NAMES IN THE CATALOG/FILE DESCRIPTION WAS INCORRECTLY GIVEN. CHECK ALL CATALOG/FILE NAMES. THE COMMAND CATALOG MAY BE USED TO LIST ALL OF YOUR CATALOG AND FILE NAMES.

STATUS 06: THE FILE SYSTEM HAS EXHAUSTED ITS SPACE FOR NEW CATALOGS AND FILE DESCRIPTORS. REPORT THE STATUS TO THE CENTRAL COMPUTER SITE, AND TRY AGAIN LATER.

STATUS 07: DEVICE TYPE UNDEFINED. THE DEVICE TYPE THAT YOU SPECIFIED FOR YOUR FILE IS UNDEFINED TO THE SYSTEM.

STATUS 10: THE SYSTEM HAS TEMPORARILY EXHAUSTED THE AVAILABLE FILE SPACE. TRY AGAIN LATER. (ALSO, PURGE ANY UNNEEDED FILES.)

STATUS 11: NON-UNIQUE NAME. THE NEW NAME THAT YOU HAVE SPECIFIED FOR THE CATALOG OR FILE TO BE MODIFIED IS A DUPLICATE OF A CATALOG OR FILE NAME EXISTING AT THE SAME LEVEL.

STATUS 12: MAX. SIZE ERROR. THE NEW MAXIMUM-SIZE SPECIFIED FOR THE FILE TO BE MODIFIED IS LESS THAN ITS CURRENT SIZE. (MAXIMUM SIZE UNCHANGED.)

STATUS 13: NO FILE SPACE. YOU HAVE USED UP ALL THE PHYSICAL SPACE ALLOTTED TO YOU FOR THE CREATION OF FILES. YOU MUST EITHER PURGE ONE OR MORE UNNEEDED FILES, OR OBTAIN A LARGER FILE-SPACE ALLOCATION.

STATUS 14: INVALID PASSWORD. A REQUIRED PASSWORD EITHER HAS BEEN GIVEN INCORRECTLY OR NOT AT ALL. THE GENERAL FORM FOR SUPPLYING PASSWORDS IN A CATALOG/FILE DESCRIPTION IS: NAME\$PASSWORD E.G.: /CAT1\$ABC/FIL1\$XYZ.

STATUS 15: FILE IS ABORT LOCKED.

STATUS 16: FILE WRITE IN BATCH ONLY.

STATUS 17: SEEK ERROR.

STATUS 20: FAILURE IN NAME SCAN.

STATUS 21: UNDEFINED DEVICE.

STATUS 22: DEVICE LINK TABLE CHECKSUM ERROR.

STATUS 23: INCONSISTENT FSW BLOCK COUNT.

STATUS 24: INTERNAL LINK TABLE CHECKSUM ERROR.

STATUS 25: REQUESTED ENTRY NOT ON LINE.

STATUS 26: NON-STRUCTURED FILE ENTRY.

STATUS 27: FILE IN DEFECTIVE STATUS.

STATUS 30: ILLEGAL PACK TYPE.

STATUS 31: ACCESS GRANTED TO IDS FILE.

STATUS 32: COLLECTION FILE ERROR.

STATUS 33: CATALOG/FILE SECURITY LOCKED

STATUS 34: ILLEGAL CHAR. YOU HAVE GIVEN A CATALOG OR FILE NAME, OR A PASSWORD, CONTAINING A CHARACTER OTHER THAN AN ALPHANUMERIC, PERIOD, OR A DASH, WHICH ARE THE ONLY LEGAL CHARS. FOR IDENTIFIERS.

STATUS 35: PERMISSION NOT GRANTED TO LIST OR PURGE REQUESTED CATALOG.

STATUS 36: FILE TABLE FULL. THE NAMED FILE CANNOT BE ACCESSED BECAUSE YOU PRESENTLY HAVE TOO MANY FILES ALREADY ACCESSED (I.E., OPENED). YOU MUST DEACCESS ONE OR MORE OF THESE OPENED FILES. USE THE COMMANDS STATUS FILES, AND REMOVE.

STATUS 37: DUPLICATE NAME. THE FILE NAME SHOWN DUPLICATE TO A NAME ALREADY IN YOUR AVAILABLE-FILE-TABLE, I.E., AN ALREADY ACCESSED FILE. IF APPROPRIATE, ASSIGN AN ALTERNATE NAME.

STATUS 40: SYSTEM LOADED. THE SYSTEM IS CURRENTLY AT PEAK CAPACITY IN SOME RESPECT, E.G.: CERTAIN INTERNAL TABLE SPACE EXHAUSTED, ETC.

STATUS 41: NO PROTECTION TABLE SPACE AVAIL.

STATUS 42: INVALID FILE CODE OR PAT POINTER.

STATUS 43: INVALID CATALOG BLOCK ADDRESS.

STATUS 44: PERMISSION DENIED - SHARED FILE.

STATUS 45: INVALID SPACE IDENTIFIER.

STATUS 46: CATALOGS BUSY.

STATUS 47 AND 50: SYSTEM MALFUNCTION. REPORT THE STATUS TO THE CENTRAL COMPUTER SITE, AND RETRY.

STATUS 51: CHECKSUM ERROR ON DEVICE.

STATUS 52: DEVICE RELEASED.

STATUS 53: NOT CURRENTLY ASSIGNED.

STATUS 54: NOT CURRENTLY ASSIGNED.

STATUS 55: NOT CURRENTLY ASSIGNED.

STATUS 56: SECURITY PARAMETER - REQUIRED.

STATUS 57: SECURITY PARAMETER - INVALID.

STATUS 60: SITE USED STATUS.

STATUS 61: \$FSYS HAS BEEN ENABLED.

STATUS 62: ILLEGAL SUBFUNCTION CODE.

STATUS 63: FILE NOT BEING MONITORED.

STATUS 64: DEADLOCK ON PAGE REQUEST.

STATUS 65: PAGE CURRENTLY BUSY.

STATUS 66: FILE NOT DUPLICATED.

STATUS 67: TDS MONITOR ALLOC ERROR.  
STATUS 70: ILLEGAL CHECKPOINT REQUEST.  
STATUS 71: ILLEGAL DCW SPECIFIED.  
STATUS 72: IMPROPER PROTECTION OPTION.  
STATUS 73: INVALID ARGLIST PARAMETER NUMBER.  
STATUS 74: SYSTEM JOURNAL NOT CONFIGURED.  
STATUS 75: FILE RESTORE LOCKED.  
STATUS 76: FILE TDS LOCKED.  
STATUS 77: ERR TDS SUBSET PAGES RELEASE.

<51> FILE filename -- I/O STATUS yy

<51< FILE filename -- I/O STATUS yy

(The two messages above refer to permanent files.)

<51> CURRENT FILE -- I/O STATUS yy

<51< CURRENT FILE -- I/O STATUS yy

(The two messages above refer to the \*SRC file.)

<51> COLLECTOR FILE -- I/O STATUS yy

<51< COLLECTOR FILE -- I/O STATUS yy

(The two messages above refer to the SY\*\* file.)

<51> WORK FILE -- I/O STATUS yy

<51< WORK FILE -- I/O STATUS yy

(The two messages above refer to all other temporary files.)

where yy is the major hardware status returned by IOS. These status codes are described in the General Comprehensive Operating Supervisor reference manual.

#### ERROR-MESSAGE 51 EXPLANATION: INPUT/OUTPUT ERRORS

AN UNRECOVERABLE READ OR WRITE ERROR HAS OCCURRED ON THE SPECIFIED FILE. AN ERROR IN READING IS INDICATED BY THE MESSAGE NUMBER GIVEN AS <51>; AN ERROR IN WRITING AS <51<. REPORT THE I/O STATUS NUMBER AND THE READ OR WRITE INDICATION TO THE CENTRAL COMPUTER SITE. ALSO, IN THE CASE OF "CURRENT FILE" or "WORK FILE", LOG OFF AND TRY AGAIN.

<52> CURRENT FILE NOT DEFINED

#### ERROR-MESSAGE 52 EXPLANATION

THERE IS NO CURRENT (\*SRC) FILE DEFINED IN YOUR FILE TABLE. THIS INDICATES EITHER A SYSTEM MALFUNCTION, OR THAT YOU ARRIVED AT THE PRESENT SUBSYSTEM VIA AN ABNORMAL PATH. SUGGEST YOU RESELECT YOUR DESIRED SUBSYSTEM, OR LOG OFF AND RETRY FROM SCRATCH.



<53> LINES IGNORED BY EDIT  
....line(s).....

ERROR-MESSAGE 53 EXPLANATION

THE LINE(S) SHOWN WERE NOT MERGED INTO YOUR CURRENT FILE BECAUSE THEY LACKED LINE NUMBERS.

<54> SYSTEM MALFUNCTION--CURRENT FILE ERROR

ERROR-MESSAGE 54 EXPLANATION

THE FORMAT OF YOUR CURRENT FILE WAS FOUND TO BE IN ERROR. REPORT CIRCUMSTANCES TO THE CENTRAL COMPUTER SITE. SUGGEST THAT YOU LOG OFF AND RETRY.

<55> CURRENT FILE TOO LARGE

ERROR-MESSAGE 55 EXPLANATION

THE COMBINED SIZE OF YOUR SOURCE FILE AND MOST RECENT MODIFICATION- OR ADDITION-INPUT IS TOO LARGE TO BE PROCESSED. SUGGEST THAT YOU SPLIT THE TEXT INTO TWO OR MORE FILES, WHICH CAN LATER BE ADJOINED.

<56> NOT CURRENTLY ASSIGNED

057 - RESTRICTED SUBSYSTEM

THE CENTRAL COMPUTER SITE HAS RESTRICTED THE USE OF THIS SYSTEM. THIS MAY BE A TEMPORARY RESTRICTION BECAUSE OF CURRENT LOAD OR A PERMANENT RESTRICTION. PLEASE NOTIFY THE CENTRAL COMPUTER SITE FOR FURTHER DETAILS.

<58> ENTRY LOC < 100

ERROR-MESSAGE 58 EXPLANATION

THE SUBSYSTEM PROGRAM TO BE EXECUTED DOES NOT HAVE THE INITIAL 100-WORD DATA AREA THAT IS REQUIRED OF TSS SUBSYSTEM PROGRAMS.

<59> FILE filename NOT IN TSS FORMAT

ERROR-MESSAGE 59 EXPLANATION

A FORMAT ERROR WAS DETECTED ON THE NAMED FILE. EITHER THE FILE IS NOT A TSS-GENERATED FILE, OR A SYSTEM MALFUNCTION HAS OCCURRED. IN THE LATTER CASE, REPORT THE CIRCUMSTANCES TO THE CENTRAL COMPUTER SITE, AND RETRY THE COMMAND.

<60> NO DATA ON FILE filename

ERROR-MESSAGE 60 EXPLANATION

THE REQUESTED FILE CONTAINS NO USER'S DATA; THE IMPLICATION IS THAT NO DATA HAS BEEN SAVED ON THIS FILE SINCE ITS CREATION.

061-063 - NOT CURRENTLY ASSIGNED.

<064> - EXECUTE TIME LIMIT EXCEEDED

THE PROGRAM TIME LIMIT SPECIFIED BY THE USER AND/OR THE INSTALLATION HAS BEEN EXCEEDED BY THE OBJECT PROGRAM.

<065> - OBJECT PROGRAM SIZE LIMIT EXCEEDED

THE SIZE OF THE OBJECT PROGRAM HAS EXCEEDED THE INSTALLATION SPECIFIED LIMIT.

<066> - SPAWN UNSUCCESSFUL -- STATUS N

A SUBSYSTEM WAS UNABLE TO SPAWN A JOB TO BATCH FOR COMPILATION AND/OR LOADING. THE REASON CODE "N" DESCRIBES ONE OF THE FOLLOWING:

- 1 - UNDEFINED FILE (FILE NOT IN APT)
- 2 - NO SNUMB
- 3 - DUPLICATE SNUMB
- 4 - NO PROGRAM NUMBER AVAILABLE
- 5 - ACTIVITY NAME UNDEFINED
- 6 - ILLEGAL USER LIMITS (TIME, SIZE, ETC.)
- 7 - BAD STATUS (R/W J\*)
- 8 - NO FILE SPACE AVAILABLE FOR PUSH-DOWN FILE
- 9 - NO \*J FILE PROVIDED

<067> - (Error message text)

THE ERROR DESCRIBED IN THE MESSAGE HAS BEEN DETECTED BY TSS WHILE IN COMMAND FILE OR DEFERRED PROCESSING MODE. THE MODE HAS BEEN DISCONTINUED DUE TO ITS OCCURRENCE.

NOTE: See Appendix A for a listing of the comment number that appears in the error message text.

068-133 - NOT CURRENTLY ASSIGNED.

134 - location INVALID DRL FILACT FUNCTION

THE DRL FILACT CALLING SEQUENCE CONTAINS AN INVALID FUNCTION NUMBER.

135 - location PRIVILEGED DRL FILACT REQUEST

USER CANNOT ACCESS THE SYSTEM MASTER CATALOG.

136 - NOT CURRENTLY ASSIGNED.

137 - NOT CURRENTLY ASSIGNED.

138 - location NO TAP\* FILE FOR DRL TAPEIN

THE TAP\* FILE IS UNDEFINED FOR TAPE INPUT.

139 - ERROR IN WRITING TAP\* FILE

AN ERROR OCCURRED WHILE WRITING IN THE TAP\* FILE.

## SECTION IV

### TERMINAL USAGE

#### GENERAL

This section contains general descriptions of and operational procedures for several remote terminals. For complete details pertaining to a particular terminal, the user should refer to the instruction manual accompanying the terminal unit.

#### TELEPRINTER OPERATION

##### Terminal Applications

The following types of teleprinter terminals, or their equivalent, may be used to communicate with the Time Sharing System:

- IBM 2741
- Teletype Models 33, 35 and 37
- GE TermiNet 300

These terminals communicate with the Time Sharing System via the Remote Terminal Supervisor (GRTS) or the Network Processing Supervisor (NPS). These interfaces are described in the Remote Terminal Supervisor (GRTS) manual and the Network Processing Supervisor (NPS) manual.

Each time a key is struck, the character is transmitted to the system and stored until the carriage return is struck. A carriage return indicates that the line is complete. The number of characters in a line may range up to 160 characters plus a carriage return.

If the terminal is equipped with a paper tape reader/punch, this device may be used for input/output. The input must be formatted the same as for keyboard input, but each line must be terminated with carriage return, line feed, rubout, rubout. The input tape must be terminated with an ASCII X-OFF character.

## Editing

Keyboard input is sent to the computer in units of complete lines. A line of terminal input is terminated by a carriage return. Therefore, corrections to a line-in-progress (i.e., a partial line not yet terminated) can be made.

A typing error detected before the line is terminated can be corrected in one of two ways. One or more characters may be deleted from the end of a partial line or the incomplete line and may be canceled. Character or line deletions are effected by means of two special characters designated as control characters. These control characters may differ between terminals.

### For teleprinter terminals

<u>character</u>	<u>control function</u>
@(commercial at sign)	character deletion
CTRL plus X keys	line deletion

### For IBM 2741 or DATEL terminals

<u>character</u>	<u>control function</u>
1/4 (or degree symbol)	character deletion
<u>+</u>	line deletion

NOTE: Line deletion does not occur until a carriage return is given or ATTN (IBM 2741) or INT (DATEL) is pressed.

The editing rules are as follows:

- Use of the character-delete control deletes from the line the character preceding the deletion character; use of  $n$  consecutive deletion characters deletes  $n$  preceding characters (including blanks) up to the beginning of the line.

For example:

\*ABCDFE would result in ABCDE being transmitted to the program file.

\*ABCDEF<sup>oooo</sup> DEF would result in ABCDEF being transmitted.

(The characters to be deleted are underlined for illustration.)

- Use of the line-delete control causes all of a line to be deleted. The characters DEL are printed to indicate deletion. For example:

```
*ACDEFG CTRL/X DEL      (all characters deleted;  
                          carriage return automatic)  
  
                          -(ready for new input)
```

or

```
*ACDEFG+ (carriage return)  
  
DEL          (all characters deleted)  
  
              -(ready for new input)
```

NOTE: CTRL/X, ATTN, or INT do not require a carriage return.

The control-character pair for each type of terminal cannot be used for other than the deletion function assigned them.

In AUTOX and AUTO, line numbers and spaces are not deleted.

### Log-On Procedure

To initiate communication with the Time Sharing System, the user performs the following steps:

- Turns on the terminal
- Obtains a dial-tone on the associated phone-set
- Dials one of the numbers of his time sharing center

The user will then receive either a busy signal to indicate that the line is not presently available or a high-pitched tone -- a "beep" -- to indicate that his terminal has been connected to the computer.

The Time Sharing System is then prepared to output a log-on message; either automatically (no terminal action required) or following a carriage return from the terminal. The following is a sample of the automatic log-on message:

```
HIS SERIES 6000 ON date AT time CHANNEL nnnn
```

where time is given in hours and thousandths of hours (hh.hhh), and nnnn is the user's channel number. This is the standard message, however the user site may put in a message of its own.

The following is a sample of the log-on message when a carriage return is required:

```
110601  
HIS SERIES 6000 ON date AT time CHANNEL nnnn
```

The number "110601" identifies the type of channel to which the terminal is connected. For a detailed explanation of the meaning of this number, refer to the Remote Terminal Supervisor (GRTS) manual or the Network Processing Supervisor (NPS) manual.

Following this message, the system asks for the user's identification:

USER ID -

The user responds, on the same line, with the user-ID assigned to him by the time sharing installation management. This user-ID uniquely identifies a particular user already known to the system. This ID is used to locate his programs and files, and accounting for his usage of the time sharing resources allocated to him. An example request and response might be:

USER ID -J.P.JONES

NOTE: User's responses are underlined for illustrative purposes.

A carriage return must be given following any complete response, command, or line of information typed by the user. If a charge number is also required for accounting purposes, the user can supply it as follows:

USER ID -J.P.JONES;1234567E

The charge number may consist of from 1 to 12 alphanumeric characters, separated from the user-ID by a semicolon. (Refer to NEWUSER command description in Section II.)

After the user responds with his user-ID, the system asks for the sign-on password that was assigned to him along with his user-ID as follows:

PASSWORD--  
XXXXXXXXXXXX

The user should type his password directly on the "strikeover" mask provided below the PASSWORD request. The password is used by the system as a check on the legitimacy of the named user. (In the event that either the user-ID or password is given incorrectly two consecutive times, the user's terminal is immediately disconnected from the system.)

On teletype-compatible devices, after the password is entered on the strikeover mask, a random alphabetic character string is typed over the password entry. The user's password is thus "sandwiched" between strikeovers for hard copy devices or totally overwritten for screen displayed devices.

On Visual Information Projection (VIP) devices, upon receipt of the password, a Reverse Line Feed (RLF) character, followed by a string of spaces, is issued to erase the entry from the screen.

User-ID and password may be given on the same line when the query "USER ID-" is issued, separated from one another by a dollar sign (\$). Charge number, when specified, must follow the password and be separated from the latter with a semicolon. Note that security is compromised by entering the password in this manner, since it is not typed on a strikeover mask. Assuming the password of user J.P. Jones is "JPJ", the following example illustrates this method of logon:

USER ID - J.P.JONES\$JPJ;1234567E

At this point, if the accumulated charges for the user's past time sharing usage equals or exceeds one hundred percent of his current resource allocation, he will receive a warning message:

RESOURCES OVERDRAWN n%

If his accumulated charges exceeds one hundred and ten percent of his current resources, he receives the following message and is immediately disconnected.

RESOURCES EXHAUSTED - CANNOT ACCEPT YOU

If the user's file space is greater than 88 percent used, he receives the following information message:

n BLOCKS FILE SPACE AVAILABLE

The number n specifies the number of 320-word blocks of unused file space for this user. This does not affect the log-on procedure, and the user is permitted to continue.

After the user has been validated, an asterisk is issued indicating readiness to accept commands and/or build files. The RUN and RESEQUENCE commands are unacceptable at this point since it is not known what type of source is to be acted upon. For example, does RUN mean to compile a BASIC or FORTRAN program, or does it mean a batch job is to be submitted for processing? The user has two alternatives available to deal with this ambiguity:

1. A mode can be established by simply entering the desired system selection, optionally accompanied by an OLD or NEW request. Permissible system selections are as follows:

BASIC  
FORTRAN  
YFORTRAN  
ALGOL  
JOVIAL  
CARDIN  
DATABASIC

Once a system selection has been made, the system remains in effect until explicitly changed (or cancelled by means of the break key). The RUN and RESEQUENCE commands can be used once the mode is established.

2. The BRN, FRN, and JRN commands can be issued independent of previous system selection (if any) and imply RUN for BASIC, FORTRAN, and CARDIN, respectively. Note that JRN cannot be used as an execution command for a CARDIN program unless that program follows CONVERT subsystem syntax. The command BSEQUENCE can be used to resequence a BASIC file and SEQUENCE can be used to resequence a non-BASIC file, independent of the current system selection.



Note that the JRN command is not identical to the CARDIN RUN command. Refer to the TSS Terminal/Batch Interface manual for details concerning the use of JRN in conjunction with CONVERT subsystem.

NOTE: A carriage return terminating each separate user response is assumed to be understood. The underlining indicates the user's response.

The following is an example of a complete log-on procedure, up to the point where the user is ready to begin file building or exercising commands:

HIS SERIES 6000 ON 05/26/77 AT 14.568 CHANNEL 0012

USER ID -J.P.JONES

PASSWORD

XXXXXXXXXX

\* - (the user begins entering input on this line)

## Entering Build Mode Input

Following the log-on procedure, the user is in build mode (as indicated by the initial asterisk) and is ready to build files and/or exercise commands. All lines of input other than commands are accumulated on the user's current file. This is normally the file that contains the program or text he wants to work with. If he is building a new file, his current file will initially be empty.

If he has recalled an old file (OLD filename) the content of the named old file will initially be on his current file. Any input (except control commands) will either be added to, merged with, or replace lines in the current file, depending upon the relative line numbering of the lines in the file and the new input. (Refer to "Correction or Modification of Line-Numbered Files" below).

Following each line of input (that is not a command) and terminating carriage return, the subsystem supplies an initial asterisk, indicating that it is ready to accept more input. In the case of command language input, the user is normally returned to build mode following execution of the process requested by the command.

A line of file building input must begin with a line number contained within the first eight character positions of the line. This number may optionally be preceded by one or more initial blanks. The line number facilitates correction and modification of the source program. The line number is always terminated, (i.e., immediately followed) by a non-numeric character, which may be a blank.

\*

## Correction or Modification of Line-Numbered Files

The correction or modification of the current file in line number sequence proceeds according to the following rules:

- Replacement: a numbered line will replace any identically numbered line that was previously typed or already contained on the current file; i.e., the last entered line numbered nnn will be the only line numbered nnn in the file.
- Deletion: a line consisting of a line number only, (i.e., nnn), will cause the deletion of any identically numbered line that was previously typed or is already contained on the current file.
- Insertion: a line with a line number value that falls between the line number values of two existing lines will be inserted in the file between those two lines.

At any point in the process of entering file building input in line-numbered subsystems, the LIST command may be given, which results in a clean, up-to-date copy of the current file being printed. In this way, the results of any previous corrections or modifications can be verified visually. (The several forms of the LIST command are described in detail in Section II.) Following the command OLD filename, the LIST command can be used initially to inspect the contents of the current source file; i.e., the "old" program.

### Automatic Terminal Disconnections

Once communication with the Time Sharing System has been established, any question or request must be answered within ten minutes. If these time limits are exceeded, the user's terminal will be disconnected.

### Log-Off Procedure

To terminate the user's current session with the Time Sharing System and disconnect the terminal, the BYE or LOGOFF command may be given.

\*BYE

or

\*LOGOFF

A report of the user's time sharing usage charges is given, as illustrated by the following example, and the terminal is disconnected:

```
**COST: $ 0.17 TO DATE: $ 206.11=21%  
**ON AT 15.000 - OFF AT 15.016 ON 04/19/77
```

If the BYE command is used, prior to the issuance of the user's usage charges, the AFT is scanned for the user's temporary files and the user is queried as to their disposition if such files exist.

To terminate the current session without disconnecting the terminal, the command NEWUSER may be given in place of BYE. This procedure allows another user to log-on immediately following. The current user's log-off report is then printed and a new log-on sequence is initiated. NEWUSER may also be used to change the charge number, but without going through the log-off/log-on procedure.

CAUTION: Failure to follow log-off procedures as described above may result in unpredictable problems (lines or files remaining busy, etc.). Certain data sets do not automatically disconnect after log-off from the terminal. In such cases, it is necessary to manually disconnect the data set by lifting the handset, pressing the talk button, and hanging up the handset when the dial tone is heard.

## Terminating an Output Process

A lengthy listing or other output of information at the terminal, initiated for example by a LIST command, may be prematurely terminated by the use of the interrupt control peculiar to the type of terminal in use. This interrupt control is as follows:

- For teleprinter terminals -- the BREAK key
- For typewriter-like terminals -- the ATTN or INT key

This control can also be used for abnormal termination of a program execution. However, the user is cautioned against indiscriminate use of this control since the results of its use are in some cases unpredictable (in regard to the status of files, for example). The subsystem will normally return to build mode or to the subsystem selection level following the use of an interrupt control.

## Paper Tape Input in Build Mode

In order to supply file-building input from paper tape, the user gives the command TAPE (#TAP if the subsystem is Text Editor). The subsystem responds with READY. If the tape reader is ready, it will be turned on automatically. If it is not ready, the user should position his tape in the tape reader and start the device. Input is terminated when an X-OFF character is read by the paper tape reader, or the tape is stopped and the user types X-OFF.

The tape may be prepared off-line from the keyboard, or it may be the result of previous output punched by the paper tape unit. If prepared off-line, it should include carriage returns to terminate each line, just as if entering data on-line, plus explicit line feeds to obtain legibility on the terminal printer during preparation and transmission. The carriage return and line feed must be followed by two rubout characters for terminal timing considerations.

Command language may not be included on the tape. The input should be preceded by several rubout characters and terminated by an X-OFF followed by several rubout characters. Neither the X-OFF nor the rubout characters will appear in the file.

As with keyboard input, a maximum of 160 characters is permitted per line of paper tape input. Excessive lines will be truncated at 160 characters, with the remaining data placed in the next line. A maximum of two disk links (7680 words) of paper tape input will be collected during a single input procedure, except in LUCID mode, which has a limit of six links. All data in excess of two disk links will be lost.

## Building File from Non-ASCII Paper Tape

In order to supply file building input from non-ASCII paper tape (unaltered eight-bit codes), the user gives the command LUCID instead of TAPE. The system will read in the tape and store the data on a file without editing or parity modifications. The system does not delete or act on any characters in the data stream, such as DEL, X-OFF, CR, etc. The input will be terminated when a pause of over one second occurs in the data transmission. Termination does not require an X-OFF character, as does normal paper tape input via a Front-End Network Processor.

NOTE: LUCID cannot be used if data communication is via a Low Speed Line Adapter (LSLA) or an Asynchronous Communication Base (ACB) on a DATANET 355/6600 Front-End Network Processor.

During paper tape input via a Front-End Network Processor, the paper tape input will stop when an error message is to be sent to the terminal.

### Automatic Paper Tape Input

At any point during the operation of the Time Sharing System and at a time when the user must supply keyboard input, a previously prepared paper tape in special format may be used to simulate a sequence of responses, one line at a time. The system need not be in build mode and direct (i.e., conversational) responses, file building input, and/or commands may be entered.

This feature allows the preparation of a paper tape for input to the Time Sharing System and/or subsystem(s) prior to connection with the system and allows terminal operation without supervision during the connection. The paper tape input may be for a specific subsystem or production program execution only, or may include anything from log-on through log-off procedures. Obviously such a tape must be error free.

The required format for each input line is as follows:

- data string (up to 80 characters)
- carriage return
- X-OFF
- RUBOUT (may be multiple, but one is minimum requirement)

Character-delete control characters may be used. Line-delete controls must be used as follows:

- data string (to be deleted)
- (line-delete control) character
- X-OFF
- RUBOUT (one is minimum)
- corrected data string
- carriage return
- X-OFF
- RUBOUT

NOTE: Parity errors encountered during paper tape input may cause the terminal to be disconnected.

It is suggested that extraneous line feeds not be included in the tape. If, however, the user desires line feeds for terminal printer legibility, they should be either between the data string and carriage return, or one line feed immediately following X-OFF.

To initiate automatic paper tape input, the user should position the tape and start the reader at any time that keyboard input is required.

The terminal is automatically disconnected if no input is received within 10 minutes of the request for such input, whether via paper tape or keyboard.

#### KEYBOARD/DISPLAY TERMINAL OPERATION

The keyboard/display terminals are cathode-ray tube display devices which are similar in operation to the teleprinter terminals. This section describes operation of some of the types of display devices commonly used with the Time Sharing system, these are:

- DATANET 760 VIP (Visual Information Projection)
- 765/775/785/7700 Series VIP

The keyboard for the 775/785 Series VIP is shown in Figure 4-1. Most of the display devices have a similar keyboard. Some of the keys and their function are discussed here, but for a complete description, the user should refer to the manual for the specific device.

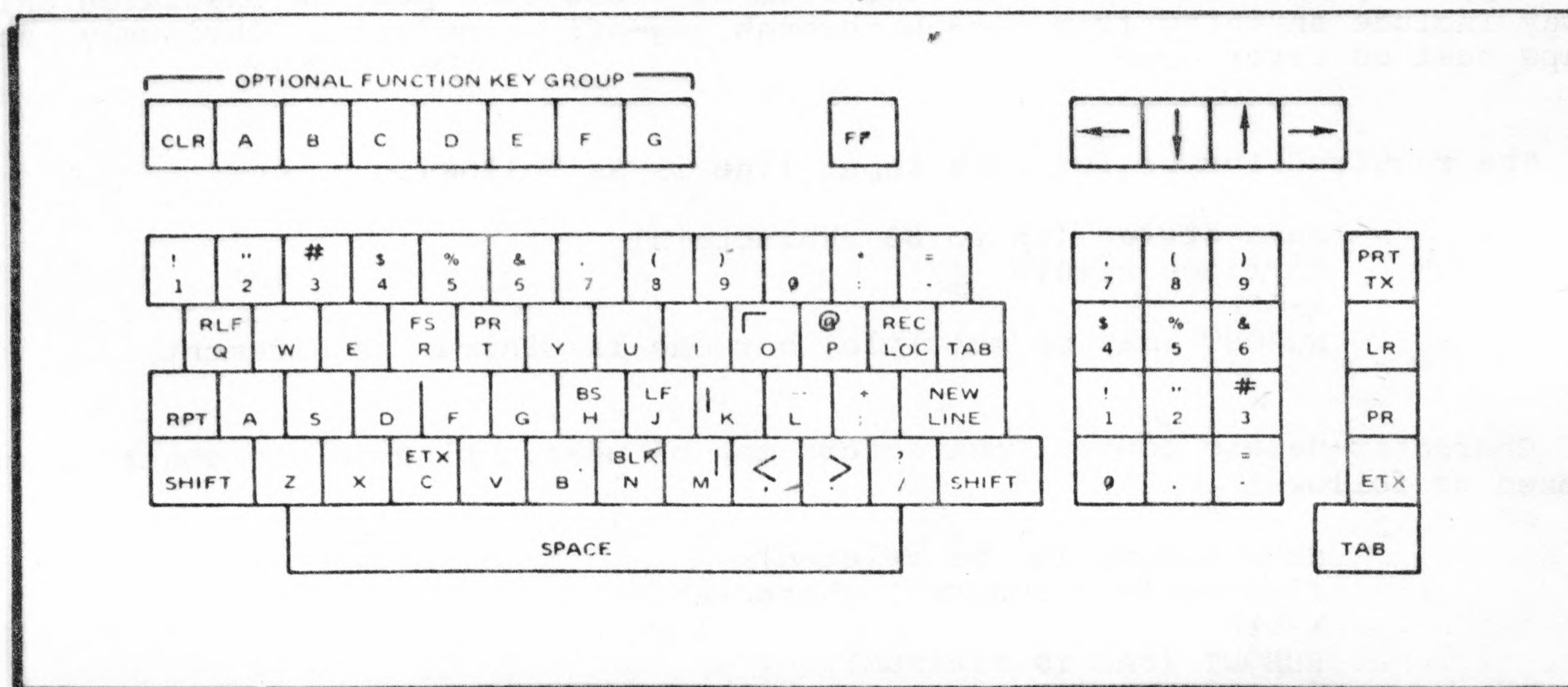


Figure 4-1. Keyboard for 775/785 Series VIP Keyboard

## General Characteristics

The Time Sharing System can interface with most of the VIP terminals. Both synchronous and asynchronous units are available, with line speeds of 1200, 2000, 2400 or 4800 bits per second. A complete page of input may be composed before transmission to the Time Sharing System, and a complete page of output may be displayed. The page consists of 4 to 26 lines, depending on the model. (See Figure 4-2.)

NOTE: The number of characters transmitted or received is subject to limitations of the terminal. Also the user should reference the Remote Terminal Supervisor (GRTS) manual.

Terminal Type	Device Code	Char/Line	Lines/Page	Char/Page	End of Text Symbol	Transmit Symbol	Print Symbol	Receive Symbol
DATANET 760	05	46	04	184	C	T	P	R
DATANET 760	06	46	08	368	C	T	P	R
DATANET 760	07	46	16	736	C	T	P	R
DATANET 760	10	46	26	1196	C	T	P	R
765/775 VIP	11	46	22	1012		none	none	none
785/786 VIP	12	92	22	2024		none	none	none
7700 VIP	13	80	12	960		none	none	none
7700 VIP	14	46	22	1012		none	none	none
7700 VIP	15	80	24	1920		none	none	none

Figure 4-2. Display Devices

The keyboard/display terminals differ significantly from the keyboard/printer terminals in entering data. As mentioned above a complete page can be entered by one transmission; also while the user is composing his input from the keyboard, the terminal is in effect off-line since no data is transmitted until the user initiates the proper transmit procedure.

CAUTION: The system automatically disconnects any terminal which does not input (transmit from the terminal) within ten minutes.

Also when a user requests output (e.g., LIST), only a full page is sent even though the file could be longer. The remainder of a file may be displayed, a page at a time, by continued requests for transmissions until the end-of-file is reached.

### Data Display and Transmission

The keyboard is similar to that used with hard-copy type terminals. Most of the keys are in the same physical location on the keyboard. Although the user should depend upon the instruction manual accompanying the unit for the function of special keys, some of those special keys are discussed here.

After the unit is turned on and allowed time to warm up, an entry marker should appear in the upper left-hand corner of display unit. This entry marker is the position on the display where the next character or space will be entered. As a key or space is struck the entry marker advances to the next position. When the end of a line is reached, the entry marker will move to first character position (left side) of the next line. When the end of the last line on the page is reached, the entry marker moves to the top of the screen, first character position.

NOTE: The entry marker can be positioned by the use of special keys without changing or clearing the display. The most obvious are the four arrow keys. Some devices (see Figure 4-1) have line return (LR), page return (PR), new line, backspace (BS), forward space (FS), keys which also position the entry marker without changing the display.

The entry marker also marks the point where transmission to the computer is to begin. For example to enter one line, possibly a one-word command, the steps are as follows:

1. Type the word or words :LIST  
The entry marker will now appear at the space following the "T" in LIST.
2. Depress ETX, end of text. The ETX symbol is two vertical lines (||) or C.
3. The entry marker moves another space and must be moved to the first letter to be sent, "L" in this case. LR, line return, will return the marker to the beginning of the line or with the backspace key (←).
4. Depressing TX (transmit) will send the line, LIST, to the system.

On some keyboard/display terminals the transmit sequence (ETX, LR, TX) is generated by depressing a single function key.

In general, a transmission is bracketed by the position of the entry marker, and end of text which may be one or more lines.



Log-On

After turning on the unit, the user should allow time for it to warm up. Some units require approximately 30 seconds to warm up. The entry marker should appear on the screen before continuing.

When the keyboard/display unit is ready, the user dials the number of his time sharing center. The following is a typical log-on procedure (user responses are underlined; comments in parentheses):

```
$$SC PASSWD,NN,TSS  
(screen is cleared by the system)  
112501  
HIS SERIES 6000 ON DATE AT TIME CHANNEL NNNN  
  
USER ID -  
UR-IDENT  
PASSWD - -  
                                     (password erased from screen)  
OLD FILEX  
(screen is cleared by the system)
```

Where: SC - user selected station code

PASSWD - password

NN - number of lines per page: 04, 08, 12, 16, 22, 22L, 22N, 24  
or 26

The initial log-on input:

```
$$SC PASSWD,NN,TSS
```

is a requirement of the particular system configuration and may vary. The users should be notified by their computer operations group as to the exact format required at the site. The message:

```
TERMINAL DISCONNECT   ISP
```

is displayed to notify the user that the log-on was incorrectly entered. Since the terminal is still on-line, the user may attempt to enter a corrected log-on message. It is not necessary to re-dial.

\*

## Log-Off

The log-off procedure and the log-off message are identical to that for the teleprinter. However, in addition to the BYE command the display units may be disconnected with the command, "\$\*\$DIS".

## Unique Features

In addition to the log-on procedure, the following features are unique to the keyboard/display terminals.

- TAPE/#TAPE/LUCID/#LUCID command cannot be used.
- More than one line of data may be transmitted to or received from the system at one time.
- Special character-delete and line-delete control characters are not applicable, as all errors may be corrected by positioning the entry marker over the erroneous character and typing the correct one.
- A LIST or other output commands will display only one page, up to 2024 characters. If a file is longer than one page, the remainder of the file may be displayed by either depressing the print (PRT) key or repeating the output command.
- A "BREAK" (interrupt) signal is transmitted to the system by means of the following control message:

\$\*\$BRK

This message can be used to interrupt some lengthy output process, such as the unwanted remainder of a long listing, or to interrupt execution of a user's program.

- With the use of algebraic subsystems BASIC and ABACUS, the up-arrow (↑) symbol used as the exponentiation operator is replaced by a BLK (blink) character preceding the exponent. The blink character itself is displayed as a blank, and causes the exponent character(s) following the blank, in turn, to blink.

## 7700 SERIES VIP TAPE CASSETTE AND PRINT PAGE ADAPTER OPERATIONS

The 7700 Series VIP can read or write to a tape cassette unit or direct data to a print page adapter for hard copy printing. All three units are capable of offline operation. For a description of offline operation, refer to the 7700 Series Visual Information Projection (VIP) Systems manual, Order Number AL29.

### Output to Cassette

Cassette output is initiated with a WRITE TAPE n command where n is the tape number (n=1 or 2, default tape number =1). The Time Sharing System directs output to the designated tape cassette unit. The output is also displayed on the screen. The WRITE EOF n command disables the cassette output mode of operation and writes an end-of-file (EOF) on the current tape.

Example:

```
*OLD TEST
*WRITE TAPE 1
*LIST
10 THIS IS A TEST      Displayed on screen and
20 OF THE TAPE        sent to tape cassette.
30 CASSETTE FUNCTIONS
40 END
*WRITE CEOF 1         Writes EOF on tape.
EOF
*
```

Input From Cassette

Cassette input is initiated with a READ TAPE n command which is similar to the TAPE command used to initiate paper tape input. Data from the cassette is transmitted in variable sized blocks of up to a maximum block size equal to the screen size. The cassette tape must have an EOF (written on the tape with the WRITE CEOF n command) to terminate the TAPE READ sequence.

Example:

```
*NEW
*READ TAPE 1
10 THIS IS A TEST      Sent from cassette
20 OF THE TAPE        to screen and to
30 CASSETTE FUNCTIONS the system.
40 END
EOF
*
```

Echo Back

The 7700 terminal operator can have the data that is keyed in recorded on the print page adapter as well as on the system by depressing the PRINT KEY. The Time Sharing Executive "echoes" the input block back to the printer.

Backspace Cassette

The BSP TAPE n command enables the terminal user to backspace the cassette tape one record.

Rewind Cassette

The REW TAPE n command enables the terminal user to rewind the cassette tape.

## Output to Printer

The PTON command enables subsystem output to be routed to the print page adapter. The PTOF disables the printer mode of operation in TSS.

### Example:

```
*OLD TEST
*PTON
*LIST
```

All output goes to printer - not displayed on the screen.

```
*PTOF
*
```

or

```
*NEW
*PTON
*CATALOG
```

Catalog output goes to printer - not displayed on the screen.

```
*PTOF
*
```

## Continuous Output Mode

When data is routed to the cassette unit or to the printer, a continuous stream of data is transmitted. While the data is visible on the screen, for cassette functions, the blinking asterisk is omitted and there is no need to depress the print key for each page.

## Summary of 7700 Cassette/Printer Commands

PTON	print on
PTOF	print off
WRITE TAPE n	write cassette n
WRITE CEOF n	write EOF on cassette n
READ TAPE n	read cassette n
BSP TAPE n	backspace cassette n one record
REW TAPE n	rewind cassette n

Where: n is optional; if not specified, the default value is 1. If specified, n is either 1 or 2.

Caution: These commands can only be used on 7700 Series VIP (device 13, 14, or 15 octal).

## COMMANDS FOR VIP TERMINALS

In addition to the cassette/printer commands, the commands described below are only for VIP-type terminals.

### Form Feed Commands

The no form feed command (NFORM) allows the user to control the transmission (from the Time Sharing Executive) of the form feed character after the prompt for a page request. The NFORM command can be entered at system selection level at build input level. It causes the cursor to be returned to column one - in character position one of the screen. This leaves all of the previous data on the screen. The command FORM reverses the NFORM command, whereby form feeds are again transmitted after the prompt for a page request.

### Case Commands

These commands are for VIP terminals which can display both uppercase and lowercase characters. At log-on all lowercase characters are transliterated to uppercase characters. The command LCASE allows both uppercase and lowercase characters to be transmitted to the VIP terminal. The command UCASE reverses the operation.

## SECTION V

### SERVICE SUBSYSTEMS AND PROGRAMS

#### ABACUS SUBSYSTEM

The ABACUS (ABC) subsystem is an algebraic-expression evaluator analogous to the function of ABACUS is that of a powerful desk calculator, with the ability to calculate and remember the value of symbolic variables. It also features summation operation and employs commonly used mathematical constants and functions.

#### Use of ABACUS

\*ABC

The initial call may contain, on the same line, the expression to be evaluated; e.g., \*ABC 1.379+2. Otherwise, ABACUS issues a question mark (?) as a request for input. The possible forms of input are:

? expression

? x = expression

where x is a variable

? FOR x = a,b,c;  
expression in x

where a,b,c specify a  
range of values for x

? FOR x = a,b,c;  
y = expression in x

where y is also a  
variable

From one to three FOR specifications may be employed before the expression, separated by semicolons; i.e., FOR x = ...; FOR y = ...; FOR z = ...;

The results of each expression evaluation are printed immediately. ABACUS then issues another request for input (?). A null response (i.e., carriage return only) or DONE causes an exit from the subsystem.

An expression is composed of operators, numbers and/or variables, and/or constants and functions, conforming to ordinary arithmetic and algebraic rules. The permissible operators are:

+ (addition)  
- (subtraction)  
\* (multiplication)  
/ (division)  
↑ (exponentiation)  
& (summation)

Parentheses may be used to indicate grouping of operations, according to standard usage.

### Numbers

Numbers may be written as:

Integers: e.g., 1, -25, 7063  
Fractions: e.g., .1, -.0005, .3681400  
Mixed numbers: e.g., 1.5, 812.764  
Scientific notation: e.g., 1E10, 2.41E-3, -3215E7

Numeric operands may contain up to 18 significant digits. Printed result values are limited to a maximum of seven places in the fractional part. Precision is kept internally to 18 places, however.

### Variables

Variables (names to which numeric values can be assigned) are composed of one to eight alphanumeric characters, the first of which must be alphabetic; e.g., A, B5, SUMSQUAR. There are two types of variables, according to usage: (1) FOR variables, i.e., those defined in a FOR specification, and (2) label variables; these appear on the left of an equal sign but not preceded by FOR. In input of the form "X = expression", X is a label variable. The distinction to be noted between the two types is that label variable values are remembered between expression evaluations. The values assigned to a FOR variable hold only for the expression associated with the FOR specification(s); they are not remembered for a subsequent expression. For example:

? X = SIN (30/RADIAN)	(X is a label variable)
X = +0.5	- (answer)
? X ↑ 2*27.9	
+6.975	- (answer)

## Constants and Functions

Constants and functions available in ABACUS are:

<u>Constant Name</u>	<u>Predefined Value</u>
PI	3.14159...
RADIAN	57.295.....
E	2.71828...

} internal precision to  
18 significant digits

<u>Function Name</u>	<u>Meaning</u>
ABS (x)	Absolute value of x
ATN (x)	Arctangent of x
COS (x)	Cosine of x
EXP (x)	e to the power of x
LOG (x)	Natural logarithm of x
SIN (x)	Sine of x
SQR (x)	Square root of x
or	
SQT (x)	
TAN (x)	Tangent of x
INT (x)	Integer of x

For trigonometric functions, x denotes an angle measured in radians.

Function names are reserved words; i.e., they cannot be used as variable names. Constant names are not reserved; actually they are remembered variables with preset values. Their value may be changed by using the name as a label variable.

### Summation Operator and FOR Variables

The summation operator, & , may only appear at the beginning of an expression, and the entire portion of the expression following it is assumed to be the argument to be summed (regardless of the use of parentheses). From one to three variables may be given a range of values for the summation by means of FOR statements.

The FOR statement(s) must precede the associated expression in the same line of input, separated by semicolons. The form of the FOR statement is:

FOR x = a, b, c;

Where: a - Initial value of x.  
b - Limiting value of x.  
c - Step value or increment (optional).



If the step value, c, is not specified, 1 is assumed. Substitutions for a, b, and c may be positive or negative integers, expressions, or predefined variables.

For example:

```
? FOR X = 1, 5; FOR Y = 7, 50, 9; Z = & (X+Y)*PI
Z = 2199.1149
```

In summations, all FOR variables are treated as summation indices and in the case of summations over two or three FOR variables, the indicated summations are nested. Each summation variable takes on the values a, a+c, a+2c, ... up to but not exceeding the value b. Thus the expression above would expand as follows:

$$Z = \sum_{X=1,2,\dots,5} \sum_{Y=7,16,\dots,43} (X+Y)\pi$$

$$Z = ((1+7)\pi + (1+16)\pi + (1+25)\pi + \dots + (5+34)\pi + (5+43)\pi)$$

Although an expression containing a summation operator must be preceded by one or more FOR specifications (in order to be meaningful), FOR variables may also be used in expressions that do not contain the & operator. For example:

```
? FOR A = 3,11,2; FOR B = 1,3; X = A↑B
```

In these cases, the expression will be evaluated separately for each possible combination of FOR values (as is done in FORTRAN). The output from the example expression just above would appear as:

<u>A</u>	<u>B</u>	<u>X</u>
3	1	3
3	2	9
3	3	27
5	1	5
5	2	25
5	3	125
7	1	7
7	2	49
7	3	343
9	1	9
9	2	81
9	3	729
11	1	11
11	2	121
11	3	1331

If a label variable is used, as in the above example (X), the last determined value is remembered for the variable.

## Continuation Lines

ABACUS accepts only one expression at a time, with or without associated FOR specifications. Normally, this is represented on one line of terminal input. If, however, the expression (with any preceding FOR specifications) requires more than one line, the first line can be terminated with a \$ plus carriage return combination--rather than a carriage return only. This denotes the next line to be continuation of the first, and ABACUS responds with another input request (?).

## Order of Evaluation and Use of Parentheses

Expressions are evaluated from left to right, with operations performed in the following order:

1. functions
2. ↑
3. \* and /
4. + and -
5. &

Care must be exercised, especially with regard to successive exponentiation, to ensure that the order of evaluation implied by the above rules is the order intended. If the intention is otherwise, parentheses must be used to force the desired order of evaluation.

All operations within a subexpression enclosed in parentheses will be performed before any operations to the right of that subexpression. Grouping of operations to any depth may be indicated by means of nested sets of parentheses. An exception is the summation operator, & , which may not be enclosed in parentheses.

Implicit multiplication is allowed preceding a parenthesized subexpression, but not between two such subexpressions. For example:  $3(x)$  is equivalent to  $3*(x)$ ; but  $(x)(y)$  is illegal, and must be written as  $(x) * (y)$ .

The argument of a function, which may be any expression, must be enclosed in parentheses.

## Mode and Precision of Calculation

All calculations are performed in double precision floating-point, with consequent precision (but not accuracy, necessarily) to 18 places. Displayed results are limited to a maximum of seven places in the fractional part (rounded), but 18 significant digits are carried internally. This may result in a small discrepancy between displayed intermediate and final results, in a sequence of related evaluations.

## ASCII-TO-ASCII CONVERSION SUBSYSTEM

### ASCASC Subsystem/Command

The FORTRAN, TSS ALGOL and TSS JOVIAL language systems require the following translations between the time sharing format ASCII data files (media code 5) and the standard system format ASCII data files (media code 6): ASCII media codes are described in the TSS System Programmer's Reference Manual, in Section VI.

- Time-sharing format ASCII files may be converted to standard system format ASCII files to be used as input data for FORTRAN, TSS ALGOL, and TSS JOVIAL.
- Standard system format ASCII files must be converted to time sharing format ASCII files which can be listed at a terminal.

The ASCASC subsystem performs these translations. ASCASC may be called at the subsystem selection level or at the command level at the build input level of the language system requiring the translation. The format of ASCASC is as follows:

```
ASCASC filedescr 1; filedescr 2
```

Where: filedescr 1 - Input file to be converted

filedescr 2 - Output file to be created

### Execution

In the execution of the ASCASC command, the input file is read and converted to the format required for the output file. The input file's record control word is checked to determine the format of the file. If the record media code is 5, the file is in time sharing ASCII format. If the record media code is 6, the file is in standard system ASCII format. Based on this determination, one of the following translations is performed:

1. If the input file is in time sharing ASCII format (character-oriented file), the characters in the file are read and converted to the word-oriented standard system ASCII format for the output file.
2. If the input file is in standard system ASCII format, the words in the file are read and converted to the time sharing ASCII format for the output file. Up to 72 characters will be converted.

If neither record media code 5 or 6 is found in the record control word, a message is sent to the user to tell him that the file he specified is not an ASCII file.

## FILE SYSTEM

The Time Sharing System utilizes the capabilities of the GCOS file system, which is a logical mechanism for storing and retrieving permanent files and is common to all system programs operating under the General Comprehensive Operating Supervisor (GCOS). A file system can store many files on many unspecified external, background storage devices, and the user normally need not be concerned with the device his file is on nor with the characteristics of the device.

### File System Structure

The file system is described in detail in the File Management Supervisor reference manual. However, the main features of interest to the time sharing user are repeated here.

The file system is a tree structure whose origin is the system master catalog. The primary nodes of the tree are user's master catalogs; the lower level nodes are subcatalogs created by the user. The terminal points of the structure are the files themselves. Figure 5-1 shows the file system's hierarchical structure. \*

### Catalogs and Files

A catalog consists of a definition containing a catalog name, password, and permissions. Since it contains no user data, a catalog can be neither read nor written except by the file system itself. An ACCESS function is provided, however, to direct the file system in the creation and modification of subcatalogs.

A file, as known to the GCOS file system, consists of a definition containing file name, file size, password, permissions, and a description of the physical file space. The file definition is distinct from the physical file space which may contain user data and can be read or written.

### Passwords

Passwords may be attached to any catalog or file. A password allows a user to traverse a catalog/file string. A user can get to a given catalog or file only if he can give the passwords for all higher level catalogs in the string. (When traversing a string, a password must not be given if none has been attached.) The originator of a given string is required to give the necessary passwords when traversing a string.

## Permissions

When a file or catalog is created or modified, the creator or modifier may specify what actions are permitted on the file, catalog, or subordinate files or catalogs by what users. Permissions may be specified in general or for specific named users. Specific permissions replace, not extend, general permissions. Both general and specific permissions may be specified for a catalog to apply to subordinate files or catalogs. When there are several levels of subordination, the permissions at each level are accumulated.

The permissions and their function are listed below:

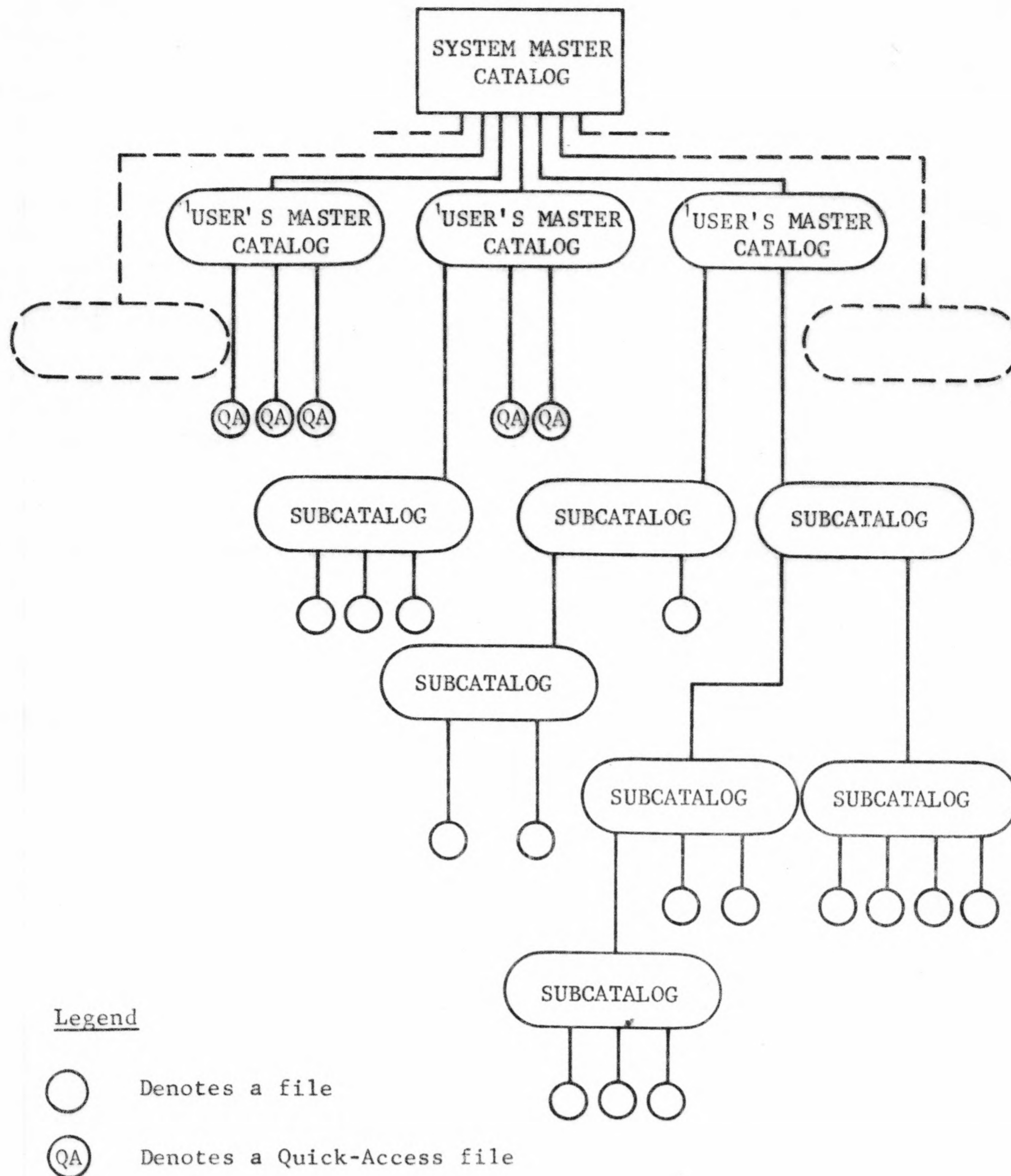
READ or R	Allow transfer of information from file to program but not from program to file.
WRITE or W	Allow transfer of information both from file to program and program to file. WRITE permission implies READ permission.
APPEND or A	Currently treated as READ permission. WRITE permission implies APPEND permission.
EXECUTE or E	Allow transfer of information from file to program but only for a compiler or loader. Once these have done their work, do not allow any transfer between program and file. Anyone with READ permission (or WRITE since it indicates READ) has EXECUTE permission.
RECOVERY or REC	Allow WRITE when file is abort locked or has defective space. Anyone with RECOVERY permission is also given permission to WRITE and hence READ.
PURGE or P	Allow file to be deleted (file description to be deleted and file space to be returned with or without prior overwrite of space) or catalog to be deleted and all subordinate files to be deleted. Anyone permitted to PURGE can also perform any of the actions permitted by RECOVERY, including WRITE and hence READ.
CREATE or C	Allow files and catalogs to be entered as subordinate to this catalog.
LOCK or L	Allow a user to security lock the file or catalog (which security locks subordinate files) or to remove an existing security lock. A security lock does not apply to a user with LOCK permission (since they are able to remove the lock).
MODIFY or M	Allow catalog or file description to be modified and allow entries to be made in catalog for subordinate files or catalogs. Anyone permitted to MODIFY is allowed to perform any actions since he could change permissions to give himself permission to perform these. Hence MODIFY includes CREATE, LOCK, and PURGE, which in turn includes RECOVERY and hence WRITE and READ.

## General, Specific, and EXCLUDE Permission

A permission given to everyone is called a general permission; one given only to a named user is called a specific permission. If a file or catalog has both general and specific permissions specified for it, the general permissions apply to everyone except those users named in specific permission specifications. A general permission to WRITE, for example, can be restricted to a READ permission for a particular user by naming the user with a specific READ permission.

A user never has both the permissions specified for everyone and also those specified specifically for him. The user has only those permissions specified specifically for him, or if there are none, then those specified for everyone.

In order to restrict a user from all general permissions without at the same time giving him some permission, the user can be named in an EXCLUDE permission. The EXCLUDE, which is only a specific permission, withdraws general permissions from the named users without giving them any specific permission.



All user-ID's must be unique within the system; all subcatalog and file names are automatically qualified by the user's master catalog name and the names of any intermediate subcatalogs. The system master catalog cannot be accessed by the normal user.

---

<sup>1</sup>Identified by the user-ID.

Figure 5-1. Logical Structure of the File System

When operating under the Time Sharing System, each file requested by a NEW or OLD command is a temporary working file, called the current file, that will disappear at the end of a user's session at the terminal unless he saves it. In the case of OLD, the current file is effectively a scratch copy of the named permanent file, allowing modifications to be made and tested without changing the original OLD file, unless the user explicitly resaves on that file.

File content is written (i.e., stored) on a permanent file only by means of the SAVE or RESAVE commands. In addition, the SAVE command implicitly causes the file to be created. If the file named by the SAVE command does not already exist, the system automatically creates a permanent, external file and writes the contents of the current working file onto it. If the SAVE operand consists simply of a file name, the file created will be of the quick-access type. This type file emanates directly from a user's master catalog without intervening subcatalogs (see Figure 5-1). If a catalog-string precedes the file name in the SAVE operand, the subcatalog(s) named was previously created via the ACCESS subsystem.

If the file to be written on already exists (whether created by a prior SAVE, by ACCESS, or by a batch activity), a RESAVE command causes the contents of the current file to be written to the named permanent file, replacing whatever contents existed in the named file.

If the characteristics of the quick-access type of file are suitable to the user's requirements, the command language facilities offered by the system are sufficient; ACCESS need not be used. \*

## ACCESS SUBSYSTEM

### Capabilities

For users who wish to utilize some or all of the capabilities of the file system, the ACCESS subsystem provides an interface. This interface allows the user to perform the following:

- Create hierarchical structures of subcatalogs and files.
- Attach passwords to his subcatalogs and files.
- Give general permission to all other users to access his files in specified ways.
- Give specific permissions, by user-ID.
- Protect a given file or set of files against any mode of access.
- Gain the permitted types of access to another user's files.
- Gain the permitted types of access to files created in the batch-processing environment.
- Modify catalog name, password, and/or permissions on an existing catalog.
- Modify file name, size, password, and/or permissions on an existing file.



- Purge or release an existing file or catalog/file string.
- List all catalogs and files which emanate from a given catalog.
- Rename files temporarily for a given job.
- Create a random file with a logical-record-size attribute, if such a requirement exists.

### Use of ACCESS

ACCESS consists of a number of functions which provide a conversational facility for the following:

- Creating and purging/releasing catalogs and files.
- Modifying catalog and file attributes (name, size, password, permissions).
- Accessing and deaccessing files.
- Listing catalogs and files.

The operation of ACCESS consists of responses, via the terminal, to a sequence of English language questions. All of the standard vocabulary associated with the user's responses may be abbreviated for convenience in input. A nonconversational short form of input is also provided for more experienced users and users with batch FILSYS experience.

ACCESS is not a means of reading or writing permanent file content. OLD and SAVE/RESAVE perform these functions. ACCESS is selected, for example, to create a file (i.e., the file definition and the file space), before the substantive file content is built.

Using the ACCESS subsystem, a file may be created by specifying the ACCESS subsystem name, the create command (CF), and the file description with associated parameters in one string. For example:

```
*ACCE CF,/CAT1$PASSWD1/CAT2$PASSWD2/FIL1$PASSWD3,B/1,3/,R
```

The creation of the file is performed as in the conventional question-response sequence in the ACCESS subsystem. After this function is performed by ACCESS, the user is returned to the calling level.

NOTE: If the following conditions are met, the user is immediately returned to the calling level upon completion of the function being performed.

1. The ACCESS command and the function to be performed are entered on a single line.
2. No error is encountered while ACCESS is processing the function.

3. The input line is not terminated by the delimiter \* or \*\*.

For example:

\*ACCE CF,NEWFIL,B/4/

would result in return to build mode with no intervening output.

If a nonconversational ACCESS request is terminated by the delimiter "\*", ACCESS will perform the indicated function and automatically switch to conversational mode, retaining the catalog structure of the nonconversational request. If "\*\*\*" delimits the request, the FUNCTION? message will be issued. Although this is normally done in nonconversational mode after performing a function, the "\*\*\*" delimiter provides a means of cancelling the effect of the above. For example:

\*ACCE CF,NEWFIL,B/4/\*\*

would result in issuing the FUNCTION? request upon completion of performing the CF function.

Some general rules can be cited for the use of ACCESS.

1. The ability of a user to access files and otherwise manipulate catalog/file structures (e.g. modifying and purging) depends upon his knowing the necessary file definitions. Beyond this, the file system has two file and catalog protection features--passwords and permissions.

Permissions provide the file creator with a positive protection feature. If permissions are not explicitly granted, his catalogs and files are completely protected by default. The user must assign to others any degree of access he wishes them to have. But, since specific permissions for a given user do not add to, but replace, any general permission that may have been given, specific permissions may be used to exclude a given set of users from one or more types of access.

Passwords provide an additional level of protection. If passwords are assigned by the creator of a catalog/file string, they must be supplied in order to pass through the string.

The creator of a catalog/file string is exempt from any ACCESS mode restrictions he imposes (i.e., he implicitly has all permissions for his own catalogs and files), but he must give all passwords.

MODIFY permission which, when attached to a catalog, allows all or selected users to change file names, catalog names, file size, passwords, and/or permissions, and also implies the ability of the specified user(s) to create catalogs and/or files emanating from the associated catalog.

2. The definition of a particular catalog or file must include the names of all higher level catalogs that must be traversed to arrive at that point. The catalog string would include at least the user's master catalog. A file definition, then, is the complete catalog string plus the file name.

3. The user's master catalog is implicitly created for him the first time a file is created. It has no password or permissions associated with it, but it may subsequently be modified by the user to include them.
4. If a delimiter immediately precedes the carriage return for an input line, the question MORE? is issued to the user requesting continuation input.

### Identifiers and Delimiters in User Responses

User responses are composed of the following:

- Identifiers
- Keywords
- Word delimiters
- Line delimiters

Identifiers consist of file names, catalog names, user-IDs and passwords. They can be composed of alphabets, numerics, periods, and minus signs. Each identifier can be up to 12 characters in length except for file names specified with the Access and Deaccess File functions, which are limited in length to eight characters; however, in response to the question FILE NAME?, issued by the Access File function, a file name of up to 12 characters may be specified if followed by an alternate name of eight characters or less, enclosed in quotation marks. The alternate name must be used for all references to such a file until after it has been deaccessed.

Keywords consist of function names, access types (permissions), and several file type parameters of limited interest. Keywords are used in responses to questions. All keywords, except EXCLUDE, DELETE and GEN'L, can always be abbreviated to the initial character, or a 2-character acronym in the case of function name (e.g., R for READ permission or CC for Create Catalog function).

The file size specification in the response to FILE NAME, SIZE (IN LLINKS), MAX SIZE, MODE? (Create File function), is a decimal number denoting the number of blocks required. This may be considered a special case of a keyword. MAX SIZE may be omitted, in which case the maximum size will be as specified for the initial size. If the MAX SIZE is to be unlimited, the letter "U" may be given.

Word delimiters are the slant or virgule (/), the dollar sign, and the comma. Blanks may be used freely in responses except within the function identifier (either a blank or comma may delimit the identifier).

## WORD DELIMITERS

The / delimiter has two functions:

1. In catalog-strings, / indicates that a subcatalog name follows and is concatenated to the preceding catalog in the string. An initial / indicates that the following subcatalog-string (if any) is concatenated to the user's master catalog. A response to CATALOG STRUCTURE TO WORKING LEVEL? of / and carriage return is equivalent to the user's own user-ID; i.e., it positions the user to his own master catalog.
2. In specific permissions, a / indicates that a user-ID follows.

The \$ delimiter is used only to concatenate a password to a catalog or file name.

The , delimiter is used as a general separator for keywords; i.e., for separating access types and sizes, and separating file names from following keywords or sizes.

## LINE DELIMITERS

The line delimiters are a carriage return, an asterisk plus carriage return, or a double asterisk plus a carriage return. Each of these serves to terminate a response, but with a different effect.

1. A carriage return following a response generally signifies that the user wishes to remain at the same catalog position (if relevant), and proceed to the next question in logical sequence. This may be the next question in a set, or the initial question again.

When only a carriage return is given, (i.e., a null response) however, it has several possible meanings:

- In response to the question CATALOG STRUCTURE TO WORKING LEVEL? a carriage return only is equivalent to the user's own user-ID or a / and carriage return. Any of these responses requests that the user be positioned to his own master catalog.
- A carriage return only following a question that logically requires a response (e.g., NEW CATALOG NAME?), causes an immediate return to the question FUNCTION?.
- The question SPECIFIC PERMISSIONS? recurs each time a response is given (delimited by a carriage return), since only one set of specific permissions can be given in each. If only a carriage return is given, the information received so far is processed, and the first question below CATALOG STRUCTURE TO WORKING LEVEL? is reissued (i.e., NEW CATALOG NAME? or FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE?) allowing a new catalog or file to be created at the same catalog level.
- A carriage return only or the response DONE to FUNCTION? causes the subsystem to terminate.

2. If a single asterisk plus a carriage return is given in reply to a question, either with or without a substantive response, ACCESS processes the information it has and returns to the first question at the same catalog level (e.g., to skip any further questions in the set). ACCESS, of course, must have sufficient information to process.
3. If a double asterisk plus a carriage return is given, either with or without a substantive response, ACCESS processes the information it has and returns to the question FUNCTION?. It implies that the user is finished with the current function. ACCESS, of course, must have sufficient information to process.
4. Multiple operations for a function can be specified on the same line (delimited by semicolons) for any nonconversational ACCESS function. For example, to create two files, FILA and FILB, subordinate to CAT1, the following input could be entered:

CF,/CAT1/FILA,B/1,12/,R;/CAT1/FILB,B/1,12/

### ACCESS Functions

The initial communication from ACCESS, following subsystem selection, is a request for a choice of function; i.e., FUNCTION?.

The functions that may be requested and the effect produced by each function are as follows (function may be spelled out or abbreviated as indicated by the underlining):

- CREATE CATALOG - Creates a subcatalog.
- CREATE FILE - Defines file space and attributes for a given file name.
- ACCESS FILE - Brings a file into the Available File Table.
- DEACCESS FILE - Takes a file out of the Available File Table.
- MODIFY CATALOG - Modifies the name, password, and/or permissions associated with a given catalog.
- MODIFY FILE - Modifies the name, maximum size, password, and/or permissions associated with a given file.
- PURGE CATALOG - Deletes a catalog from the system along with any subcatalogs and files which are subordinate to it. All released file space is overwritten.
- PURGE FILE - Deletes a file from the system, overwriting the released file space.
- RELEASE CATALOG - Deletes a catalog from the system, along with any subcatalogs and files which are subordinate to it. Any released file space is not overwritten.
- RELEASE FILE - Deletes a file from the system, but without overwriting the released file space.
- LIST CATALOG - Lists the names of the catalogs and files which emanate from this catalog.
- LIST SPECIFIC - Lists in detail the description of the catalog or file specified.

Following the response to FUNCTION?, ACCESS asks the user to describe the catalog string, catalog, or file. Each function has a fixed set of questions, with several of the questions common to each set. Some of the questions do not logically require a response; e.g., PASSWORD? (there may be none). If no response is applicable, only a carriage return is given.

All the functions, except DEACCESS FILE, first request a definition of the existing catalog string. Then the name of the catalog or file to be processed is next, along with size attributes in the case of a file. Passwords and permissions are then requested, as appropriate.

#### SHORT FORM USAGE OF ACCESS FUNCTIONS

Once the user has become familiar with the conversational, or question/response sequence, form of ACCESS, he may use a short form of function specification which effectively eliminates questions normally asked. In this short form, the function name (e.g., CREATE FILE) is followed directly by all the user-entered information needed to complete the function specification, usually all on one line. Each item of information is separated from other items by commas.

The information entered in this short form is much the same as that given as responses in the conversational mode, but with additional keywords. The format is similar to that of the batch file system (FILSYS) input.

Certain functions (MODIFY CATALOG and LIST SPECIFIC) require short-form usage for operations involving only the user master catalog; i.e., the conversational mode cannot be used for these ACCESS requests.

The general format, in response to the initial question FUNCTION?, is:

function-name, catalog/file string, option, ..., option

Where: catalog/file string is the same as in conversational responses, and options are:

<u>Form</u>	<u>Function</u>
PASSWORD } /password/ PASS	password assignment
access-type	general permissions
access-type/ user-ID, ..., user-ID/	specific permissions
BLOCKS } /x,y/ LINKS } SIZE } LINKS }	size assignment
MODE/mode/	mode assignment
LRS/SIZE/	logical record size

DEVICE/name or type/ request a specific device name  
or type and must be specified  
as follows:

DSS167  
DSS170  
DSS180  
DSS181  
DSS190  
DSS191  
DSS270  
MSS800  
MS0310  
MS0400

AF )  
OPEN )

access file after creating it

CLEAR

zero (erase) file space after  
creating and accessing it

\*  
Access type and mode are defined under each applicable function description. Options may appear, comma separated, in any order. The keywords BLOCKS and LINKS may be abbreviated to the first letter, as may the access-type and mode options. Options unique to the Modify Catalog and Modify File functions are described along with those functions.

All replies may be extended to two or more typing lines by terminating a line with a word delimiter (slant, comma, or dollar sign plus carriage return), at a convenient point, implying that the input is not complete but is to be carried over to the next line or lines.

#### QUESTIONS AND RESPONSES

Sets of questions associated with each function follow, along with the general form of the response to each question. The minimum required user response is underlined for illustrative purposes. Each set is followed by illustrative examples.

#### CREATE CATALOG

FUNCTION? CC

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password

NEW CATALOG NAME? cat-name

PASSWORD?

XXXXXXXXXXXX

GENERAL PERMISSIONS? access-type, ..., access-type

SPECIFIC PERMISSIONS? access-type, ..., access-type/user-ID/user-ID/...

The access-types follow; all may be spelled out, or abbreviated as underlined; except for EXCLUDE and LOCK, which must be spelled out:

<u>Permission</u>	<u>Acceptable Abbreviation</u>	<u>Attaches Permission(s)</u>
<u>READ</u>	R	R,E
<u>WRITE</u>	W	R,W,A,E
<u>APPEND</u>	A	A
<u>EXECUTE</u>	E	E
<u>PURGE</u>	P	R,W,A,E,P,REC
<u>MODIFY</u>	M	R,W,A,E,P,M,LOCK,C,REC
LOCK	(none)	LOCK
<u>CREATE</u>	C	C
<u>RECOVERY</u>	REC	R,W,A,E,REC
EXCLUDE	(none)	EXCLUDE (specific permission only)

If no response to the question SPECIFIC PERMISSION? is given, (i.e., only a carriage return), the catalog is created and the question NEW CATALOG NAME? is reissued.

Example replies (user responses are underlined):

FUNCTION? CC

CATALOG STRUCTURE TO WORKING LEVEL?

JDOE/CAT1\$ABC

This response states that there is a subcatalog named CAT1 that is concatenated directly to the user's master catalog identified by the user-ID JDOE, and that it is desired to create a new catalog from this level. The password ABC was attached to catalog CAT1 when it was created.

NEW CATALOG NAME? CAT2

This response indicates the name of the catalog, CAT2, created at this point.

PASSWORD?

~~XXXXXXXXXXXX~~

The response, AOK, is entered on the strikeover mask, indicating that this is the desired password.



GENERAL PERMISSIONS?

The lack of a response here indicates that general permission is not granted at this level. A response of READ would indicate that any unspecified user has permission to read and execute (if meaningful) any file that emanates from this catalog.

SPECIFIC PERMISSION? READ/BJONES/ASMITH

SPECIFIC PERMISSION? WRITE/ALLONG

This combination of responses states that the users who have logged onto the system under the names BJONES and ASMITH can pass through this level with read or execute permission for any files below, and that the user ALLONG can pass through with read, write, execute, and append permissions.

SPECIFIC PERMISSION?

The carriage return alone means that no further specific permissions are to be given; the catalog is now created and the question

NEW CATALOG NAME?

is reissued, allowing the user to create another catalog at the same level (i.e., also emanating from CAT1).

Alternative forms of the response to CATALOG STRUCTURE TO WORKING LEVEL? are as follows:

/CAT1\$ABC

Assuming the user to be JDOE, this response is equivalent to the one given above, JDOE/CAT1\$ABC. The initial slant indicates the user's own master catalog.

A response of / indicates that the user desires to create directly from his master catalog. This response is equivalent to his user-ID alone.

Example of short form reply:

FUNCTION? CC,/CAT1\$ABC/CAT2,PASSWORD/AOK/,READ/BJONES,  
MORE? ASMITH/,WRITE/ALLONG/

CREATE FILE

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password

FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE?

file name, initial size (llinks), maximum size (llinks), mode(R or L)

PASSWORD?

~~XXXXXXXXXXXX~~

GENERAL PERMISSIONS? access-type,...,access-type

The access types are the same as those for Create Catalog.

SPECIFIC PERMISSION?

access-type,...,access-type/user-ID.../user-ID

Random File Specification: If required, a file can be created with a random-access-treatment indication, by responding to the FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE? question as follows:

file name, initial size, max. size, R

If random (R) is specified, a further question will be asked:

LOGICAL RECORD SIZE? record size in words

Random-I/O files for Time Sharing FORTRAN may have a logical record size attribute; if use of random files does not require this attribute, a response with a carriage return only is required.

ACCESS FILE? YES, Y, CLEAR, or C

This option allows the user to access (open) a file at the time it is created. If CLEAR or C is specified, the file space will be zeroed.

Example replies (user responses are underlined):

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL?

/CAT1\$ABC/CAT2\$AOK

This response defines user-ID/CAT1/CAT2 as the catalog-string from which the file is to emanate. The initial slant indicates that the succeeding string is concatenated to the user's own master catalog.

FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE? FILL,4,12

This response asks for a file space of four llinks initially, with a maximum eventual size limit of 12 llinks, named FILL. Since mode is not specified, the file will be created for sequential (linked) usage.

PASSWORD?

~~#####~~ (null response given)

No password is assigned to this individual file.

GENERAL PERMISSIONS? READ

SPECIFIC PERMISSION?

None are granted at this level, but those granted at the level of CAT2 (CREATE CATALOG in the previous example) apply to this file.

ACCESS FILE? YES

This option allows the user to access (open) a file at the time it is created.

FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE?

This permits creation of other files at the same level.

Example of short form reply:

FUNCTION? CF,/CAT1\$ABC/CAT2\$AOK/FILL,B/4,12/,R,AF

NOTE: File mode by default is linked (sequential); i.e.,  
MODE/LINKED/.

ACCESS FILE

FUNCTION? AF

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password

FILE NAME? filename\$password"altname"

PERMISSIONS DESIRED?

access-type, ..., access-type

The following table summarizes the legal permissions and permission combinations:

<u>Type of Allocation</u> <u>Word</u>	<u>Abbrev.</u>	<u>Allowable Operations</u> <u>on File Content</u>	<u>File Conditions</u> <u>Required</u>	<u>Permissions</u> <u>Required</u>
READ	R	read	no writers, not abort locked	READ
WRITE READ,WRITE	W R,W	read and write	no other writers, not abort locked	WRITE
APPEND	A	append	no writers, not abort locked	APPEND
EXECUTE	E	execute	no writers, not abort locked	EXECUTE
READ,APPEND	R,A	read and append	no writers, not abort locked	READ and APPEND
RECOVERY	REC	read and write	no other writers	RECOVERY
QUERY	Q	read	none	READ
READ, CHANGING	R,C	read	not abort locked	READ
TEST	T	read and write to scratch file	no writers, not abort locked	READ
TEST, CHANGING	T,C	read and write to scratch file	not abort locked	READ
WRITE,C READ,WRITE,C	W,C R,W,C	read and write	not abort locked	WRITE

Random File Specification: A file can be accessed for random treatment, whether created as random or linked, by responding to the FILE NAME? question with:

filename\$password,R

or

filename\$password"altname",R

If the file was created as linked, the random treatment indication is temporary; i.e., for the current access only. If the file was created as random, the ,R specification is superfluous.

Example replies (user responses are underlined):

FUNCTION? AF

CATALOG STRUCTURE TO WORKING LEVEL?

JDOE/CAT1\$ABC/CAT2\$AOK

The user in this case is not the creator of the file to be accessed, so he must define the user's master catalog (e.g., JDOE) from which the file emanates, along with any required subcatalogs and passwords.

FILE NAME? FIL1

If a password were required, it could be concatenated to the name with a dollar sign; i.e., FIL1\$ABC. Otherwise, it will be requested.

PERMISSIONS DESIRED? READ

General Read permission was granted for this file. (Several specific Read permissions were also granted at the level immediately above CAT2.) Termination of this response with only a carriage return causes the file to be accessed and the request

FILE NAME?

to be reissued.

Example of short form reply:

FUNCTION? AF,JDOE/CAT1\$ABC/CAT2\$AOK/FIL1,R

DEACCESS FILE

FUNCTION? DF

FILE NAME? filename (or CLEARFILES, PERMFILES, STARFILES, or TEMPFILES)

The response for this function is the name of the file to be deaccessed. The name supplied is always the name under which the file was accessed, whether this was the actual name or a temporary alternate name. If CLEARFILES is used, all of the user's available files (except \*SRC) are deaccessed including his temporary files. PERMFILES or TEMPFILES may be used to remove all permanent or temporary files (except \*SRC) from the AFT, respectively. STARFILES removes all files (except \*SRC) from the AFT that contain an asterisk in the name. Note that the input collector file (SY\*\*) will never be deaccessed.

Example of short form reply to deaccess a file that was created in an earlier example:

FUNCTION? DF,FIL1

PURGE CATALOG

FUNCTION? PC

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password

CATALOG TO BE PURGED? cat-name\$password

The dollar sign is used only when the password is concatenated directly to a file or catalog name.

Example replies (user responses are underlined):

FUNCTION? PC

CATALOG STRUCTURE TO WORKING LEVEL?

/CAT1\$ABC

This response defines the subcatalog CAT1 concatenated to the user's own master catalog.

CATALOG TO BE PURGED? CAT2\$AOK

(The catalog and all catalogs and files subordinate to it are now purged.)

CATALOG TO BE PURGED?

is reissued.

Example of short form reply to purge a catalog that was created in an earlier example:

FUNCTION? PC,/CAT1\$ABC/CAT2\$AOK

PURGE FILE

FUNCTION? PF

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password

FILE TO BE PURGED? file name\$password

Password request will be issued if incorrectly given or omitted.

Example replies (user responses are underlined):

FUNCTION? PF

CATALOG STRUCTURE TO WORKING LEVEL?

JDOE/CAT1\$ABC/CAT2\$AOK

The user in this case is ALLONG, not the file creator.

FILE TO BE PURGED? FIL1

(The file is now purged.)

The request

FILE TO BE PURGED?

is reissued.

Example of short form reply to purge a file that was created in an earlier example:

FUNCTION? PF,JDOE/CAT1\$ABC/CAT2\$AOK/FIL1

RELEASE CATALOG

FUNCTION? RC

The question/response sequence and the short form reply for this function are completely analogous to those for the Purge Catalog function. The Release Catalog function would normally be used in preference to Purge Catalog -- as it is more economical -- unless the user has a very stringent file-security requirement.

RELEASE FILE

FUNCTION? RF

The question/response sequence and the short form reply for this function are completely analogous to those for the Purge File function. The Release File function would normally be used in preference to Purge File -- as it is more economical -- unless the user has a very stringent file-security requirement.

MODIFY CATALOG

FUNCTION? MC

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password

CATALOG TO BE MODIFIED? cat-name

NEW NAME? new cat-name

NEW PASSWORD? { new password }  
M#A#C#D#E#K#N#O#S { DELETE }

GENERAL PERMISSIONS? { access-type, ..., access-type }  
DELETE

SPECIFIC PERMISSION? { access-type, ..., access-type/  
user-ID.../user-ID }  
DELETE/user-ID/.../user-ID



Example replies (user responses are underlined):

FUNCTION? MC

CATALOG STRUCTURE TO WORKING LEVEL? CAT1\$ABC

CATALOG TO BE MODIFIED? CAT2\$AOK

NEW NAME?

A carriage return only response means that the catalog name is to remain unchanged.

NEW PASSWORD?

~~XXXXXXXXXXXX~~ (response "XYZ" given)

The original password AOK is replaced by XYZ.

GENERAL PERMISSIONS? READ

As originally created, general permissions were not assigned at this level. This response replaces this null set with READ and EXECUTE permission.

SPECIFIC PERMISSION? W/BJONES

This response replaces the original specific READ permission for BJONES with READ, WRITE, EXECUTE and APPEND permission.

SPECIFIC PERMISSION? DELETE/ASMITH

This response cancels any permissions for ASMITH that previously existed.

SPECIFIC PERMISSION? P,LOCK/ALLONG

This response replaces the original set of permissions for ALLONG with PURGE and LOCK.

SPECIFIC PERMISSION?

The carriage return implies that no further modifications are to be made; the changes are now processed and the question

CATALOG TO BE MODIFIED?

is reissued.

Special Short Form Option Formats

To rename a catalog:

NEWNAME/catalog/ or N/catalog/

To exclude, by user-ID, from any general permissions:

EXCLUDE/user-ID,..., user-ID/

To delete specific permissions, by user-ID:

DELETE/user-ID,...,user-ID/

To delete all general permissions:

DELETE/GEN'L/(or simply DELETE)

NOTE: EXCLUDE and DELETE may not be abbreviated.

Example of short form reply:

FUNCTION? MC,/CAT1\$ABC/CAT2\$AOK,PASSWORD/XYZ/,W/BJONES/,DELETE/ASMITH/,

MORE? P/ALLONG/,LOCK/ALLONG

#### MODIFY FILE

FUNCTION? MF

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name\$password/.../cat-name\$password/filename\$password

FILE TO BE MODIFIED? filename

NEW NAME? new filename

NEW MAX SIZE? new maximum size (in blocks)

NEW PASSWORD? { new password }  
~~XXXXXXXXXXXX~~ { DELETE }

GENERAL PERMISSIONS? { access-type,...,access-type }  
{ DELETE }

SPECIFIC PERMISSION? { access-type/user-ID/.../user-ID }  
{ DELETE/user-ID/.../user-ID }

Example replies (user responses are underlined):

FUNCTION? MF

CATALOG STRUCTURE TO WORKING LEVEL?

/CAT1\$ABC/CAT2\$XYZ

FILE TO BE MODIFIED? FIL1

NEW NAME? MASTER1

NEW MAX SIZE? 20

This response increases the maximum file size to 20 blocks (originally 12).

NEW PASSWORD?

~~XXXXXXXXXXXX~~ (response "DEPT37" given)

This response attaches the password DEPT37 (which would be in the strikeover area) to this file (none originally assigned).

GENERAL PERMISSION? DELETE

The original general READ permission is deleted.

SPECIFIC PERMISSION? P/BJONES

PURGE permission for user BJONES is added at this level. This permission applies to this file only.

Special short form option formats:

To rename a file:

NEWNAME/filename/or N/filename/

To exclude, by user-ID, from any general permissions:

EXCLUDE/user-ID,....,user-ID/

To delete, by user-ID, specific permissions:

DELETE/user-ID,....,user-ID/

To delete all general permissions:

DELETE/GEN'L/

or

DELETE

NOTE: EXCLUDE and DELETE may not be abbreviated.

To change the mode of a file:

MODE/mode/

Example of short form reply:

FUNCTION? MF,/CAT1\$ABC/CAT2\$XYZ/FIL1,N/

MORE? MASTER1/,B/20/,PASS/DEPT37/,DELETE,P/BJONES/

#### LIST CATALOG

FUNCTION? LC or LIST CATALOG

CATALOG STRUCTURE INCLUDING CATALOG TO BE LISTED?

user-ID/cat-name, ..., cat-name, n, x (mm-dd-yy), S, A, R, FIRST/name/

Example replies (user responses are underlined):

FUNCTION? LC

CATALOG STRUCTURE INCLUDING CATALOG TO BE LISTED?

/CAT1

Passwords need not be given in the catalog structure unless the catalog to be listed was created by another user. A user is permitted to list only catalogs of which he was the creator or the Library catalog (#LIB) or the command library catalog (#CMD) or catalogs belonging to other users for which he has modify permission.

A list of the catalogs and files emanating from CAT1 would now be listed.

The List Catalog provides selective listing of catalog and file names by the use of the optional parameters n, x (mm-dd-yy), S, R, A, FIRST/name/ where:

x, specifies whether date is date created (C), date of last access (A), or last date the file contents were changed (L)

mm-dd-yy, starting date for C, A, or L option

n, number of files to be listed

S, sort the names

R, reverse the order of printing

A, abbreviated list (eight per line)

FIRST/name/ starts the catalog listing at the specified cat/file name.

Any option may be omitted and the order in which they are given is immaterial.

Examples (user responses are underscored):

FUNCTION? LC

CATALOG STRUCTURE INCLUDING CATALOG TO BE LISTED?

catdescr,C(01-01-72)

Requests a list of all catalog and file names created since January 1, 1972.

catdescr,A(07-01-72),R

Requests a list of all catalog and file names which were accessed since July 1, 1972 and in reverse order (most recent to oldest).

catdescr,L(06-01-72),10

Requests a list of the first ten catalog and file names whose contents were changed since June 1, 1972.

catdescr,R,10

Requests a list of the ten most recently created catalog and file names.

catdescr,10

Requests a list of the ten oldest catalog and file names.

catdescr,FIRST/FILX/

Requests a list of catalog and file names starting at FILX.

#### LIST SPECIFIC

FUNCTION? LS

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name,...,cat-name(or)file name

CATALOG OR FILE TO BE LISTED?

Example replies (user responses are underlined):

FUNCTION? LS

CATALOG STRUCTURE TO WORKING LEVEL?

/CAT1

CATALOG OR FILE TO BE LISTED? FIL1

Passwords need not be given in the catalog structure unless the specified file or catalog was created by another user.

The description of FIL1 would now be listed.

The system will provide the following information (but not the password) about the catalog or file:

FILE NAME-  
ORIGINATOR-  
DATE CREATED-  
DATE CHANGED-(month/day/year plus T.O.D. in parentheses)  
LAST DATE ACCESSED-  
NUMBER OF ACCESSES-  
MAX FILE SIZE-  
CURRENT FILE SIZE-  
FILE TYPE-RANDOM, LINKED or I-D-S  
DEVICE-  
GENERAL PERMISSIONS-  
SPECIFIC PERMISSIONS-

In addition to a list of the user's own catalogs or files, a user may obtain a specific list of the library (#LIB), the command library (#CMD), or the catalogs or files belonging to other users for which he was the creator or has modify permission.

#### EXAMPLES OF LINE DELIMITER USE

The line delimiters can be used in several ways to either shorten the question/response sequence, or terminate a function at any given point.

Examples of the effect of different response terminations are as follows:

FUNCTION? CC

The carriage return alone implies a master catalog.

NEW CATALOG NAME? 001\*

Passwords or permissions are not wanted for this catalog and no further questions are wanted. Return is to NEW CATALOG NAME? level.

NEW CATALOG NAME? 002

PASSWORD?

~~XXXXXXXXXXXX~~ (PASS2\*\*)

No permissions are to be assigned to this catalog, and creation of catalogs at this position is finished. Return is to function level.

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL?

/002\$PASS2

FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE? 02.1,1,3

PASSWORD?

~~XXXXXXXXXXXX~~ (null response given)

GENERAL PERMISSIONS? READ

SPECIFIC PERMISSION? W/RJJONES\*\*

Creation of files at this level has been completed.

FUNCTION? carriage return (or DONE)

Finished with ACCESS.

\*

Return to the subsystem selection level.

#### SPECIAL FEATURES

Files created by means of the Create File function are not necessarily contiguous; i.e., successive links of a multilink file are not necessarily in physical sequence on the storage device. Furthermore, both the Create File and Access File functions assume that the file will be treated as a linked file. For the standard subsystems provided with the Time Sharing System, these file characteristics are suitable because linked files are most commonly used.

If, however, in the use of a given subsystem, it would be advantageous to have contiguous files, this characteristic can be specified in response to FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE?. The form of this response is:

filename,initial size C

The parameter C indicates, in Create File only, that a contiguous file is desired. No maximum size may be specified.

Similarly, if random treatment of files is required in a given user-written subsystem, a file can either be created as a random file or accessed as a random file. If created as such, it is always treated by the GCOS I/O Supervisor as a random file. If it is created as a linked file, it can be accessed as a random file, but in that case, the random treatment indication is temporary; i.e., it applies to that access only.

The forms of the random specification are as follows:

For CF, the response to FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE? is:

filename,initial size,maximum size,R

or

filename,initial size C,R

For Access File, the response to FILE NAME? is:

filename\$password,R

In both responses, the parameter R (always preceded by a comma) indicates that the named file is to be treated as a random file.

In the case of Create File only, the additional question LOGICAL RECORD SIZE? is asked, allowing the user to specify a fixed logical record size attribute as required of random files by TSS FORTRAN. If this attribute is not needed, the user may respond with simply a carriage return.

In the short form response, random files can be specified by:

MODE/RAND/ or MODE/R/

Linked files can be specified explicitly, either by:

MODE/LINKED/ or MODE/L/

or

MODE/SEQ/ or MODE/S/

or, more simply, by default.

Contiguity can also be specified in the short form response.



The following extended File System options may be exercised with nonconversational ACCESS functions CF or MF. For a detailed description of these options refer to the File Management Supervisor manual.

ABORT	/ { LOCK ROLLBACK <u>NONE</u> }	/	(What to do when writing job aborts?)
ACCESS	/ { <u>NORMAL</u> RWW CONCURRENT MONITOR }	/	(What to do when requested file is being written by another?)
VERIFY	/ { YES <u>NO</u> }	/	(Should writes be turned into write-verify instructions?)
AUDIT	/ { ALL DENIED <u>NONE</u> }	/	(What allocation requests to audit?)
INCRSAVE/	{ YES <u>NO</u> }	/	(Should file be saved if it has changed since last save?)
PAGESIZE/#WORDS/	40 ≤ # ≤ 3840 (320 assumed)		
RDERR/DUP/	(What to do to recover from read error?)		
WLOCK	(Write lock; allow only read allocations henceforth.)		
SLOCK	(Security lock the file or catalog.)		
NSLOCK	(Cancel security lock.)		
SAVE	(Save file content whenever perm save occurs.)		
NSAVE	(Do not save file content.)		

### RECOVERY SUBSYSTEM

The RECOVERY subsystem gives the terminal user the ability to make his collector file permanent and to catalog it under his User Master Catalog (UMC). Thus a user has the ability to recover his last input lines in any situation where an accidental or unexpected disconnect occurs.

The collector file will contain the lines of data entered via the terminal which have not yet been edited into the current file (see Definition, Section I, for definition of current file). The number of lines in the collector file may vary because of line length and the amount of data entered since the last edit of the collector file to the current file. In general, it will contain up to the last 70 lines. The RECOVERY subsystem, when used with the OLDP and NEWP functions provides the terminal user with the ability to recover the entire file when a disconnect occurs.

The terminal user can issue the RECOVERY command at any level; that is, at the subsystem selection level or at the build mode level.

The RECOVERY subsystem also permits the use of the ROLLBACK command to recover the collector file at the user's next terminal session. The ROLLBACK command can be issued only at the build mode level.

### Recovery Operation

In its basic operation, the RECOVERY subsystem dumps data currently on the temporary input collector file to the current file and creates and/or accesses a permanent file specified in the command (by filename) with an alternate name. If this permanent file already exists in the user's master catalog, the file is checked to assure that it conforms to the minimum requirements for an input collector file. The requirements for an input collector file are that the file must be a random permanent file and it must be at least 640 words (two blocks) long. A longer file will be accepted, but only 640 words will be used.

If the filename is not in the specified UMC, a file will be created and given predefined attributes. It will then be accessed by an alternate name. Accessing the file by the alternate name puts the alternate name in the Available File Table (AFT). The RECOVERY subsystem then switches the two file names; one representing the temporary input collector file and the other representing the permanent recovery file. Thus, all reference to the input collector file now points to the AFT entry describing the permanent recovery file. Even if this is not a random file, it will be accessed as a random file.

The procedure for termination, user log-off, or disconnect is the reverse of the procedure described above. The AFT will contain at least two entries (assuming that recovery was requested). The names associated with these entries will be switched and the files will be deallocated by the TERM module of the Time Sharing System.

When the terminal user issues the ROLLBACK command, the RECOVERY subsystem will again copy any data currently on the temporary input collector file to the current working file. It will then access the file specified in the command. When accessed, the permanent recovery file is read and any data in this file is also copied on the current working file. The first and last lines of good data on this file, preceded by an identifying message, is printed out on the terminal. Thus, when the user receives the RECOVERY NOW IN EFFECT message following a ROLLBACK command, he is ready to type into an empty 640-word collector file.

A terminal user may issue any number of RECOVERY and/or ROLLBACK commands during his session at the terminal. These commands must be given as #RECO and #ROLL if the user is in EDIT build mode. When subsequent commands are issued, the previous RECOVERY file is deaccessed, and a new RECOVERY file is created and/or accessed. The permanent file remains in the user's catalog until he specifically releases it.

If an error occurs during the creation and/or accessing of the new RECOVERY file, the terminal user will be working with a temporary input collector file and not his RECOVERY file. The data on the RECOVERY file may be unrecoverable (because of a missing end-of-file) if the terminal user tries to access this file through any other subsystem.

## QUESTIONS AND RESPONSES

The following paragraphs describe sets of questions and general responses associated with the RECOVER and ROLLBACK commands. In these descriptions, the general response to each question is underlined to set it apart.

```
*RECO FIL1$ABC  
RECOVERY NOW IN EFFECT
```

The RECOVERY subsystem is called to create and/or access FIL1 with a password ABC. The user is then returned to build mode.

```
*BASIC OLDP FIL2  
*RECO FIL3  
RECOVERY NOW IN EFFECT
```

The user has specified that FIL2 become his current working file. RECOVERY subsystem is then called to create and/or access FIL3.

```
*EDIT NEWP .SRC  
*#REC FIL4  
RECOVERY NOW IN EFFECT
```

The user requests the EDIT subsystem and a permanent current file. At the command level, the user calls for RECOVERY to create and/or access FIL4.

```
*BASIC NEWP TEXTFILE  
*10 PRINT  
*20 PRINT  
*RECO FIL5$BCA  
RECOVERY NOW IN EFFECT
```

The basic line Editor is called to sort and merge lines 10 and 20 onto the current working file. RECOVERY subsystem is then called to create and/or access FIL5 with a password BCA.

```
*BASIC NEWP SOURCE  
*RECO FIL6$CAB  
*10 PRINT "THIS IS LINE #10"  
*20 PRINT "THIS IS LINE #20"  
*30 PRINT "THIS IS LINE #30"  
*40 PRINT "THIS IS LINE #40"  
*50 PRINT "THIS IS LINE #50"  
*60 PRINT "THIS IS LINE #60"  
*70 PRINT "THIS IS LINE #70"
```

Assume that at this point the computer system disconnects. The user will do the following to recover his last input lines.

```
*BASIC OLDP SOURCE
*ROLL FIL6$CAB
FIRST AND LAST LINES OF SAVED DATA ARE:
10 PRINT "THIS IS LINE #10"
70 PRINT "THIS IS LINE #70"
RECOVERY NOW IN EFFECT
```

When the system is restarted after the disconnect, the user calls in the RECOVERY subsystem by issuing the ROLLBACK command. The RECOVERY subsystem will access FIL6 and sort and merge the data onto the current working file. When the RECOVERY NOW IN EFFECT message is issued, the user is ready to type into an empty collector file.

#### TIME SHARING MEDIA CONVERSION PROGRAM

The Time Sharing Media Conversion Program (TSCONV) is a batch program that may be run either at the central computer site or through a remote/batch (GRTS or NPS) terminal. It generates a standard format, time sharing text file from a suitable card deck, or conversely, produces a card deck from such a file, however generated, to save the file in card form.

#### Operational Description

The media conversion program performs the following functions:

- INPUT - create a standard format, time sharing text file from cards. If the INSERT or MOVE option is used, # signs are inserted between the line number and the first character of numeric data.
- OUTPUT - create a card deck from a standard format, time sharing text file. # signs between the line number and the text are deleted.

The control record (card) will be printed on the execution report.

INPUT identifies the control card requesting the file creating function and takes the following mutually exclusive options:

<u>Option</u>	<u>Result</u>
ASIS,i,j	The text file is generated from the input cards, from the columns specified by <u>i</u> to <u>j</u> . Standard columns (default option) for <u>i</u> to <u>j</u> are 1 to 80.
MOVE,i,j,m,n	The text file is generated from the input cards, from the columns specified by <u>i</u> to <u>j</u> . Line numbers are taken from columns specified by <u>m</u> to <u>n</u> . Standard columns for <u>i</u> to <u>j</u> are 1 to 72, and for <u>m</u> to <u>n</u> are 73 to 80.
INSERT,i,j,m,n	The text file is generated from the input cards and from the columns specified by <u>i</u> to <u>j</u> . Lines are sequence numbered, starting with <u>m</u> and incremented by <u>n</u> . Standard columns for <u>i</u> to <u>j</u> are 1 to 72. Standard values for both <u>m</u> and <u>n</u> are 10.
ASCII	The text file is generated from input cards, using a binary deck previously punched from this program.
COMDK,option	The text file is generated from input cards consisting of a COMDK (compressed source deck). This option is used in conjunction with the ASIS, MOVE, or INSERT options. If ALTER's are to be made at the time the file is generated, a \$ UPDATE card must be employed.
TAB,tab-char,pos-1,pos-2,...pos-n	The TAB specification must appear following the other activity options and separated from the other options by at least one blank. The tab character may be any single character except blank or reverse slant. A reverse slant "\ " followed by three digits is interpreted as the octal representation of the ASCII code for the desired tab character. The TAB specification is terminated by the first blank encountered. The tab positions specified must increase in ascending order. TAB supplied with the ASCII option has no meaning, but it is checked for correctness. Any error encountered in tab specification analysis results in a TB abort.

#### Sample INPUT Control Cards

```
INPUT,MOVE,1,60,73,80
```

Text file data is to be taken from columns 1 to 60 of the punched cards and line numbers are to be taken from columns 73 to 80.

```
INPUT,COMDK,ASIS,1,80
```

Text file data is to be taken from columns 1 to 80 of the input cards (a COMDK).

INPUT can start in any column of the control card but no imbedded blanks are allowed.

OUTPUT identifies the control card requesting the card deck producing function, and takes the following mutually exclusive options:

<u>Option</u>	<u>Result</u>
ASIS, <u>i</u> , <u>j</u>	The text file is read and a BCD card deck is punched in the columns specified by <u>i</u> to <u>j</u> . Standard columns (default option) for <u>i</u> to <u>j</u> are 1 to 80.
MOVE, <u>i</u> , <u>j</u> , <u>m</u> , <u>n</u> , <u>l</u>	The text file is read and a BCD card deck is punched, moving data to columns specified by <u>i</u> to <u>j</u> . Line numbers are moved to columns specified by <u>m</u> to <u>n</u> , right-justified. The <u>l</u> specifies the label to be punched starting in column 73, left-justified. Standard columns for <u>i</u> to <u>j</u> are 1 to 72 and for <u>m</u> to <u>n</u> , 73 to 80.
STRIP, <u>i</u> , <u>j</u>	The text file is read and a card deck is punched, stripping off line numbers, with data moved to the columns specified by <u>i</u> to <u>j</u> . Standard columns for <u>i</u> to <u>j</u> are 1 to 80.
	NOTE: With the above output options, data is converted from ASCII to BCD before punching.
ASCII	The text file is read and a binary deck containing the file text is punched. (See "Binary Card Format" below.)
TAB	See options for INPUT.

#### Sample OUTPUT Control Card

```
OUTPUT,ASIS,1,56
```

The text file is punched into columns 1 to 56 of the card deck.

OUTPUT can start in any column of the control card but no imbedded blanks are allowed.

#### Definitions

- Each line is punched on a separate card, starting in the column specified (OUTPUT function).
- A line number is an initial string of numeric characters which terminate with a nonnumeric character. Blank is considered a nonnumeric character.
- In the case of the MOVE option, the line numbers are stored right-justified in the columns specified.
- A reverse slash in input processing is treated as a line separator and is replace with a carriage return.

- The format of a line in a text file is media code 6.

### Errors

SE ABORT - A binary card is out of sequence.  
Card number is printed out.

CK ABORT - Checksum of card does not agree with the computed checksum.

NB ABORT - First data card is not binary, but ASCII was specified on control card.

CP ABORT - No control card found (keyword may be misspelled).

TB ABORT - TAB specification error.

DATA LINE TOO LONG FOR I,J FIELD

...portion of the line specified by i to j...

- Occurs on OUTPUT only. If a line of the file is too long for the specified i to j field (i.e., nonblank characters are being discarded), this warning message is issued along with the portion of the line specified by i, j. A maximum of 20 such messages may be given. The complete file is punched, as specified by the i to j field options.

### Binary Card Format

Word 1	7/9 punch and number of data words (maximum=21)
Word 2	Checksum
Word 3	Card number, starting at 0
Words 4-24	Text

Sample Deck Setups

A sample deck setup to accomplish media conversion is as follows:

```
$ SNUMB XXXXX
$ IDENT account number,name
$ USERID name$password
$ PROGRAM TSCONV
$ PRMFL OT,R/W,L,userid/filename
INPUT,ASIS
```

(Data deck)

```
$ ENDJOB
***EOF
```

*CREATE TSS file  
from cards*

A sample deck setup to accomplish media conversion in the case of a COMDK plus ALTER cards is as follows:

```
$ SNUMB XXXXX
$ IDENT account number
$ USERID name$password
$ PROGRAM TSCONV
$ PRMFL OT,R/W,L,userid/filename
$ DATA I*, ,COPY
INPUT,COMDK,ASIS,1,80
```

(Data cards -- COMDK)

```
$ ENDCOPY
$ UPDATE
```

(ALTER deck)

```
$ ENDJOB
***EOF
```

The following is a sample deck setup for an OUTPUT run:

```
$ SNUMB XXXXX
$ IDENT account number,name
$ USERID name$password
$ PROGRAM TSCONV
$ PRMFL OT,R/W,L,userid/filename
OUTPUT,ASIS
$ ENDJOB
***EOF
```



## SECTION VI

### DEFERRED PROCESSING

#### INTRODUCTION

The term "deferred processing" implies a planned time sharing session scheduled by a user to be independently initiated at some given date and time. Since the user does not actively participate in his dialog exchange with the computer during the session, he must anticipate his responses and provide an input file containing them in the order to be presented. An output file must also be provided to collect the exchanged dialog. A listing of this file produced after the deferred session terminates will appear notably similar to the log of an online user.

The command used to schedule a deferred session is DRUN. Upon receipt of this command, the user's input and output file descriptions are obtained, in addition to any optional parameter declarations. The job is then assigned a unique identifier (nnnnD) and recorded in a special deferred queue file with a status indicating that it is scheduled for initiation. After establishing the request for deferred processing, the user may continue his online session or disconnect if desired. When the job becomes eligible for initiation, it is logged on with the user-id of its originator and the status of the deferred queue entry is changed to indicate the session is in progress. When termination occurs, the status is again updated to reflect a normal or abnormal ending and the deferred session is concluded.

Supporting commands DSTS and DABT provide the user with the ability to make status inquiries regarding deferred jobs or abort jobs that have not yet reached termination. The COUT, CMOD and CPOS commands (see Section VII) may be used within the deferred session. COUT permits all output accumulated since either the beginning of the session or a previous COUT to be directed to a specified file. CMOD allows modification of the user's program switch word via arithmetic or Boolean operations, while CPOS permits conditional alteration of the normal sequential processing of input file commands, based on the contents of the switch word.

#### USER INTERFACE

The DRUN command is issued with the following syntax:

```
DRUN filedescr-in;filedescr-out;option-1;option-2;...;option-n
```

If no parameters accompany the command when it is issued, the user is asked "FILE NAMES?" and must, at this time, enter all information necessary to schedule the deferred session. Input and output file descriptions are required and must constitute the first two parameter declarations. Optional parameters may follow the output file description in any order desired.

Preparation of the input file is essential prior to scheduling a deferred session. This file may optionally have line numbers, the presence of which is established by the occurrence of a digit (0-9) as the first nonblank character of the first line. Line numbers serve no functional purpose for deferred processing other than allowing the user to specify only a segment (line number interval) of the file to be processed. If present, they are sequence checked and otherwise discarded. Note that as with CARDIN, the pound sign (#) may be used to separate the line number from the text when the latter begins with a numeric character. Double pound signs (##) are required when the first textual character is itself a #.

DRUN parameters may optionally be provided on the first line of the input file (or input file segment), consistent with the following syntax:

```
##filedescr-out;option-1;option-2;...;option-n
```

The output file description is required and must constitute the first parameter of the line. If not provided, it must be declared explicitly null, as follows:

```
##;option-1;option-2;...;option-n
```

Parameter declarations supplied with the DRUN command override similar declarations on the ## line. Thus, when an output file description is included in both the DRUN command line and the ## line of the input file, the latter declaration is ignored.

The following example illustrates the input file for a deferred session that results in spawning a CARDIN job:

```
010##OTFILE
020CARDIN
030OLD PERMSAVE
040RUN
050N(;)
060BYE
```

Line 010: Since this is the first line of the file, the ## identifies it as containing DRUN parameters. In this case, the only parameter given is the name of the output file.

Line 020: CARDIN is the response to the initial \* query. Note that user-id and password are not included in the file for the logon sequence. This function is implicitly performed for the deferred session.

Line 030: File PERMSAVE becomes the current file with this use of the OLD command.

Line 040: The deferred session is now in build mode on behalf of the CARDIN subsystem selection. The RUN command initiates the batch job.

Line 050: N(;) is the response to the question, "CARD FORMAT,DISPOSITION?", issued by the CARDIN RUN subsystem.

Line 060: The BYE command terminates the deferred session. This may be omitted when the user does not care to have the logoff statistics included in the output file.

A listing of file "OTFILE" produced when the deferred session terminates might appear as follows:

```
HIS SERIES 6000 ON 04/25/77 AT 11.583 DEFERRED # 6772D
```

```
USER ID-JOHNDOE
```

```
*CARDIN
```

```
*OLD PERMSAVE
```

```
*RUN
```

```
  SNUMB # 2367T
```

```
CARD FORMAT,DISPOSITION ?
```

```
N(;)
```

```
*BYE
```

```
**COST: $ 0.22 TO DATE: $ 316.53= 32%
```

```
**ON AT 11.583 - OFF AT 11.589 ON 04/25/77
```

The input and output file descriptions declared for a deferred session must conform to the conventional format discussed in Section II, File Designation. These files must be permanent and up to three levels of subcatalogs may be specified for each. This limit is imposed by the space available on the deferred queue file and, when substitutable arguments are declared, is further reduced to a combined total of four subcatalog levels. All passwords necessary to allocate either file must accompany its description. A line number interval may optionally be declared for the input file and either description may include an alternate name and/or permissions.

Continuation input is requested for a file description (or the next ## line of the input file obtained) when the input line is prematurely terminated with a slash, dollar sign, comma, left parenthesis or the leading quote of an alternate name declaration. The semicolon, used to separate parameters from one another, always implies continuation when it occurs as the last character of the line.

A limited number of user responses required for a deferred session may be substituted for the input file description on the DRUN command line. All such responses must be enclosed in quotes and separated from one another by reverse slants (\) or ampersands (&). Thus, the deferred session illustrated in the previous example could be requested as follows:

```
DRUN "##OTFILE\CARDIN\OLD PERMSAVE\RUN\N(;)\BYE"
```

or equivalently,

```
DRUN "CARDIN\OLD PERMSAVE\RUN\N(;)\BYE";OTFILE
```

The quoted string may be prematurely terminated anywhere, causing continuation lines to be requested until the occurrence of the end quote. Maximum string length is not easily predictable and is a function of the presence of substitutable arguments, the number of subcatalogs qualifying the output file, and the number of individual responses occurring in the string. In general, two or three lines can usually be accommodated. Note that two consecutive line delimiters or a line delimiter immediately preceding the terminating quote represents the equivalent of a null response. A quote, reverse slant or ampersand may be used as text in the character string by preceding it with an ESC (escape) character.

The output file for a deferred session is not utilized until the session terminates, whereupon its allocation is attempted in accordance with the file description declared when the session was scheduled. If the file is found to be nonexistent, it is created for the user, provided (1) an alternate name was not specified, or (2) requested permissions, if present, include only a subset of R/W/A/E. Permissions and/or a password may optionally be attached to the file when it is created by including them in the description. A special output file declaration, \*NULL, may be provided if the user does not want to save his output.

The output file name declaration \*ID may optionally be specified with the DRUN command, causing the job identifier (nnnnD) to be used for the file name. This declaration is also applicable for CRUN and CPOS, provided the command is executed in a deferred processing environment. \*ID may be qualified by subcatalogs and permissions with which to create the file may accompany the description.

Example: \*DRUN INFIL;/CAT1/\*ID

Optional parameters that may accompany a DRUN request include the following:

- Earliest Session Initiation Date

Form: YY/MM/DD, YY-MM-DD, MM/DD/YY or MM-DD-YY

This declaration indicates the earliest date on which the deferred session can be initiated. YY represents the last two digits of the year, MM is the month and DD is the day. MM or DD may consist of either one or two digits. If not specified, the current date is assumed.

- Earliest Session Initiation Time

Form: HH:MM or HH.TTT

This declaration indicates the earliest time of day on which the deferred session can be initiated. HH represents the hour (0 HH 24) and may consist of one or two digits, while MM indicates the minute within the hour and must constitute two digits. TTT is a 1-3 digit fractional hour specification. If not declared, the session will either be initiated at the earliest possible time on the target date or at a preferred time on that date optionally specified by the site.

## Maximum Processor Time Limit

Form: nn...n

This declaration, consisting of an integer representing seconds, limits the total processor time permitted for the deferred session. The site itself can impose such a limit for all deferred sessions, overriding the user's declaration if the latter is larger. If neither site nor user imposes this limit, the deferred session is allowed to run for an unlimited period of time.

## Substitutable Arguments

Form: (arg-1,arg-2,...,arg-n) or (arg-1;arg-2;...;arg-n)

Substitutable arguments provide a means for having the character string implied by the i-th argument declaration substituted for any occurrence of the characters #i appearing in the input file (the input file itself is not modified). Up to eight arguments may be specified, each consisting of 1-12 characters. Permissible characters include alphanumerics, the period, dash, colon, slash, dollar sign, pound sign and asterisk. Blanks included in a character string are retained and all alphabets are forced uppercase.

In some instances it may be necessary to preserve the case of alphabets and/or utilize characters not otherwise permitted for an argument. For such requirements the argument may be enclosed by quotation marks and constitute up to ten characters. An argument declared in this manner is unrestricted with respect to content (it may contain any ASCII characters) and is substituted exactly as it appears.

Example: DRUN INFIL;OTFIL;("/CAT/FIL,R","(10,100)")

Individual substitutable arguments declared on the DRUN command line normally override the corresponding argument declaration on the ## line of the input file; i.e., the latter declaration is ignored. When override is not desired, the specific command line argument may be declared null by entering two consecutive commas. Note that the number of arguments specified on the command line may differ from the number specified on the ## line of the input file. Missing arguments of the shorter list are treated as if they were explicitly declared null. The number of arguments substituted corresponds to the larger list.

When a question mark (?) occurs anywhere in a nonquoted substitutable argument, the entire argument string is issued to the user, representing a request to enter the actual value of that argument. Thus, the argument list, (YFOR,FILENAME?,LIST-OR-RUN?), declared for a DRUN results in issuing the questions, "FILENAME?" and "LIST-OR-RUN?" to the user for obtaining the actual values of arguments #2 and #3. A quoted string (as described above) may be entered if desired.

Certain special arguments representing specific information may be declared, as given in the following table:

<u>Arg</u>	<u>Implies Substitution of</u>	<u>Format</u>
#DATE	Current date	YY/MM/DD
#TIME	Current time of day	TT.TTT or T.TTT
#USERID	Logon user-id	XXX...X
#ACCOUNT	User account number	XXX...X
#CHANNEL	Channel number	NNNN
#SNUMB	Last snumb # generated	NNNNT

The character string which is substituted for such an argument declaration corresponds to the value of that argument when the DRUN command is issued. Thus, for example, the argument #CHANNEL equates to the channel number associated with the user who scheduled the deferred session. The special arguments may be abbreviated by the first character of their name. For example, #D is equivalent to #DATE. The following input file illustrates use of the #ACCOUNT argument to establish the same account number for the deferred session as was in use when the session was scheduled:

```
##OTFILE;(#A)
NEWU #1
CARD OLD PERMSAVE
RUN
N(;)
BYE
```

A series of 1-6 pound signs (##...#) may be declared for any substitutable argument and implies the substitution of a corresponding number of digits obtained by converting the lower half of the user's Program Switch Word (PSW) to decimal. When fewer than six pound signs are specified, leading digits of the converted PSW are discarded; i.e., the value substituted consists of the least significant digits. Leading zeros are included, when necessary.

The lower half of the PSW for a deferred session is initially set to the corresponding PSW value of the user who scheduled the session. This provides, among other things, the ability of a deferred session to schedule itself again at a later time or date. The input file (named INFILE) for the deferred session illustrated in the following example results in spawning our familiar CARDIN job on a specified day at 10:00 P.M. and rescheduling itself to be initiated the next day at the same time, continuing until (and including) a specified termination day. The example assumes the lower half of the PSW is initially zero and uses substitutable arguments to request starting date, ending date, month and year for the deferred sessions. Note that an immediate session is initiated to schedule the first requisite DRUN.

```

##*NULL; (FIRST-DAY?,LAST-DAY?,MONTH?,
##YEAR?,##)
CPOS NEO;+4
CMOD #1;+1
DRUN INFILE;OTFILE;#4/#3/#1;22:00;(#1,#2,#3,#4)
BYE
CARDIN
OLD PERMSAVE
RUN
N(;)
CPOS GT#2;+3
CMOD +1
DRUN INFILE;OTFILE;#4/#3/#5;22:00;(#1,#2,#3,#4)
BYE

```

- Substitution-Implying Character

Form: ARG/c

The characters #i are normally used to request substitution of the i-th argument. The ARG/c declaration permits the user to specify any character (c) other than # to denote argument substitution. This is necessary, for example, when the pound sign occurs as text in the input file.

- Delete Character

Form: DEL/c

This declaration requests deletion of all occurrences of the specified character (c) appearing in the input file. Its use is primarily intended for preparing null response lines. Such a line might consist solely of the declared character. Note that an implied (default) delete character is not provided.

- Output File User Response Disposition

Form: <sup>Default</sup>  
INCLUDE or EXCLUDE

The output file produced for a deferred session normally includes the user's responses as they appear on the input file; i.e., the INCLUDE option is implied. The EXCLUDE option may be declared when the user wishes to eliminate all lines containing responses on the output file. Abbreviations INC or EXC are permitted.

- Output File Case

Form: UPPER or LOWER

Alphabetic text generated on the output file may be forced to upper or lowercase by the use of the appropriate option. If neither option is exercised, the case of alphabetic text is preserved. Abbreviations U and L are permitted.

• Deferred Session Restart

Form: RESTART

A system interruption occurring while a deferred session is in progress normally results in marking the session aborted when Time Sharing is restarted. The RESTART option, which may be abbreviated RES, requests the session to be reinitiated from the beginning if an interruption occurs.

Deferred sessions originated by a user which are scheduled to run at some later date and/or time may be rescheduled to run as soon as possible by utilizing the DRUN Command with the following syntax:

```
DRUN #job-id-1;job-id-2;...;job-id-n
```

Alternatively, the user may request all of his sessions scheduled for the current date which did not specify a specific start time to be initiated by issuing the following directive:

```
DRUN #ALL
```

The DSTS and DABT commands utilize similar syntax, except the pound sign (#) is omitted. DSTS permits status inquiries for deferred sessions originated by a user. The status message received is one of the following:

```
nnnnD - SCHEDULED TO RUN YYMMDD AT TT.TTT
```

```
nnnnD - RESCHEDULED TO RUN YYMMDD AT TT.TTT
```

```
nnnnD - EXECUTING
```

```
nnnnD - TERMINATING
```

```
nnnnD - ABORTED BY DABT YYMMDD AT TT.TTT
```

```
nnnnD - TERMINATED NORMALLY YYMMDD AT TT.TTT
```

```
nnnnD - ABORTED YYMMDD AT TT.TTT FOR REASON: (reason text on next line)
```

```
nnnnD - ABORTED DUE TO SYSTEM INTERRUPTION
```

The acronym ASAP (As Soon As Possible) is substituted for date and time in the status message for a scheduled or rescheduled job when it is overdue for initiation.

The DABT command may be utilized to abort one or more deferred sessions which have not yet reached termination. An "ALL" request results in aborting all jobs scheduled by the user, without regard to the specified starting date and/or time. Those jobs which are neither scheduled nor executing are ignored.

The parameter "REMO" is recognized by the DABT command processor as a request to remove all of the user's DRUN jobs on the deferred queue file which have terminated or been aborted. This feature is useful in reducing the output received with a "DSTS ALL" after many deferred jobs have been run.



Job identifiers declared with the DRUN, DSTS or DABT commands must belong to the requesting user. When the identifiers do not accompany the DSTS or DABT command on the same line, the query, "JOB ID?" is issued, at which time the identifier(s) must be declared. Continuation input is requested when the input line ends with a semicolon.

#### FUNCTIONAL SUMMARY

A deferred session is logically divided into four stages of processing--scheduling, initiation, execution and termination. Scheduling occurs upon receipt of the DRUN command and consists of logging the request on the deferred queue file. The queue file, cataloged in the GCOS3 UMC, serves as a collection medium for all information necessary to later initiate the deferred session. Such information includes the job identifier (nnnnD) assigned for the session, user-id, starting date/time, processor time limit, input and output file descriptions and all substitutable arguments declared.

When a job is logged on the queue file, the next candidate for initiation is determined by the DRUN Subsystem and the TSS Executive is notified of its job identifier, user-id and earliest starting date/time. Note that due to TSS load conditions, it may not be possible to initiate the session at the exact prescribed time; however, when conditions permit, a UST (User Status Table) is developed for the session and the DRUN Subsystem is invoked with an indication that the deferred session is to begin. At this time, all information relevant to the session is retrieved from the queue file and a "command file" is prepared as for the CRUN command.

Preparation of the command file involves creating a random temporary file named \*CFP and copying the input file to it, one line per hardware sector. Initial ## parameter lines are ignored, since they were processed when the session was scheduled. When requested, argument substitution and removal of DEL character occurrences takes place during the copy; however, this is the extent of input file content analysis and no guarantee is made regarding legitimate command constructs or syntax. An initial and final sector is generated on \*CFP, bordering the sectors developed from lines of the input file. The initial sector contains the user-id of the session originator and is utilized by the logon subsystem. The final sector contains a COUT command, accompanied with all information necessary to produce the output file when the session terminates.

After marking the status of the deferred queue entry to indicate the session is executing and notifying the TSS Executive of the next candidate for initiation, command file mode is enabled and control passed to the logon subsystem for user validation. During command file processing, the TSS Executive obtains each required user response by reading the appropriate \*CFP sector and directs all generated output to \*CFP starting at the sector immediately following the generated COUT command.

Normal termination occurs when a BYE or the generated COUT command is encountered at system selection level or while in build mode. In either case, the DRUN Subsystem is invoked to produce the output file and change the queue file status to indicate the deferred session has terminated. The output file description is obtained from the \*CFP sector containing the generated COUT command, after which the file is created (if necessary) and allocated. If the EXCLUDE option was declared, the output segment only of \*CFP is read, formatted and written to the user's output file. Otherwise, input and output segments are collated to produce the file. Collation is made possible by referencing the sector number of each user response in the appropriate sector of the \*CFP output segment.

Abnormal termination can occur for a variety of error conditions detected by either the TSS Executive or the DRUN Subsystem. When possible, the output file for the aborted session will be produced and in all cases the reason message for the abort is saved on the queue file for DSTS inquiry purposes.

## SECTION VII

### COMMAND FILE PROCESSING

#### INTRODUCTION

Command file processing is a non-interactive mode of online Time Sharing utilization, during which user responses for terminal input requests are obtained from a file. The dialog exchanged between user and computer for a command file application may optionally be directed to the user's remote device, or collected on a file for later perusal.

Command file processing is initiated upon receipt of the CRUN command. CRUN parameter declarations and syntax conventions are nearly identical to DRUN--the command used to schedule a deferred session. The execution phase of such a session constitutes a command file application. Deferred processing is discussed at length in Section VI. It is essential to review this material first, as it contains instructions for input file preparation and describes the functional characteristics of command file processing.

#### USER INTERFACE

Command file processing is initiated on behalf of the requesting user when the CRUN command is issued, consistent with the following syntax:

```
CRUN filedescr-in;filedesc-out;option-1;option-2;...;option-n
```

When no parameters accompany the command, the user is asked "FILE NAMES?: and must, at this time, provide all information necessary to initiate the CRUN application. An input file description is required and must constitute the first parameter declaration. As with DRUN, a limited number of user responses may be substituted for the description, all of which must be enclosed by quotes and separated from one another by reverse slants or ampersands. When the input is specified in this manner it is permissible to omit the CRUN command itself.

Example: \*"CATA";\*

The EXCLUDE option is the default case for this type of entry.

The output file description is optional, but when present must constitute the second parameter. When only an input file description is specified, or is followed by two consecutive semicolons, all keyboard output generated for the command file application is directed to the user's remote device upon completion of the application. In a deferred processing environment, all such output destined for the remote device is discarded. Output file management and description variations are similar to DRUN conventions, with the exception that a temporary file is created for the user when (1) the named file does not already exist, and (2) the description consists solely of a file name. The current file may be declared for either the input or output description (both, if desired) and is denoted by an asterisk (\*). Other optional CRUN parameters include the following:

<u>FORM</u>	<u>MEANING</u>
nn...n	Maximum processor time limit
(arg-1,arg-2,...arg-n)	Substitutable arguments
ARG/c	Substitution-implying character
DEL/c	Delete character
INCLUDE or EXCLUDE	Output file user response disposition
UPPER or LOWER	Output file case

These parameters, when specified, must follow the output file description and may occur in any order. The function of each is as described for DRUN.

The input file for a CRUN application may optionally contain line numbers and/or first line parameter declarations. Parameters unique to deferred processing applications are ignored when encountered as such first line declarations. This relaxation permits the same input file to be utilized for either a command file or deferred processing application.

The presence of line numbers on the input file is determined by the occurrence of a digit (0-9) as the first nonblank character of the first line. When an input file without line numbers otherwise begins with a numeric, an initial line may be introduced containing only the characters, ##; e.g.,

```
##
100 DIMENSION DATE(2)
200 CALL DATIM(DATE,TIME)
300 PRINT 100,DATE,TIME
400 100 FORMAT("0DATE: ",2A4," - TIME:",F7.3)
500 STOP;END
RUN
```

The input file illustrated above is shown merely for the sake of example and would have limited utility for a CRUN application, since it assumes the user is in build mode on behalf of the FORTRAN system selection and expects the current file to be in an initialized state. These shortcomings, in addition to the ## line requirement, can be overcome by replacing the ## line with the command:

```
*FORT NEW
```

The response of FORT NEW results in establishing FORTRAN as the current system selection with an empty current file.

Various special response lines are recognized by the TSS Executive when command file processing is in progress. Such lines are identified by the prefix, \$\$, immediately followed by one of the commands:

TALK	NULL
FILE	COPY
USER	DELE
QUIT	LBL
TRAP	BRK
MARK	
REM	

\$\$ response lines may appear anywhere in the input file; i.e., they are not restricted to recognition at system selection or build mode levels. \$\$TALK permits the user to become interactive with his command file application. Upon receipt of this command, the last buffer of output generated on \*CFP (normally the last line requesting an input response) is issued to the user's remote device. From this time until receipt of a \$\$FILE command, all dialog exchanged between user and computer is directed to both the remote device and \*CFP. Normal command file processing is resumed when \$\$FILE is issued as the response for any input request. The actual response for this request is obtained from the response line on \*CFP following the \$\$TALK response line.

An interesting application of \$\$TALK is to obtain a printer listing of the log produced by a Time Sharing session. This is easily accomplished by initiating a CRUN of the following input file at the beginning of the session:

```
##*NULL;DEL/  
$$TALK  
COUT TEMPF  
JPRINT TEMPF:IDENT(M26KLC554,RGHASEN,STATION G)
```

The printer listing is produced when the \$\$FILE command is given.

\$\$USER is a special form of \$\$TALK, allowing a single response line to be supplied by the user. As with \$\$TALK, the last line requesting an input response is issued to the user's remote device; however, upon receipt of the response, command file processing is automatically resumed.

The \$\$QUIT response line may appear anywhere on the input file or may be given as an interactive response. Its purpose is to immediately discontinue command file processing. The output file is not produced, nor is \*CFP removed from the AFT.

In some instances it may be desirable to have a specific input request issued to the user's remote device upon receipt of a \$\$TALK, \$\$USER or \$\$QUIT response line. This may be accomplished by affixing an equal sign (=) to the \$\$ function, followed by the desired prompt message. For example, the response line,

```
$$USER=WHAT IS YOUR ACCOUNT NUMBER?
```

results in issuing the question, "WHAT IS YOUR ACCOUNT NUMBER?" to the user's remote device. The output file is not affected by this feature and will contain the true prompt.

A command file or deferred processing application is immediately aborted upon the occurrence of any error or exception condition detected by a subsystem. In some instances, this action may be undesirable; e.g., use of a certain command might knowingly cause what appears to be an error, but has no effect whatsoever on the application. The \$\$TRAP ON (ON is optional) and \$\$TRAP OFF response lines provide a means of enabling and disabling this feature at will. All applications are originally initiated with the trap mode enabled; however, nested CRUNs are initiated with whatever mode is currently in effect.

When the trap mode is disabled, the user can determine the success or failure of the command executed with the previous response line by testing bit 13 of the program switch word with the CPOS command. If the previous command was successfully executed, bit 13 will be reset (off); otherwise it will be set (on). Thus, in the following example, the JRN command is not issued if the required file fails to become the current file.

```
*NULL;(FILENAME?)
$$TRAP OFF
OLD1
CPOS 13;+2
JRN
```

Note that it is necessary to examine the bit immediately following the command in question; i.e., other response lines cannot intervene between the command and the CPOS used to test its outcome.

The `$$NSYS` response line is the inverse of `$$SYST`, causing a system level synchronization error when encountered at system selection level.

When the `$$MARK` response line is encountered during a command file application, the message text starting in character position 9 is issued to the user's remote device. The MARK function can appear anywhere on the input file and may be used to notify the user when specific stages of processing are reached. The function is ignored when encountered in a deferred session.

The following example utilizes the function to display the snumb of the last batch job submitted by the user:

```
##;EXC;(#S)
$$MARK LAST JOB SUBMITTED WAS #1
```

The `$$REM` function provides a means of entering commentary notes on the input file. Since it is not included as a response line on `*CFP`, care must be exercised in calculating the relative offset for CPOS repositioning; i.e., all `$$REM` lines must be ignored for such calculations.

The `$$NULL` response line is the equivalent of a line containing a null response (CR only).

`$$COPY` causes the Time Sharing Executive to send (to the user's terminal) a copy of all input data read and all output produced by the commands. The function is deactivated by the directive, `$$COPY OFF`.

`$$DELETE` causes the Time Sharing Executive to refrain from writing output to the `*CFP` file during command file execution. Output deleted by this feature is irrevocably lost. The function is deactivated by the directive, `$$DELETE OFF`.

`$$LBL` defines a label (associated with the following line) which may be referenced as a transfer point with the CPOS command. A maximum of eight labels may be defined, each of which must consist of one to nine alphanumeric characters, including the period and dash. Commentary information may be included on a `$$LBL` line, separated from the label declaration by a colon or semicolon. A label referenced with the CPOS command must constitute the operation subfield (second parameter) of the command and be preceded by a dollar sign (\$) .

The following examples are thus functionally equivalent in all respects:

Example 1:

```
CMOD 0
$$LBL REPEAT-IT
LIST
CMOD +1
CPOS LE5; $REPEAT-IT
```

*Label*  
*CPOS*  
*condition; consequence*  
*Go to consequence*  
*either a LBL or numeric line decrement*

```
IF TRUE GOTO LBL
```

Example 2:

```
CMOD 0
LIST
CMOD +1
CPOS LE5;-2
```

Two special labels, `..BREAK` and `..ABORT`, may optionally be defined with the `$$LBL` function. Each time the break key is depressed during execution of a CRUN application, control is passed to the response line associated with the `..BREAK` label, provided it has been defined. Similarly, the occurrence of any error in either a CRUN or DRUN application (which will result in abnormal termination of the application) causes control to be passed to the response line associated with the `..ABORT` label. When termination occurs, or if a second error is encountered, control is either passed to the `..ABORT` procedure of the previous level of processing (in the case of nested CRUNs), or the application is terminated. The reason code (1-16) for the abort is available to the user's `..ABORT` procedure and may be materialized in the lower half of the Program Switch Word via the "ERROR" (or more simply "E") option of the CMOD command; e.g.,

```
$$LBL ..ABORT : COME HERE IF ABORT OCCURS
CMOD E
$$REM DIRECT OUTPUT TO *SRC IF ERROR CODE $\geq$ 6
CPOS GE6;$$SKIP
COUT *
$$LBL SKIP
```

Error codes (decimal) and their associated meaning are as follows:

<u>Code#</u>	<u>Meaning</u>
1	COMMAND FILE NONEXISTENT
2	COMMAND FILE I/O ERROR
3	COMMAND FILE FORMAT ERROR
4	EXCESSIVE OUTPUT GENERATED
5	INVALID USE OF DRL T.CFIO
6	USER ERROR #nnn DETECTED BY TSS
7	PPT START OR LINE SWITCH ATTEMPTED
10	PROCESSOR TIME LIMIT EXCEEDED
11	INPUT LINE LENGTH TOO LONG



- 12       COMMAND FILE INPUT EXHAUSTED
- 13       TERMINATED BY DABT WHILE EXECUTING
- 14       ERROR DETECTED IN SSname   PROCESSING
- 15       SYSTEM LEVEL SYNCHRONIZATION ERROR
- 16       \*\$FUNCTION NOT PERMITTED

The error numbers referenced in code 6 text are listed in Appendix A. The SSname in code 14 represents the name of the subsystem in execution at the time the error was detected.

The response line `*$BRK` may appear anywhere in the input file for a command file or deferred processing application and has the effect of simulating a break when the response line is processed. The break will not terminate the CRUN or DRUN application, but rather will behave as though the break key was depressed during a live session.

Command File processing with VIP devices is permitted; however, the response lines `*$COPY`, `*$USER`, `*$TALK`, `*$MARK`, and `*$QUIT` cannot be utilized.

Nested CRUNs may occur in a command file or deferred processing application to a maximum depth of four levels. The effect of encountering a CRUN during such an application is to suspend (pushdown) its processing until the new requested command file application terminates, whereupon processing of the previous application is resumed (popped up). When an error occurs during nested command file processing, output accumulated for all levels is generated and the entire application terminated. Note that each level of nested processing requires a command file in the user's AFT. Such files are identified by names `*CFP` through `*CFT` and care must be exercised to avoid AFT removal during application. NEWUSER processing preserves these files for the new user, thus allowing a command file application to change user-ids at will. Both user-id and password must follow a NEWUSER command on the CRUN input file. The following example illustrates a command file application to change user-ids with no intervening output issued:

```
*CRUN "NEWU\JDOE\JDPASSWD";*NULL
```

A break or disconnect (via DRL DRLDSC) may be used to prematurely terminate a command file application. Output generated for the application up to the point of interruption is directed to the requisite file(s), or discarded when destined for the user's remote device.

#### COMMAND LOADER INTERFACE

The Command Loader Subsystem, discussed in Section V of the TSS System Programmer's Reference Manual, may be utilized to initiate a command file application. This subsystem is invoked upon receipt of any unrecognized command. Such a "command" is construed to be a file description conforming to one of the following conventions:

- catalog/filename
- /filename

*4 level deep*

- filename (implies CMDLIB/filename)

The mode (random vs. sequential) of the target file dictates the function to be performed; i.e., random infers loading the resident bound program and passing control to it, while sequential implies initiation of a command file application with the designated file used as input. For the latter case, optional parameter declarations may accompany any of the file description conventions, consistent with CRUN command constructs and syntax. Assume, for example, the following input file is resident in CMDLIB with general read permission under the name, SIEV. Its purpose is to support a local command called SIEVE which produces a list of the user's first-level catalog and file names containing a specified character pattern.

```
##*NULL;(PATTERN?)  
"CATA";*  
-D;4  
"-PS:/#1/*"
```

The application may be initiated by typing SIEVE (or SIEV), whereupon the query, "PATTERN?", is issued to obtain the desired character pattern. The pattern may alternatively accompany the SIEVE command, as follows:

```
*SIEVE;;(FIL)
```

Note that this represents the equivalent of:

```
*CRUN CMDLIB/SIEV,R;;(FIL)
```

When only substitutable arguments must accompany the file name, it is permissible to substitute a space for the double quotes and left parenthesis. Thus, the SIEVE command could equivalently be entered as follows:

```
*SIEVE FIL
```

The TSS logon sequence may be augmented with user processing by creating a quick-access file named ..INIT. Following user-id and password validation, the logon subsystem passes control to the Command Loader with a special indication to process the ..INIT file. If ..INIT does not exist, the logon sequence is considered complete. Otherwise, the file is processed in the conventional manner. The message, "..INIT FILE UNUSABLE", is issued if any error is detected during such processing.

## SUPPORTING COMMANDS

Certain control functions may be exercised during a command file or deferred processing application with use of the commands COUT, CMOD and CPOS. COUT permits all output accumulated since either the beginning of the application or a previous COUT to be directed to the user's remote device or, when a file description accompanies the command, to the designated file. The description, when present, must constitute the first parameter declaration and is managed exactly the same as the output file description for CRUN. The optional parameter, EXCLUDE, may be specified when it is desired to eliminate lines containing user responses in the output. The following example utilizes COUT to place a specified SYSOUT report (generated by a CARDIN job with JOUT disposition) in the current file:

```
##*NULL;(SNUMB?,ACTIVITY?,REPORT-CODE?,DISP?)
JOUT #1
ACTIVITY #2
PRINT #3
#4
COUT *
-D;5 F;*
-B;4 D;4
```

Assuming this input file is resident in the user's own catalog under the name "EXTRACT", an example application might be initiated as follows:

```
*/EXTRACT 6837T,1,74,RELEASE
```

The CMOD and CPOS commands, while not entirely restricted for use in command file applications, permit count-controlled repetition or conditional execution of a command sequence. CMOD allows modification of the user's Program Switch Word (PSW) via arithmetic or Boolean operations and is used as follows:

```
CMOD operation-1;operation-2;...;operation-n
```

Each requested operation affects only the lower half (bits 18-35) of the PSW and may include any of the following forms:

<u>Form</u>	<u>Meaning</u>	<u>Restrictions</u>
n	Set C(PSW) bits 18-35 to n	$n < 262144$
+n	Add n to C(PSW) bits 18-35	$n < 262144$
-n	Subtract n from C(PSW) bits 18-35	$n < 262144$
Si	Set bit i	$18 \leq i < 36$
Si-j	Set bits i through j	$18 \leq i \leq j < 36$
Ri	Reset bit i	$18 \leq i < 36$
Ri-j	Reset bits i through j	$18 \leq i \leq j < 36$
Ni	Negate bit i	$18 \leq i < 36$
Ni-j	Negate bits i through j	$18 \leq i \leq j < 36$
D	<u>Display</u> C(PSW) bits 0-35	
E	Place abort code in bits 18-35	

The CPOS command permits either conditional alteration of the normal serial processing of input file commands or conditional execution of a single command, based on the contents of the PSW. Its general form is as follows:

CPOS expression;operation

The first parameter represents a Boolean expression which, when true, causes the operation implied by the second parameter to be performed. Expression operands may include any of the following:

<u>Form</u>	<u>Is True When:</u>	<u>Restrictions</u>
i	Bit i of the PSW is on	$0 \leq i < 36$
LTn	Bits 18-35 of C(PSW) are less than n	$n < 262144$
LEn	Bits 18-35 of C(PSW) are less than or equal to n	$n < 262144$
GTn	Bits 18-35 of C(PSW) are greater than n	$n < 262144$
GEN	Bits 18-35 of C(PSW) are greater than or equal to n	$n < 262144$
EQn	Bits 18-35 of C(PSW) are equal to n	$n < 262144$
NEn	Bits 18-35 of C(PSW) are not equal to n	$n < 262144$

Permissible operators include + (OR), \* (AND), - (EXCLUSIVE OR) and / (NOT). Although / is a unary operator involving only one operand, by convention A/B is taken to mean A\*/B. The expression is evaluated from left to right (without regard to operator hierarchy) and the resulting truth value (TRUE or FALSE) displayed. When false, or if a requested operation does not accompany the expression, CPOS processing is terminated. Otherwise, the operation is performed, constituting either a branch to a given response line on the input file or execution of a single specified command. A branch declaration is identified by an integer offset, optionally prefixed by + or - to denote forward or backward positioning (absence of the sign prefix implies forward positioning) or a label prefixed by a dollar sign. The offset represents the number of the response lines to forward or backspace, relative to the line containing the offset itself. Thus, the interpretation of the command,

```
CPOS 18+GE5*LT10;+2
```

is, "when bit 18 of the PSW is on or the magnitude of its lower half is greater than or equal to 5 and less than 10, then skip the command immediately following the CPOS". The expression itself is optional and, when omitted, causes the requested operation to be unconditionally performed; e.g.,

```
CPOS ;+2
```

When an offset declaration requesting a forward space exceeds the number of response lines remaining, the application is terminated normally. Thus, the command,

```
CPOS EQ0;9999
```

results in terminating the command file application in which it appears when the lower half of the PSW is zero.

The requested operation may instead be a command to be conditionally executed, optionally accompanied by any necessary parameter declarations. For example,

```
CPOS 34-35;CRUN INFIL;OTFIL
```

would initiate the requested CRUN when either bit 34 or 35 (but not both) of the PSW is set.

Both CMOD and CPOS request continuation input when the last character of the line is a semicolon. A null response to such a request indicates completion. When parameters do not accompany either command (on the same line), the query, "FUNCTION?", is issued. The user must, at this time, enter all information necessary to perform the required operation.

APPENDIX A

EXECUTIVE ERROR MESSAGES

Error  
Code

Text

1	001 - INCORRECT PRIMITIVE
2	002 - (dddddd) INVALID FILE I/O COMMAND
3	003 - (dddddd) INVALID DCW
4	004 - (dddddd) INVALID DRL ARGUMENT
5	005 - (dddddd)INVALID DRL CODE
6	006 - LEVEL OF CONTROL TOO DEEP
7	007 - BAD PROG. DESC.
8	008 - LOOP IN PRIMITIVES
9	009 - SYSTEM UNKNOWN
10	010 - PROGRAM TOO LARGE TO SWAP
11	011 - (dddddd) INCORRECT CORE FILE USAGE
12	012 - (dddddd) PRIVILEGED I/O ATTEMPTED
13	013 - (dddddd) DRL USERID NOT PERMITTED
14	(dddddd)ILLEGAL DRL RELMEM REQUEST
15	015 - (dddddd) CANNOT RESET USER ID
16	016 - (aaaaaa) OVERFLOW FAULT
17	017 - (aaaaaa) ILLEGAL OP CODE
18	018 - (aaaaaa) MEMORY FAULT
19	019 - (aaaaaa) FAULT TAG (aaaaaa) FAULT
20	020 - (aaaaaa) DIVIDE CHECK FAULT
21	021 - (ssssss) BAD STATUS - SWAP OUT
22	022 - (ssssss) BAD STATUS - SWAP IN
23	023 - (ssssss) BAD STATUS - LOAD
24	(dddddd) TALK PERMISSION NOT GRANTED
25	(dddddd) WRITE ATTEMPTED ON READ-ONLY FILE - ffffffff
26	(dddddd)READ ATTEMPTED ON EXECUTE-ONLY FILE - ffffffff
27	024 - (aaaaaa)MME FAULT
28	028 - (dddddd)REWIND ATTEMPTED FOR RANDOM FILE - ffffffff
29	029 - ILLEGAL SYSTEM SELECTION
30	134 - (dddddd)INVALID DRL FILACT FUNCTION #
31	135 - (dddddd)PRIVILEGED DRL FILACT REQUEST
32	138 - (dddddd)NO TAP* FILE FOR DRL TAPEIN
33	139 - ERROR IN WRITING TAP* FILE
34	(dddddd)DRL ABORT - CANNOT WRITE ABRT FILE
35	(dddddd)DRL ABORT - ABRT FILE WRITTEN
36	NOT ENOUGH CORE TO RUN JOB
37	SORRY - OUT OF SWAP SPACE. TRY AGAIN.
38	(dddddd)FILE ADDRESS ERROR
39	(dddddd)DRL ABORT - ABRT FILE I/O ERROR
40	(dddddd)DRL ABORT - ABRT FILE TOO SMALL
41	(ssssss)BAD STATUS DRL SAVE/RESTOR - ffffffff
42	(dddddd)H* FILE NOT IN AFT - ffffffff
43	064 - EXECUTE TIME LIMIT EXCEEDED
44	025 - (aaaaaa)LOCKUP FAULT
45	065 - OBJECT PROGRAM SIZE LIMIT EXCEEDED
46	(dddddd)INCORRECT ENTRY TO DRL TASK
47	(dddddd)H* PROGRAM NAME UNDEFINED - ffffffff

48 (dddddd)H\* FILE CATALOG FULL - ffffffff  
49 (dddddd)TALLY OR CHARACTER COUNT INCORRECT  
50 (dddddd)BAD DRL SAVE DATA LOC  
51 (dddddd)H\* FILE NOT INITIALIZED - ffffffff  
52 (dddddd)H\* FILE MUST BE RANDOM - ffffffff  
53 026 - (aaaaaa)OP-NOT-COMPLETE FAULT  
54 (dddddd)H\* FILE PROGRAM NAME REQUIRED - ffffffff  
55 027 - (aaaaaa)COMMAND FAULT  
56 (dddddd)LINKED FILE I/O CANNOT SPAN > 63 LLINKS - ffffffff  
57 UNASSIGNED  
58 (dddddd)INVALID TIME FOR DRL GWAKE  
59 UNASSIGNED  
60 (dddddd)INVALID SNUMB FOR DRL JOUT  
61 (dddddd)PRIVILEGED DRL  
62 (dddddd)INVALID DRL JOUT FUNCTION #  
63 MEMORY PARITY ERROR  
64 SY\*\* I/O ERROR

Legend:

fffffff - Name of the file associated with the error.  
dddddd - Location of the derail which caused the error.  
aaaaaa - Address in the subsystem at which the error occurred.  
ssssss - Bad file I/O status received.

INDEX

#CMD  
 #CMD 2-8

#RECOVER  
 #RECOVER 2-22

#TAPE  
 #TAPE 2-25

\*SRC  
 \*SRC 2-2

ABACUS  
 ABACUS 1-1  
 ABACUS (ABC) subsystem 5-1  
 ABACUS subsystem 2-6

ABC  
 ABACUS (ABC) subsystem 5-1

ABNORMAL  
 abnormal termination 4-9

ACCESS  
 ACCESS 1-2  
 ACCESS FILE 5-16, 5-23  
 ACCESS Functions 5-16  
 ACCESS subsystem 2-6, 5-11  
 Access type 5-18  
 SHORT FORM USAGE OF ACCESS  
 FUNCTIONS 5-17  
 Use of ACCESS 5-12

AFT  
 AFT 2-2

ALGOL  
 TSS ALGOL 1-1

APPEND  
 APPEND 5-8

APRINT  
 APRINT 2-6

ASCASC  
 ASCASC 2-6  
 ASCASC Subsystem/Command 5-6

ASCBCD  
 ASCBCD 2-6

ASCII-TO-ASCII  
 ASCII-TO-ASCII CONVERSION SUBSYSTEM  
 5-6

ASIS  
 ASIS 2-21, 5-41

ATTN  
 ATTN or INT key 4-9

AUTOMATIC  
 AUTOMATIC 2-6  
 automatic creation of line numbers  
 2-6  
 Automatic Mode 2-2  
 Automatic Paper Tape Input 4-10  
 Automatic Terminal Disconnections  
 4-8

AVAILABLE  
 Available File Table 2-2

BASIC  
 BASIC 1-1

BCDASC  
 BCDASC 2-7

BINARY CARD  
 Binary Card Format 5-43

BPRINT  
 BPRINT 2-7

BPUNCH  
 BPUNCH 2-7

BREAK  
 BREAK key 4-9

BUILD  
 Build Mode Input 4-7  
 Building File from Non-ASCII Paper  
 Tape 4-10

BYE  
 BYE 2-8

CALCULATION  
 Mode and Precision of Calculation  
 5-5



CARDIN  
CARDIN 1-1

CASE  
Case Commands 4-18

CATALOG  
CATALOG 2-8  
Catalogs and Files 5-7  
CREATE CATALOG 5-16, 5-18  
FILE NAMES, CATALOG NAMES, AND  
PASSWORDS 2-5  
LIST CATALOG 5-16, 5-31  
MODIFY CATALOG 5-16, 5-27  
PURGE CATALOG 5-16, 5-25  
RELEASE CATALOG 5-16, 5-27

CHARACTER-DELETE  
character-delete control 4-2

CHARGE  
charge number 4-4

CLEARFILES  
REMOVE CLEARFILES 2-22

CODE  
CODE 2-10  
record media code 2-6

COLLECTOR  
collector file 2-2

COMDK  
COMDK 5-41

COMMAND  
Case Commands 4-18  
Command Loader 1-2  
COMMANDS FOR VIP TERMINALS 4-18  
Time Sharing System commands 2-5

CONSTANTS  
Constants and Functions 5-3

CONTINUATION  
Continuation Lines 5-5

CONTROL  
character-delete control 4-2  
interrupt control 4-9  
line-delete control 4-3

CONVERSION  
ASCII-TO-ASCII CONVERSION SUBSYSTEM  
5-6

CPY  
CPY 2-11

CREATE  
CREATE 5-8  
CREATE CATALOG 5-16, 5-18  
CREATE FILE 5-16, 5-21

CURRENT  
current file 2-2

DATA  
Data Display and Transmission 4-13

DEACCESS  
DEACCESS FILE 5-16, 5-25

DECK SETUPS  
Sample Deck Setups 5-44

DELETE  
DELETE 2-13

DELIMITER  
EXAMPLES OF LINE DELIMITER USE 5-33  
Identifiers and Delimiters in User  
Responses 5-14  
Line delimiters 5-14, 5-15  
Word delimiters 5-14, 5-15

DISCONNECTIONS  
Automatic Terminal Disconnections  
4-8

DISPLAY  
Data Display and Transmission 4-13

DONE  
DONE 2-14

EDITING  
Editing 4-2

EDITOR  
EDITOR 2-14  
TEXT EDITOR 1-1

ENTRY  
entry marker 4-13

ERASE  
ERASE 2-14

ERROR  
Error messages 3-1

ETX  
ETX symbol 4-13

EVALUATION  
Order of Evaluation and Use of  
Parentheses 5-5

EXCLUDE  
General, Specific, and EXCLUDE  
Permission 5-9

EXECUTE  
EXECUTE 5-8

EXECUTIVE  
Time Sharing Executive 1-1

FDUMP  
FDUMP 1-2

FILE  
ACCESS FILE 5-16, 5-23  
Available File Table 2-2

## FILE (cont)

Building File from Non-ASCII Paper  
Tape 4-10  
Catalogs and Files 5-7  
collector file 2-2  
CREATE FILE 5-16, 5-21  
current file 2-2  
DEACCESS FILE 5-16, 5-25  
File and Record Control Subroutines  
1-2  
FILE NAMES, CATALOG NAMES, AND  
PASSWORDS 2-5  
Line-Numbered Files 4-7  
MODIFY FILE 5-16, 5-29  
New File 2-2  
Old File 2-2  
PURGE FILE 5-16, 5-26  
Random File Specification 5-21  
RELEASE FILE 5-16, 5-27

## FILE SYSTEM

File System Structure 5-7  
Logical Structure of the File  
System 5-10

## FOR

FOR statement 5-3  
FOR Variables 5-3

## FORM FEED

Form Feed 4-18

## FORT

TSS FORT 1-1

## FUNCTION

ACCESS Functions 5-16  
Constants and Functions 5-3  
questions associated with each  
function 5-18  
SHORT FORM USAGE OF ACCESS  
FUNCTIONS 5-17

## GENERAL

general permissions 5-8  
General, Specific, and EXCLUDE  
Permission 5-9

## GET

GET 2-15

## HELP

HELP 1-2  
HELP message explanations 3-1  
HELP subsystem 3-1

## HOLD

HOLD 2-15

## IDENTIFIERS

Identifiers and Delimiters in User  
Responses 5-14

## INPUT

Automatic Paper Tape Input 4-10  
Build Mode Input 4-7  
Keyboard input 4-2  
Paper Tape Input 4-9

## INSERT

INSERT 5-41

## INT

ATTN or INT key 4-9

## INTERRUPT

interrupt control 4-9

## JABT

JABT 2-15

## JDAC

JDAC 2-15

## JOUT

JOUT 1-2

## JOVIAL

TSS JOVIAL 1-1

## KEY

ATTN or INT key 4-9  
BREAK key 4-9

## KEYBOARD

Keyboard input 4-2  
KEYBOARD/DISPLAY TERMINAL OPERATION  
4-11  
keyboard/display terminals 4-12

## KEYWORDS

Keywords 5-14

## LABEL

label variable 5-3

## LIB

LIB 2-16

## LINE

Continuation Lines 5-5  
EXAMPLES OF LINE DELIMITER USE 5-33  
Line delimiters 5-14, 5-15

## LINE NUMBER

automatic creation of line numbers  
2-6  
line number 4-7  
Line Numbers 2-1  
Line-Numbered Files 4-7

## LINE-DELETE

line-delete control 4-3

## LINELENGTH

LINELENGTH 2-16

## LIST

LIST 2-16  
LIST CATALOG 5-16, 5-31  
LIST SPECIFIC 5-16, 5-32

## LISTE

LISTE 2-17

## LISTH

LISTH 2-17

LISTL  
LISTL 2-17

LOADER  
Command Loader 1-2

LOCK  
LOCK 5-8

LODS  
LODS 1-2

LODT  
LODT 1-2

LODX  
LODX 1-2

LOG-OFF  
Log-Off Procedure 4-8

LOG-ON  
Log-On 4-14  
Log-On Procedure 4-3

LOGICAL  
Logical Structure of the File  
System 5-10

MANUAL  
Manual Mode 2-1

MARKER  
entry marker 4-13

MEDIA  
record media code 2-6

MEDIA CONVERSION  
Media Conversion Program 1-2, 5-40  
TIME SHARING MEDIA CONVERSION  
PROGRAM 5-40

MESSAGE  
Error messages 3-1  
HELP message explanations 3-1

MODE  
Automatic Mode 2-2  
Build Mode Input 4-7  
Manual Mode 2-1  
mode 5-18  
Mode and Precision of Calculation  
5-5

MODIFY  
MODIFY 5-8  
MODIFY CATALOG 5-16, 5-27  
MODIFY FILE 5-16, 5-29

MOVE  
MOVE 2-21, 5-41

NAMES  
FILE NAMES, CATALOG NAMES, AND  
PASSWORDS 2-5

NEW  
NEW 2-18  
New File 2-2

NEWP  
NEWP 2-18

NEWUSER  
NEWUSER 2-18

NON-ASCII  
Building File from Non-ASCII Paper  
Tape 4-10

NORM  
NORM 2-21

NUMBER  
charge number 4-4  
Numbers 5-2

OLD  
OLD 2-18  
Old File 2-2

OLDP  
OLDP 2-19

OLDP#  
OLDP# 2-19

OPERATION  
KEYBOARD/DISPLAY TERMINAL OPERATION  
4-11  
TELEPRINTER OPERATION 4-1

OPERATOR  
summation operator 5-3

ORDER  
Order of Evaluation and Use of  
Parentheses 5-5

OUTPUT  
Terminating an Output Process 4-9

PAPER TAPE  
Automatic Paper Tape Input 4-10  
Building File from Non-ASCII Paper  
Tape 4-10  
Paper Tape Input 4-9

PARENTHESES  
Order of Evaluation and Use of  
Parentheses 5-5

PARITY/NOPARITY  
PARITY/NOPARITY 2-20

PASSWORD  
FILE NAMES, CATALOG NAMES, AND  
PASSWORDS 2-5  
password 4-4  
Passwords 5-7

PERM  
PERM 2-20

PERMISSION  
  general permissions 5-8  
  General, Specific, and EXCLUDE  
    Permission 5-9  
  Permissions 5-8  
  Specific permissions 5-8

PRECISION  
  Mode and Precision of Calculation  
    5-5

PRINT  
  PRINT 2-20

PROCEDURE  
  Log-Off Procedure 4-8  
  Log-On Procedure 4-3

PROCESS  
  Terminating an Output Process 4-9

PROGRAM  
  Media Conversion Program 1-2, 5-40  
  TIME SHARING MEDIA CONVERSION  
    PROGRAM 5-40

PURGE  
  PURGE 2-21, 5-8  
  PURGE CATALOG 5-16, 5-25  
  PURGE FILE 5-16, 5-26

QUESTIONS  
  QUESTIONS AND RESPONSES 5-18, 5-39  
  questions associated with each  
    function 5-18

RANDOM  
  Random File Specification 5-21

RBUG  
  RBUG 1-3

READ  
  READ 5-8

RECORD  
  record media code 2-6

RECORD CONTROL  
  File and Record Control Subroutines  
    1-2

RECOVER  
  RECOVER 2-21

RECOVERY  
  RECOVERY 5-8  
  RECOVERY SUBSYSTEM 5-37

RELEASE  
  RELEASE 2-22  
  RELEASE CATALOG 5-16, 5-27  
  RELEASE FILE 5-16, 5-27

REMOVE  
  REMOVE 2-22  
  REMOVE CLEARFILES 2-22

RESAVE  
  RESAVE 2-22

RESEQUENCE  
  RESEQUENCE 2-22

RESPONSES  
  Identifiers and Delimiters in User  
    Responses 5-14  
  QUESTIONS AND RESPONSES 5-18, 5-39

ROLLBACK  
  ROLLBACK 2-23, 5-38

RUN  
  RUN 2-23

RUNOFF  
  RUNOFF 1-1

SABT  
  SABT 1-3, 2-24

SAVE  
  SAVE 2-24

SCAN  
  SCAN 1-3

SEND  
  SEND 2-24

SMCL  
  SMCL 2-24

SPECIFIC  
  General, Specific, and EXCLUDE  
    Permission 5-9  
  LIST SPECIFIC 5-16, 5-32  
  Specific permissions 5-8

SPECIFICATION  
  Random File Specification 5-21

STATEMENT  
  FOR statement 5-3

STATUS  
  STATUS 2-25

STRIP  
  STRIP 2-21, 2-25

STRUCTURE  
  File System Structure 5-7  
  Logical Structure of the File  
    System 5-10

SUBROUTINES  
  File and Record Control Subroutines  
    1-2

SUBSYSTEM  
  ABACUS (ABC) subsystem 5-1  
  ABACUS subsystem 2-6  
  ACCESS subsystem 2-6, 5-11  
  ASCII-TO-ASCII CONVERSION SUBSYSTEM  
    5-6

SUBSYSTEM (cont)  
  HELP subsystem 3-1  
  RECOVERY SUBSYSTEM 5-37

SUBSYSTEM/COMMAND  
  ASCASC Subsystem/Command 5-6

SUMMATION  
  summation operator 5-3

SY\*\*  
  SY\*\* 2-2

SYMBOL  
  ETX symbol 4-13

SYSTEM  
  SYSTEM 2-25  
  SYSTEM DESCRIPTION 1-1  
  Time Sharing System commands 2-5

TAPE  
  TAPE 2-25

TELEPRINTER  
  TELEPRINTER OPERATION 4-1

TERMINAL  
  Automatic Terminal Disconnections  
    4-8  
  COMMANDS FOR VIP TERMINALS 4-18  
  KEYBOARD/DISPLAY TERMINAL OPERATION  
    4-11  
  keyboard/display terminals 4-12  
  VIP terminals 4-12

TERMINATING  
  Terminating an Output Process 4-9

TERMINATION  
  abnormal termination 4-9

TEXT  
  TEXT EDITOR 1-1

TIME  
  Time Sharing System (TSS) 1-1

TIME SHARING  
  Time Sharing Executive 1-1  
  TIME SHARING MEDIA CONVERSION  
    PROGRAM 5-40  
  Time Sharing System commands 2-5

TRANSMISSION  
  Data Display and Transmission 4-13

TSS  
  Time Sharing System (TSS) 1-1  
  TSS ALGOL 1-1  
  TSS FORT 1-1  
  TSS JOVIAL 1-1  
  TSS YFORT 1-1

USER  
  Identifiers and Delimiters in User  
    Responses 5-14

USER-ID  
  user-ID 4-4

VARIABLE  
  FOR Variables 5-3  
  label variable 5-3  
  Variables 5-2

VIP  
  COMMANDS FOR VIP TERMINALS 4-18  
  VIP (Visual Information Projection)  
    4-11  
  VIP terminals 4-12

VISUAL INFORMATION PROJECTION  
  VIP (Visual Information Projection)  
    4-11

WORD  
  Word delimiters 5-14, 5-15

WRITE  
  WRITE 5-8

YFORT  
  TSS YFORT 1-1