



**NOS VERSION 1
TIME-SHARING USER'S
REFERENCE MANUAL**

**CDC® COMPUTER SYSTEMS:
CYBER 170 SERIES
CYBER 70
MODELS 71, 72, 73, 74
6000 SERIES**

COMMAND INDEX

<u>Command</u>	<u>Description</u>	<u>Page</u>
ACCESS	Selects ACCESS subsystem	4-8
APL	Selects APL interactive interpreter	4-12
APPEND	Adds file(s) to end of specified indirect access permanent file	5-6
ASCII	Selects full ASCII or correspondence character set (128 characters)	4-3
ATTACH	Provides access to direct access permanent files	5-6
AUTO	Automatically generates line numbers	4-3
BASIC	Selects BASIC subsystem	4-9
BATCH	Selects batch subsystem	8-1
BEGIN	Initiates processing of CCL procedure	4-12
BINARY	Selects binary input mode	4-13
BRIEF	Suppresses all full and partial headers	4-5
BYE	Logs user out of the system and disconnects the terminal	4-14
CALL	Initiates processing of a KCL procedure	4-14
CATLIST	Lists permanent file catalog information	5-13
CHANGE	Changes attributes of a permanent file	5-9
CHARGE	Enters charge and project number for accounting	4-16
CLEAR	Releases local files	4-16
CONVERT	Converts character sets (63/64)	A-1
	Lists abbreviated system status	4-16
CSET	Selects terminal character set mode (ASCII or normal)	4-5
DAYFILE	Lists system's record of user's time-sharing activity	4-16
DEBUG	Activates, terminates, or resumes CYBER Interactive Debug mode	4-19
DEFINE	Creates direct access permanent files	5-9
DIAL	Sends messages to interactive terminal	4-19
EDIT	Selects Text Editor	4-19
ENQUIRE	Lists system and job status information	4-21
EXECUTE	Selects execute subsystem	4-9
FORTRAN	Selects FORTRAN subsystem	4-10
FTNTS	Selects FTNTS subsystem	4-11
FULL	Selects full duplex mode	4-5
GET	Obtains copy of indirect access permanent file(s)	5-10
GOODBYE	Same as BYE command	
HALF	Clears full-duplex mode	4-5
HELLO	Logs the current user out of the system and reinitiates login sequence	4-24
HELP	Provides system command information	4-24
I key	Interrupts output to terminal	4-24
LENGTH	Lists name, length, type, and last status of specified file	4-25
LIB	Retrieves copy of indirect access permanent file from user number LIBRARY	5-11; 5-26
LIMITS	Provides list of system resources available to user and extent of use	4-39
LIST	Lists a file (with header)	4-26
LNH	Lists a file (without header)	4-27
LOGIN	Same as HELLO command	
LOGOUT	Same as BYE command	

(continued on inside back cover)



**NOS VERSION 1
TIME-SHARING USER'S
REFERENCE MANUAL**

**CDC® COMPUTER SYSTEMS:
CYBER 170 SERIES
CYBER 70
MODELS 71, 72, 73, 74
6000 SERIES**

REVISION RECORD

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

**Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112**

or use Comment Sheet in the back of this manual.

© 1975, 1976, 1977, 1978, 1979, 1980

by Control Data Corporation

All Rights Reserved

Printed in the United States of America

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-	4-23	J	6-13	H	Index-2	J		
Command Index	J	4-24	J	7-1	J	Index-3	J		
Title Page	-	4-25	J	7-2	H	Index-4	J		
ii	J	4-26	J	7-3	H	Index-5	J		
iii/iv	J	4-27	J	7-4	H	Index-6	J		
v	J	4-28	J	7-5	H	Comment Sheet	J		
vi	J	4-29	J	7-6	H	Command Index	J		
vi-a/vi-b	J	4-30	J	7-7	G	Back Cover	-		
vii	J	4-31	J	7-8	G				
viii	J	4-32	J	7-9	G				
1-1	H	4-33	J	8-1	H				
1-2	J	4-34	J	8-2	H				
1-3	J	4-35	J	8-3	H				
1-4	J	4-36	J	9-1	G				
1-5	H	4-37	J	9-2	H				
2-1	G	4-38	J	9-3	G				
2-2	G	4-39	J	9-4	G				
2-3	G	4-40	J	9-5	H				
2-4	H	4-41	J	9-6	G				
2-5	G	4-42	J	9-7	H				
2-6	G	4-43	J	10-1	G				
2-7	H	5-1	H	A-1	G				
2-8	J	5-2	H	A-2	G				
2-9	J	5-3	J	A-3	G				
2-10	J	5-4	J	A-4	G				
2-11	J	5-5	H	A-5	G				
2-12	H	5-6	H	A-6	G				
3-1	J	5-7	H	B-1	G				
3-2	J	5-8	J	B-2	H				
3-3	J	5-9	J	B-3	H				
3-4	J	5-10	H	B-4	H				
3-5	H	5-11	J	B-5	J				
3-6	J	5-12	J	B-6	J				
3-7	J	5-13	H	B-7	J				
3-8	J	5-14	H	B-8	J				
3-9	J	5-15	J	B-9	J				
3-10	H	5-16	J	B-10	J				
4-1	J	5-17	H	B-11	J				
4-2	J	5-18	J	B-12	J				
4-3	J	5-19	H	B-13	J				
4-4	H	5-20	H	B-14	J				
4-5	J	5-21	H	B-15	J				
4-6	J	5-22	H	B-16	J				
4-7	J	5-23	H	B-17	J				
4-8	J	5-24	H	B-18	J				
4-9	J	5-25	H	B-19	J				
4-10	J	5-26	J	C-1	H				
4-11	J	6-1	G	C-2	H				
4-12	J	6-2	G	C-3	J				
4-13	H	6-3	H	C-4	H				
4-14	H	6-4	J	C-5	J				
4-15	J	6-5	J	C-6	H				
4-16	J	6-6	J	C-7	H				
4-17	J	6-7	H	D-1	H				
4-18	J	6-8	H	E-1	H				
4-19	J	6-9	H	E-2	G				
4-20	J	6-10	H	F-1	G				
4-21	J	6-11	H	F-2	G				
4-22	J	6-12	H	Index-1	J				

PREFACE

The Network Operating System (NOS) was developed by Control Data Corporation to provide a multimode job-processing capability for CDC® CYBER 70 Series, Models 71, 72, 73, and 74; CDC CYBER 170 Series; and CDC 6000 Series Computer Systems. Multimode job processing allows concurrent processing of local and remote batch and remote interactive (time-sharing) jobs.

AUDIENCE

This manual is written for the user who has some experience with a time-sharing system. If the user has no such experience, he should read the NOS Time-Sharing User's Guide before reading this manual. Since examples are written in either FORTRAN or BASIC, a knowledge of those languages is also helpful. It is recommended that a new user read the glossary in appendix C to become familiar with the terms used in this and other manuals.

ORGANIZATION

This manual describes the communication between the user of a time-sharing terminal and the system. It contains both a description of the subsystems available to the user and the procedures necessary to use these subsystems, but it does not contain programming information for these subsystems. This manual describes the command set available to all time-sharing users. Included are terminal control, subsystem selection, file creation, and file maintenance commands. The more experienced user may be interested in the ability to enter control statements, either one at a time or as a batch job image, from the terminal. These features are described in sections 7 and 8.

CONVENTIONS

Throughout this manual, the following conventions are used.

- The word system refers to NOS.
- A default value refers to the value the system supplies for a parameter when the user does not specify one. Site personnel may change this value during installation.
- \textcircled{R} denotes the message terminator key on the keyboard. This may be the RETURN, CR, CARRIAGE RETURN, or NEW LINE key, depending on the type of terminal. The user of a time-sharing terminal must end each line with a \textcircled{R} . Since this is always true, \textcircled{R} is noted only when the user may be unsure if the message should be terminated. The system responds to \textcircled{R} by performing a carriage return and line feed operation (positioning the carriage at the first character position on the next line).
- Examples of actual terminal sessions that appear in this manual were produced on an ASCII terminal unless otherwise specified. Uppercase characters represent terminal output; lowercase characters represent user input. (However, user input that is displayed within the text of this manual is shown in uppercase characters.) All terminal activity is either displayed on a screen or printed on continuously fed paper. Examples in this manual assume printing. The vertical spacing in examples does not necessarily coincide with the spacing that appears on the user's terminal.

- References to FORTRAN include both FORTRAN Extended Version 4 (under FTNTS subsystem) and FORTRAN Version 5 (under FORTRAN subsystem).
- ESCAPE denotes the escape key on the terminal keyboard. Depending on the terminal, this key may actually be ESCAPE or ESC.
- BREAK denotes the interrupt key on the terminal keyboard. Depending on the terminal, this key may actually be INT, INTRPT, or BREAK.
- CNTL denotes the control key on the terminal keyboard. Depending on the terminal, this key may actually be CNTL, CTRL, or CNTRL. The display code transmitted by control characters may vary among terminals. The user should refer to the terminal operator's guide for information about the functions of the control characters at his terminal. CNTL followed by a character directs the user to hold down the control key and type the indicated character.
- Extended memory for the CYBER 170 Models 171, 172, 173, 174, 175, 720, 730, 750, and 760 is extended core storage (ECS). Extended memory for CYBER 170 Model 176 is large central memory (LCM) or large central memory extended (LCME). In this manual ECS refers to all forms of extended memory on the CYBER 170 Series.

RELATED PUBLICATIONS

The following manuals contain additional information for the user. The NOS Manual Abstracts is a pocket-sized manual containing brief descriptions of the contents and intended audience of all NOS and NOS product set manuals. The abstracts manual can be useful in determining which manuals are of greatest interest to a particular user. The Software Publications Release History serves as a guide in determining which revision level of software documentation corresponds to the Programming System Report (PSR) level of installed site software.

<u>Control Data Publication</u>	<u>Publication Number</u>
ALGOL Version 4 Reference Manual	60496600
ALGOL Version 5 Reference Manual	60481600
APL Version 2 Reference Manual	60454000
BASIC Version 3 Reference Manual	19983900
COBOL Version 4 Reference Manual	60496800
COBOL Version 5 Reference Manual	60497100
COMPASS Version 3 Reference Manual	60492600
CYBER Interactive Debug Version 1 Reference Manual	60481400
CYBER Loader Version 1 Reference Manual	60429800
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual	60495700
FORTRAN Extended Version 4 Reference Manual	60497800

<u>Control Data Publication</u>	<u>Publication Number</u>
FORTRAN Version 5 Reference Manual	60481300
Modify Instant	60450200
Modify Reference Manual	60450100
NOS Version 1 Applications Programmer's Instant	60436000
NOS Version 1 Batch User's Guide	60436300
NOS Version 1 Export/Import Reference Manual	60436200
NOS Version 1 Manual Abstracts	84000420
NOS Version 1 Reference Manual, Volume 1	60435400
NOS Version 1 Reference Manual, Volume 2	60445300
NOS Version 1 Systems Programmer's Instant	60449200
NOS Version 1 Terminal User's Instant	60435800
NOS Version 1 Time-Sharing User's Guide	60436400
PL/I Version 1 Reference Manual	60388100
Software Publications Release History	60481000
Sort/Merge Version 4 and 1 Reference Manual	60497500
TAF/TS Version 1 Data Manager Reference Manual	60453100
TAF/TS Version 1 Reference Manual	60453000
TAF/TS Version 1 User's Guide	60436500
Text Editor Reference Manual	60436100
Update Version 1 Instant Manual	60456000
Update Version 1 Reference Manual	60449900
XEDIT Version 3 Reference Manual	60455730

Publication ordering information and latest revision levels are available from the Literature Distribution Services catalog, publication number 90310500.

DISCLAIMER

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

CONTENTS

1. GENERAL DESCRIPTION	1-1	Time-Sharing Job Commands	4-12
Introduction	1-1	Limits Command	4-39
Terminals	1-1		
Subsystems	1-2		
Input/Output Conventions	1-3	5. PERMANENT FILE COMMANDS	5-1
Length of Input/Output Lines	1-3	Introduction	5-1
System Messages During Input	1-3	Permanent File Command Format	5-1
Termination of Input Line	1-3	Permanent File Processing Commands	5-6
Correction of Input Line	1-3	CATLIST Command	5-13
Deletion of Input Line	1-4	Command Format	5-14
Input to Executing Program	1-4	Information Format	5-17
Interruption of Executing Program	1-4	Examples	5-20
Termination of Executing Program	1-4	Alternate User Access	5-23
Control Bytes	1-5	Private File Category	5-23
		Semiprivate File Category	5-24
		Public File Category	5-25
		LIB Command	5-26
2. LOGIN AND LOGOUT PROCEDURES	2-1		
Introduction	2-1	6. ADDITIONAL SYSTEM FEATURES	6-1
Login Procedure	2-1	Program Editing	6-1
Preparation	2-1	Resequencing Line Numbers	6-2
Login Initiation	2-2	Job Suspension	6-4
Checking Switches	2-3	Recovery	6-8
Dialing In	2-3	Error Control	6-13
Identifying Terminal	2-4		
Login Sequence	2-7		
Logout Procedure	2-11		
3. FILE USAGE	3-1	7. REMOTE JOB SUBMITTAL	7-1
Introduction	3-1	General Description	7-1
File Structure	3-1	Entering SUBMIT Command	7-1
Reserved File Names	3-2	Reformatting the Submit File	7-2
Permanent Files	3-2	Error Processing	7-9
Temporary and Local Files	3-4		
File Sorting	3-6	8. BATCH SUBSYSTEM	8-1
Local File Control	3-9		
4. SYSTEM COMMANDS	4-1	9. PAPER TAPE OPERATIONS	9-1
Introduction	4-1	General Description	9-1
Command Processing	4-1	Teletypewriter	9-1
Terminal Control Commands	4-3	Control Characters	9-2
Subsystem Selection Commands	4-8	Input Lines	9-2

Punching a Tape Off-Line	9-3	10. BLOCK EDIT MODE TERMINAL USE	10-1
Tape Mode	9-3		
Punching a Tape On-Line	9-3		
Corrections	9-4		

APPENDIXES

A. CHARACTER SETS	A-1	D. MASS STORAGE DEVICE STATISTICS	D-1
B. TERMINAL MESSAGES	B-1	E. TEXT EDITOR COMMAND INDEX	E-1
C. GLOSSARY	C-1	F. XEDIT COMMAND INDEX	F-1

INDEX

FIGURES

2-1 Login Procedure	2-2	2-2 Sample Keyboards	2-5
---------------------	-----	----------------------	-----

TABLE

2-1 Functions of Time-Sharing Terminal Controls	2-6
---	-----

GENERAL DESCRIPTION

1

INTRODUCTION

NOS provides an interactive terminal user with time-sharing capabilities. It can be used from any location by dialing the computer's telephone number.

NOS is a powerful software system that satisfies a wide spectrum of computational needs. Its conversational capabilities permit users to debug their programs much faster than other data processing systems. Its mass storage and advanced file maintenance techniques permit large amounts of information to be stored at high speeds. The system can be used from terminals in remote locations, thereby eliminating travel to the computer site by allowing users to work in their own offices.

NOS is capable of several concurrent processing modes. The following are the processing modes available.

- Time-sharing.
- Local batch.
- Remote batch.
- Transaction.

NOS performs time-sharing and transaction processing through the time-sharing executive and remote batch processing through Export/Import. NOS can also perform time-sharing, remote batch, and transaction processing through the Network Access Method (NAM) communications software. These processing modes are implemented, respectively, by the following NAM applications: Interactive Facility (IAF), Remote Batch Facility (RBF), and Transaction Facility (TAF).

The primary emphasis of this manual is on time-sharing processing that involves the time-sharing executive. Local batch processing is explained in the NOS Reference Manual, volume 1.

TERMINALS

All communication with the system can take place through a remote terminal. Programs or data are sent to the system according to specific rules or commands. The system responds by sending its answers to the terminal. If the user enters an incorrect command, the system rejects the command and prints the reason for its rejection.

Two types of terminals can be used to communicate with the system.

- ASCII code compatible terminals (such as Teletype models 33, 35, 37, and 38; Memorex 1240; and so on).
- Correspondence code terminals (such as Novar 5-41, Datel 30, IBM 2741, and so on).

If a correspondence code terminal is used, or if transmission speeds greater than 10 characters per second are required, the system may require that the communications-system equipment be modified.

Users can perform most time-sharing operations with either the ASCII or correspondence code terminal. Any operating difference that exists between the two terminals is described when it affects procedures covered in this manual.

SUBSYSTEMS

After identifying himself to the system, the user can specify a subsystem to be used. The available subsystems are as follows:

<u>Subsystem</u>	<u>Description</u>
Access	Allows communication between two users at interactive terminals. A validated user can log in at an interactive terminal (not the central site console) and communicate with any other user at another interactive terminal currently logged into the system.
BASIC	Provides an environment specifically designed for the maintenance and execution of BASIC programs.
Batch	Provides the user with a batch control statement capability from the terminal. This subsystem enables a user to type control statements at a terminal that normally would have to be entered from a card reader at the central site, entered from a remote batch terminal, included in a procedure file, or included in a job submitted to the central site.
Execute	Allows a user to execute a previously compiled program. Since compilation accounts for a majority of the time spent on a job, not having to recompile a program each time provides an efficient means of running a frequently used program.
FORTRAN or FTNTS	Provides an environment specifically designed for maintaining and executing FORTRAN programs. The FORTRAN subsystem uses the FORTRAN version 5 compiler, and the FTNTS subsystem uses the FORTRAN Extended version 4 compiler.
Null	Allows the user to perform file manipulations and other time-sharing operations without subsystem association.
Transaction	Allows the validated time-sharing user to access the Time-Sharing Transaction Facility (TAF/TS).

INPUT/OUTPUT CONVENTIONS

The following conventions and standards apply to a time-sharing terminal.

LENGTH OF INPUT/OUTPUT LINES

The input line can consist of a maximum of 150 characters. The output line can be any length; however, it should not exceed the page width of the terminal being used. If the user attempts to input or output more characters per line than are allowed, the additional characters overprint at the end of the line, unless the terminal being used has an automatic carriage return feature.

SYSTEM MESSAGES DURING INPUT

The following diagnostic messages can appear during input from the terminal.

- | | |
|-----------------------------|---|
| *OVL* | Line overflow. This message is issued when more than 150 characters have been entered since the last carriage return (the entire line is lost when the 151st character is entered). |
| *RE-ENTER LAST LINE* | Data was lost during the last line of input. The user should reenter the last line. |

TERMINATION OF INPUT LINE

The user must terminate each line of input information by pressing **CR**. This tells the system that the current input line is complete. The system responds by positioning the carriage at the beginning of the next line. The user can then enter additional input information on the new line.

CORRECTION OF INPUT LINE

The user can correct entry errors in the input line, before pressing **CR**, by using the backspace character. The backspace character on ASCII code terminals is either the BACK SPACE key or, on terminals that do not have a BACK SPACE key, the CNTL H keys. On correspondence code terminals, the BACK SPACE key is used. One character (including spaces) is deleted for each backspace character entered and the backspace character is not printed. If the beginning of the line is reached, further backspace characters are ignored. For example, when the input line contains

BAX←SJK←←IC

it is interpreted by the system as the BASIC subsystem selection command. **←** indicates a backspace.

DELETION OF INPUT LINE

When the user discovers an error before pressing **CR**, he can delete the current line in one of the following ways. On ASCII code terminals, press the ESCAPE key or press the CNTL [keys. On correspondence code terminals, press the ATTN key. The system ignores the entire input line, responds by printing *DEL*, and positions the carriage at the first character position on the next line. The following example uses the ESCAPE key.

```
new,test  
  
READY.  
10 program t(output)  
20 print 5  
30 6 format 9*this is a*DEL* ←————— ESC key is pressed.  
30 5 format (*this is it*)  
40 end  
1nh  
  
10 PROGRAM T(OUTPUT)  
20 PRINT 5  
30 5 FORMAT (*THIS IS IT*)  
40 END  
READY.
```

INPUT TO EXECUTING PROGRAM

A ? output to the terminal normally indicates that the executing program has requested input. However, the program may include question marks in its normal output. Programs coded in COMPASS or calling a COMPASS subprogram can suppress the input prompt. (Refer to Program Control of Terminal Activity in the NOS Reference Manual, volume 2.)

INTERRUPTION OF EXECUTING PROGRAM

To interrupt an executing program that is currently transmitting output to an ASCII code terminal, press the I key or the BREAK key. Press the ATTN key on a correspondence code terminal.

This process is called job suspension and is described in section 6.

TERMINATION OF EXECUTING PROGRAM

To terminate an executing program that is currently transmitting output, press the S key on an ASCII code terminal or the ATTN key on a correspondence code terminal. To terminate an executing program that is not currently transmitting output, enter the STOP command.

CONTROL BYTES

The user can design an interactive program to control terminal activity by including control bytes in his output. Control bytes perform operations such as controlling the positioning of the printing element and defining alternate input modes.

A control byte is a 12-bit quantity, right-justified in bit position 0, 12, 24, 36, or 48 of a central memory (CM) word. The user must be careful since data can be mistaken for a control byte. For example, the characters :D at the beginning of a line and followed by an end-of-line may log out the user if written to the terminal output file, since the code 0004 is transmitted. The following control bytes are available.

<u>Byte</u>	<u>Description</u>
0001 or 0002 }	End-of-block
0003	Auto input
0004	Log out user
0005	Initiate transparent input mode
0006	Initiate binary input mode
0007	Initiate binary output
0011	Initiate ASCII output
0013	End-of-string

The use of these control bytes is described in detail in the NOS Reference Manual, volume 2.

LOGIN AND LOGOUT PROCEDURES

2

INTRODUCTION

This section discusses the login procedure and the logout procedure. The user cannot harm the system by making incorrect entries or other mistakes at the terminal. Mistakes can have only diagnostic consequences; that is, they only produce error messages (usually ILLEGAL COMMAND) and do not damage the system or hardware in any way. Generally, the user is allowed as many chances as necessary to make a correct entry. However, if he is unsuccessful at login four times in succession, the terminal is automatically disconnected. If this happens, the user should check his user number and his procedures and try again.

LOGIN PROCEDURE

The procedure has three basic steps.

<u>Step</u>	<u>Description</u>
Preparation	The user gathers information he needs before login.
Login initiation	The user connects the terminal to the system.
Login sequence	The user identifies himself as a valid user of the system and enters optional commands.

PREPARATION

Before attempting to connect to the system, the user must have the answers to the following:

- What kind of terminal is being used (correspondence code or ASCII code)?
- What kind of print does the terminal use (standard or APL)?
- How is the terminal coupled to the computer (data set, acoustic coupler, or hardwired)?
- If the terminal is not hardwired, what is the phone number?
- Does the system require a family name? If so, what is it?
- What is the user number? Everyone must have a user number to gain access to the system at login.
- Is there an associated password? If so, this password must be used along with the user number to gain access to the system.
- Are charge number and project number required? If so, it is also necessary to enter a valid charge number and project number to gain access to the system.

LOGIN INITIATION

The user is now ready to perform the steps that lead to the login sequence. The steps are:

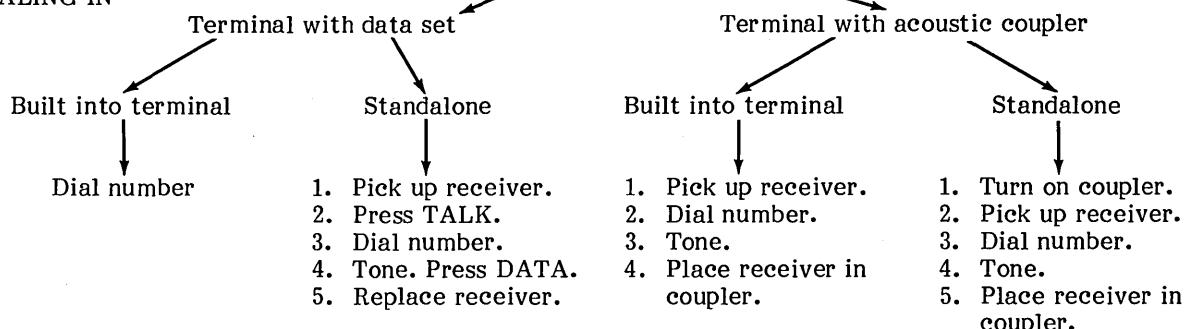
1. Checking switches.
2. Dialing in.
3. Identifying terminal.

The following paragraphs describe these steps; figure 2-1 summarizes them. Due to the variety of terminals and sites, the action taken in each step may vary considerably from site to site: the user should consult the owner's manual, provided with the terminal, and site documentation.

CHECKING SWITCHES

Turn power on.
 FULL duplex or HALF duplex: usually HALF.
 Parity: set switch on ASCII code terminal to EVEN position; set switch on correspondence code terminal to ODD position.
 Line speed: set to speed specified by installation.

DIALING IN



IDENTIFYING TERMINAL

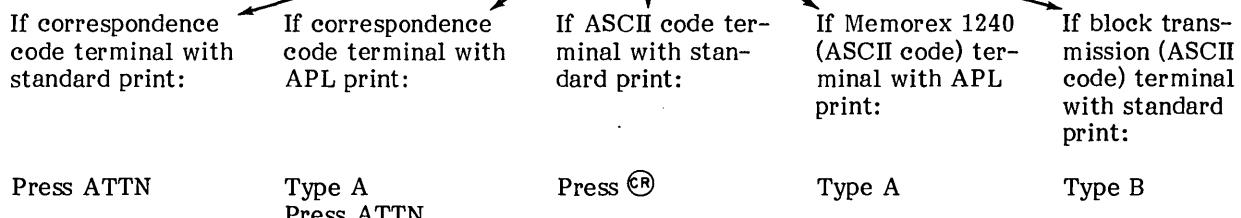


Figure 2-1. Login Procedure

Checking Switches

To check switches, the user should do the following:

- Find the power switch on the terminal and turn it to the ON position.
- Does the terminal have a duplex switch? If it does, ensure that it is turned to the HALF position. In the FULL position, characters entered from the terminal are not printed.
- Does the terminal have a parity switch? If it does, set it to the ODD position for correspondence code terminals and to the EVEN position for ASCII code terminals.
- Does the terminal have a line-speed switch? If it does, set it to the position that has been selected for the particular site.
- If the data set is separate from the terminal, turn it on and check the connection between the data set and the terminal.

Dialing In

The dial-in procedure connects the terminal to the computer through a data set or an acoustic coupler. Either one may be located in the terminal or as a separate unit beside the terminal. Hardwired terminals are already connected to the computer so the dial-in procedure is not necessary.

If the data set is located in the terminal, a telephone dial is located on the terminal panel. When the terminal is turned on, the dial tone is heard. Dial the computer number. When the connection is made, the terminal is ready to begin the login sequence.

If the data set is separate from the terminal, the user must do the following:

1. Pick up the phone receiver.
2. Press the TALK button on the face of the phone; the button lights.
3. Dial the computer number.
4. When a continuous high-pitched tone sounds, press the DATA button on the face of the phone; the button lights.
5. Replace the phone receiver. Connection is made and the terminal is ready to begin the login sequence.

If the acoustic coupler is located in the terminal, it automatically turns on when power is applied to the terminal. If the acoustic coupler is a separate unit, it must be turned on after power is applied to the terminal. In either case, a standard telephone is used in the following manner.

1. Pick up the receiver and listen for a dial tone.
2. Dial the computer number.
3. A continuous high-pitched tone indicates that the call has been answered and connection has been established.
4. Place receiver in rubber suction cups of acoustic coupler with the cord at the end indicated on the coupler.

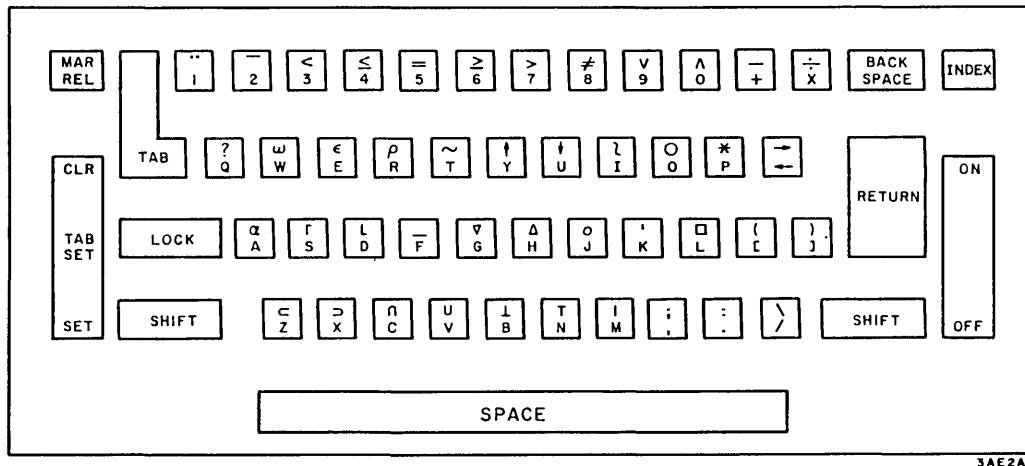
Identifying Terminal

Once the user establishes connection with the computer, it may be necessary to identify the terminal before the login sequence can begin. In some terminals, the login sequence is automatically initiated when the dial-in is completed; in others, it requires the typing of a letter, the pressing of a particular key, or both. In all cases, the system indicates that the login sequence may begin by typing out three lines: date and time, header label, and a request for family name or a request for user number.

The following terminal types require the indicated actions to initiate the login sequence.

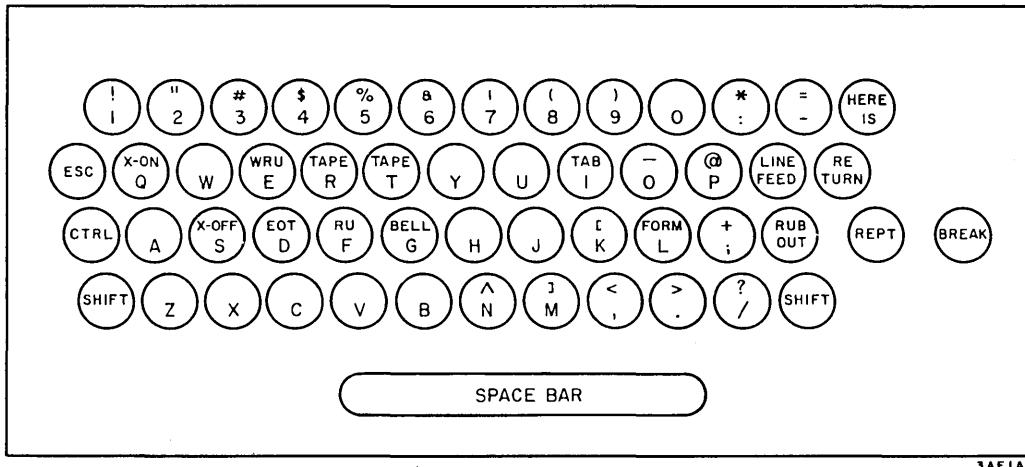
<u>Terminal Type</u>	<u>Action</u>
Correspondence code terminal with standard print	Press ATTN key.
Correspondence code terminal with APL print	Type the letter A and then press ATTN key.
ASCII code terminal with standard print	Press \textcircled{CR} .
Memorex 1240 (ASCII code) terminal with APL print	Type the letter A.
Block transmission (ASCII code) terminal with full display screen editing capability and standard print (for more information on block edit mode terminals, refer to section 10)	Type the letter B.

Figure 2-2 shows sample keyboards for two common terminals. Table 2-1 summarizes the functions of the keys as they appear on typical ASCII and correspondence code terminals.



3AE2A

Novar 5-41 Keyboard (APL Print)



3AE1A

Teletype Model 33 Keyboard (Standard Print)

Figure 2-2. Sample Keyboards

TABLE 2-1. FUNCTIONS OF TIME-SHARING TERMINAL CONTROLS

Function of Control Key or Command	Controls						
	Typical ASCII Code Terminals				Typical Correspondence Code Terminals		
	Teletype 33/35	Teletype 37/38	CDC 713 Display	Memorex 1240	IBM 2741	Novar 5-41	Datel 30
Delete current input line	ESC or CTRL [ESCAPE or CTRL [ESCAPE or CNTRL [ESC or CNTL [ATTN	ATTN	ATTN
Backspace	BACK SPACE or CTRL H	BACK SPACE or CTRL H	BACK SPACE or CNTRL H	BACK SPACE or CNTL H	BACK SPACE	BACK SPACE	BACK SPACE
Enter text mode	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command
Exit text mode	BREAK† or ETX (CTRL C)	INTRPT† or ETX (CTRL C)	BREAK† or ETX (CNTRL C)	INT†	ATTN†	ATTN†	ATTN†
Interrupt program (during output)	BREAK or I key	INTRPT or I key	BREAK or I key	INT or I key	ATTN	ATTN	ATTN
Interrupt program (no output)	BREAK	INTRPT	BREAK	INT	ATTN	ATTN	ATTN
Terminate program (during output)	S key	S key	S key	S key	ATTN	ATTN	ATTN
Terminate program (no output)	STOP command	STOP command	STOP command	STOP command	STOP command	STOP command	STOP command
Punch leader on paper tape	DELETE RUB OUT or NULL	DELETE or NULL	NA	NA	NA	NA	NA

†Effective only when entered in first character position of line (empty input line).

LOGIN SEQUENCE

The login sequence begins with the terminal typing out three lines. The first line is in the format:

yy/mm/dd. hh.mm.ss.

in which the date is given in year/month/day and the time in hours. minutes. seconds, using the 24-hour format. For example, if the login began at 12 minutes and 44 seconds after 2 o'clock in the afternoon on February 3, 1979, the first line would read:

79/02/03. 14.12.44.

The second line is the header identifying the site which may give the company name, the operating system, and the version of the operating system. In the examples in this manual, the following is used.

CDC MULTI-MODE OPERATING SYSTEM NOS 1

The third line is a request for a family name, if the site is dividing its mass storage devices into families. The terminal types

FAMILY:

The user types on the same line the name of the family for which he is assigned and presses **CR**. For example, if he is assigned to family AAA, he types

FAMILY: AAA

The family name identifies the mass storage device(s) that contains the permanent files for a specific NOS system. Each system has its own family of permanent file devices. Thus, if the system is providing backup support for another NOS system, if this login is being made into an alternate NOS system (two or more sets of permanent file devices identified with family names), or if the user is logging into a multimainframe system, the family name entered identifies the device(s) that contains the user's normal permanent file family.

The device containing the validation file for the family must be present in the alternate system in order for the user to log in. The validation file is used to determine if the user number and password entered are valid.

When his family is the default family name for the system, the user need only press **CR**.

FAMILY: CR

The system then requests the user number. When family names are not in effect, the request for a family name is omitted and the request for a user number is the third line typed out in the login sequence. The request for user number is

USER NUMBER:

The user types his user number on the same line and presses **CR**. For example, if his user number is XX234XX, he types

USER NUMBER: XX234XX

The system then responds

PASSWORD

XXXXXXXX

The second line results from the system overtyping a variety of characters. The object is to preserve password secrecy. After creating this row of blackouts, the cursor moves back to the first blackout. If a password is in effect for this user, he types it over the blackouts and presses **CR**.

If a user has exhausted the security count on the user number entered, the system issues the message

ILLEGAL USER ACCESS - CONTACT SITE OPR.

The supplied user number is denied all access to the operating system until the security count has been reset.

When the request for a user number appears, the user may enter his user number and password on the same line. To do this, he types a comma after his user number, types in his password on the same line, and then presses **CR**. For example, if he had previously given himself the password PASS23, he could now complete the request for user number as:

USER NUMBER: XX234XX,PASS23

When this is done, the system omits the two lines composed of PASSWORD and blackouts.

When the request for a family name appears, the user may enter the family name, user number, and password on the same line. If the user's family is the default family name for the system, he first types a comma and then follows with his user number and password. For example:

FAMILY: AAA,XX234XX,PASS23

or

FAMILY: ,XX234XX,PASS23

When this is done, the system omits the three lines composed of USER NUMBER, PASSWORD, and blackouts.

If the family name, user number, or password is not acceptable, the system responds:

IMPROPER LOG IN, TRY AGAIN.

FAMILY:

If the user is unsuccessful at logging in four times in succession, the system responds:

ILLEGAL TERMINAL.

and then disconnects the terminal from the system.

If the family name, user number, and password are acceptable, the system prints the terminal number and type on the next line. For example:

TERMINAL: 45,TTY

This is terminal 45 in the time-sharing configuration for this site. The TTY indicates it is an ASCII code terminal with standard print. If a terminal with a display screen is in use, the user should write down the terminal number (45 in this example) in case recovery is attempted (refer to section 6). Possible terminal types are:

<u>Type</u>	<u>Description</u>
TTY	ASCII code terminal with standard print.

<u>Type</u>	<u>Description</u>
COR	Correspondence code terminal with standard print.
CORAPL	Correspondence code terminal with APL print.
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print.
BLKEDT	Block transmission (ASCII code) terminal with full display screen editing capability and standard print.

If the maximum number of users are logged in, the system replies SYSTEM BUSY, PLEASE TRY LATER. If the operator has closed the system, the message SYSTEM CLOSED is received. The user must wait and try again later.

Normally, however, the system responds:

RECOVER/SYSTEM:

or

RECOVER/CHARGE:

Either form requests the user to respond with a single choice. RECOVER is the special case in which the user is logged in and is unexpectedly disconnected. He can log back in and try to continue from where he was disconnected by typing RECOVER and the previously associated terminal number on the same line (refer to section 6).

If the RECOVER/CHARGE response is issued, the user may type CHARGE, followed by a valid charge number and project number.

CHARGE,chargeno,projectno

For greater security, the user can enter the CHARGE command with no parameters. The system responds

ENTER CHARGE NUMBER:

? ~~XXXXXXXXXX~~

The user then types his charge number over the area the system has blacked out and presses **CR**.

The system then requests the user's project number in the same manner.

If the user's charge and project numbers are valid, the system responds

READY.

The user then enters the command for the subsystem to be used (refer to RECOVER/SYSTEM response below) or any other valid command (described in sections 4 and 5).

If the RECOVER/SYSTEM response is issued, the user enters the command for the subsystem to be used or any other valid command (described in sections 4 and 5). The following are the commands for calling subsystems.

<u>Command</u>	<u>Description</u>
ACCESS	To communicate with another user at an interactive terminal. (Only validated users can enter the access subsystem. Refer to the LIMITS Command in section 4.)
BASIC	To use the BASIC subsystem with the BASIC compiler.
BATCH,nnnnn	To enter batch control statements interactively from the terminal; nnnnn is optional and specifies the field length to be used (if omitted, 0 is assumed).
EXECUTE	To execute a previously compiled program.
FORTRAN	To use the FORTRAN subsystem with the FORTRAN version 5 compiler.
FTNTS	To use the FORTRAN subsystem with the FORTRAN Extended version 4 compiler.
NULL	To perform file manipulations and other time-sharing operations without subsystem association.

In response to the ACCESS command, the system prints:

READY.

if the user is validated for this subsystem.

In response to the BASIC, FORTRAN, FTNTS, or EXECUTE command, the system prints:

OLD, NEW, OR LIB FILE:

In response to the BATCH,nnnnn command, the system prints:

\$RFL,nnnnn.
/

nnnnn is the specified field length.

In response to the NULL command, the system prints:

READY.

NOTE

A subsystem is automatically selected by the system at login if this is specified in the system validation file. The validation file determines the internal system controls associated with each valid user number. Refer to LIMITS Command in section 4 for further information.

In response to the request:

OLD, NEW, OR LIB FILE:

the user specifies the appropriate file type. This is done by entering one of the following commands.

<u>Command</u>	<u>Description</u>
OLD	For a file previously saved as an indirect access permanent file.
NEW	For a new primary temporary file.
LIB	For an indirect access permanent file that resides in the catalog of special user number LIBRARY.

The system responds:

FILE NAME:

The user enters a one- to seven-character file name.

lfn

lfn is the file name.

If a file called by the OLD or LIB command does not exist, the system responds:

pfn NOT FOUND.

pfn is the name of the permanent file requested. The user then reenters the appropriate file command and file name.

If no errors are detected, the system responds:

READY.

LOGOUT PROCEDURE

When the user has completed his work at the terminal, he logs out. The system automatically logs out the terminal if no activity is registered in any 10-minute period.[†] To log out of the system, the user types

GOODBYE

or

BYE

or

LOGOUT

[†]If the user is validated for no-timeout terminal status, the system does not automatically log out the terminal after periods of inactivity. Refer to the LIMITS Command in section 4 for further information.

The system responds by printing:

user number LOG OFF hh.mm.ss.

user number SRU s.ss UNTS

and then disconnecting the terminal. hh.mm.ss is the time of logout and s.ss is a measure of the system resources used from login to logout.

While the terminal is connected, another user may wish to use the terminal. To log the present user out of the system and reinitialize the login sequence, the user types:

HELLO

or

LOGIN

The system logs the current user out of the system, issues the normal logout messages, and automatically initiates a new login sequence. The new user proceeds as described in this section, starting with Login Sequence.

INTRODUCTION

With the exception of some central memory tables, all data in the system is either a file or a part of a file. This section describes mass storage file organization and explains the fundamentals of file processing. Refer to the NOS Reference Manual, volume 1, for detailed information concerning files.

FILE STRUCTURE

A file is the largest collection of information the user can address by name (one through seven characters). It begins with a beginning-of-information (BOI), supplied by the system, and ends with an end-of-information (EOI). The system automatically writes an EOI as the last physical item of information on all files. A file consists of zero or more logical records or zero or more logical files, each logical file consisting of zero or more logical records.

A logical record, which can be fixed or variable in length, is a group of related words or characters. In this manual, it is also called a record (refer to appendix C). A logical record is independent of its physical environment; that is, a logical record on punched cards contains the same information as its counterpart on mass storage. A logical record consists of zero or more physical record units (PRUs) and ends with an end-of-record mark (EOR). CYBER Record Manager may define records with different structures and may subdivide a record into its own record types.

A PRU is the smallest division of data on a device, and its size depends on the device. On all mass storage devices, a PRU is 640 6-bit characters (64 central memory words). The size of a file in PRUs is limited by the user's validation (refer to LIMITS Command in section 4) and the mass storage device on which the file resides (refer to appendix D).

A logical file is zero or more logical records. It ends with an end-of-file mark (EOF). Several NOS utilities allow the user to write logical EOF marks on a file (for example, XEDIT and FORTRAN). In this way, a file actually becomes a multifile file as shown below.

(BOI)...data...(EOR)...data...(EOR)(EOF)...data...(EOR)(EOF)(EO)

The following is a typical file.

One line of an invoice forms an item, a complete invoice forms a record, a set of such records forms a file, and the collection of invoice files forms a multifile file.

If a file is positioned at BOI, and the user enters a read operation, the system reads the first logical record. If the user enters a write operation, the system starts writing data on the first PRU and terminates the logical record with an EOR and EOI. If the file is positioned at EOR or EOF, and the user enters a read operation, the system reads the next record, EOF, or EOI. If the file is positioned at EOR or EOF, and the user enters a write operation, the system adds the data, an EOR, and an EOI to a new physical record following the current EOR or EOF. A file positioned at EOI is positioned at the end of the last physical record of the file. If the user enters a read operation when a file is positioned at EOI, the system reads the EOI and transmits no data. If the user enters a write operation, the system writes over the existing EOI and adds a new logical record to the file followed by an EOR and EOI.

The following examples illustrate possible positions within a file.

File 1 ...data...(EOR)(EOF)(EOI)



File 2 ...data...(EOR)...data...(EOR)(EOF)...



File 3 ...data...(EOR)(EOI)



1 File positioned at BOI.

2 File positioned at EOR.

3 File positioned at EOF.

4 File positioned at EOI.

5 File positioned at intermediate point in data.

RESERVED FILE NAMES

Several file names have special significance to the system. In some cases, the system considers these file names to be reserved. In other cases, use of these file names causes unpredictable results. The user should take care when using these file names. The message:

RESERVED FILE NAME.

indicates that the user has attempted to use one of these file names illegally.

The reserved file names are:

INPUT	SCR1
OUTPUT	SCR2
PUNCH	SCR3
PUNCHB	SCR4
P8	ZZxxxxx
SCR	

ZZxxxxx refers to file names beginning with ZZ. Many NOS products, such as CYBER Control Language, use internal scratch files whose names begin with ZZ. Use of files beginning with ZZ may cause unpredictable results. For additional reserved file names, the user should consult the reference manual for the product he is using.

PERMANENT FILES

Permanent files are files that are stored in the system so that users can access them during a terminal session or in subsequent sessions. They are not released at logout. All time-sharing users can access permanent files after they are logged in. Validated users can also create permanent files (refer to LIMITS Command in section 4 for information concerning user validation limits). Permanent files remain in the system until they are released by a user or site personnel. A user with write permission can release a permanent file by entering the PURGE command.

Permanent files that are lost or destroyed can be recovered by site personnel if a backup copy is maintained. The user indicates whether a backup copy should be maintained when he enters the CHANGE, DEFINE, or SAVE command (refer to section 5).

The user number entered during login identifies a specific part of the catalog in the permanent file system. This catalog contains a list of file names saved under the user number as well as pertinent information about each file. All permanent file requests are made to this part of the catalog unless the user specifies an alternate user number or an auxiliary device in the file request. An auxiliary device is a mass storage permanent file device used to supplement the family devices. Family devices are the normal permanent file devices associated with the system. If a validated user specifies that the file resides on an auxiliary device, the request is made to the catalog on that device. If a user specifies an alternate user number on a permanent file request, the request is made to the part of the catalog that contains the alternate user's file names.

User numbers that contain asterisks (*) represent users with automatic read-only permission to files in catalogs of other users. The user number must match the alternate user number in all characters that are not asterisks. For example, a user with user number *AB*DE* can read and execute all permanent files of the following users.

UABCDEF
UABDDEE
MABCDE1
MAB1DE3

All permanent files are classified according to the manner in which they are accessed: indirect or direct access. The mode of access is determined by the command used to create the file. How the file is created also determines which command the user must enter to access the file. The characteristics for each type of file follow.

Indirect Access

An indirect access file is a permanent file that the user cannot access directly. The user creates an indirect access file by entering the SAVE or REPLACE command. To access the file, the user enters the OLD, LIB, or GET command.

The system makes a copy of the permanent file for use during the terminal session. The copy, instead of the permanent file, is referenced in all subsequent operations. This feature allows the user to make changes to a copy of a file without changing the permanent file. If the copy is altered, the user must enter the REPLACE command to update the permanent file. If the user is not allowed to update the permanent file or if he wants to keep both the original file and the modified file, he can create a new indirect access file by entering the SAVE command and specifying a new file name.

The system allocates mass storage for indirect access permanent files in blocks of one or more PRUs. Each PRU contains 64 central memory words. The block size allocated for indirect access files can be much smaller than the block size of direct access files (refer to appendix D). Thus, to avoid using more mass storage than is actually needed and more system resources than are necessary, the user should make small files indirect access permanent files.

Direct Access

A direct access file is a permanent file which can be linked directly to the user's job. The user creates a direct access file by entering the DEFINE command. Unlike indirect access files, the system does not create a copy of the permanent file for user access. The user enters the ATTACH command to gain access to a direct access file. When entering the ATTACH command, the user can reference the file by a name other than its permanent file name. The user should be careful when modifying a direct access file because data is written directly on the permanent file rather than on a copy.

Direct access files have a write interlock feature; that is, if a user has attached the file in write mode, it cannot be attached by another user. Similarly, a user cannot attach the file in write mode until all users currently accessing the file have released it. Attaching the file in modes other than write mode can also affect the use of that file by others. Refer to the NOS Reference Manual, volume 1, for a complete discussion of mode restrictions on direct access files.

The system allocates mass storage for direct access permanent files in large blocks (refer to appendix D). Therefore, direct access files are generally used for large files such as data base files.

Direct access files usually reside on disk. However, if the installation has a Mass Storage Facility (MSF), some direct access files are stored there. MSF is suited for storing large direct access files that are accessed infrequently. The user can determine where a file is located by entering:

CATLIST,LO=F,FN=pfn.

Refer to section 5 for more information on permanent file commands.

TEMPORARY AND LOCAL FILES

A temporary file is either a nonpermanent file created by the user or a copy of an indirect access permanent file. The user can create, access, and release temporary files while logged into the system. All temporary files are released when the user logs out of the system. Refer to section 5 for a full description of the commands used to manipulate files.

A local file is any file assigned to a user's job. This includes all temporary files and all direct access permanent files assigned to the job. The parameter lfn on most time-sharing job commands and permanent file commands (refer to sections 4 and 5) signifies a local file name.

One type of temporary file, the primary file, has special significance in certain time-sharing commands. A primary file can be a new file created with the NEW command or a copy of an existing indirect access permanent file retrieved with the OLD or LIB command. Only one primary file is active or available to the user at any given time. For example, the command:

OLD,LFN1

retrieves a copy of indirect access permanent file LFN1 from permanent file storage for use as the primary file. If the user enters the command:

OLD,LFN2

a copy of indirect access permanent file LFN2 is retrieved for use as the primary file, and the previous primary file (LFN1) is released. To access LFN1 again, the user must reenter:

OLD,LFN1

or enter:

GET,LFN1

If the GET command is used, a copy of permanent file LFN1 is retrieved for use as a temporary file, and LFN2 remains the current primary file.

Most operations performed on files by time-sharing terminal commands are performed on the primary file unless another local file is specified in a command parameter. For example, the command

SAVE

retains the primary file in the permanent file system. However, the command

SAVE,ABC

retains file ABC in the permanent file system. File ABC could be either the primary file or another temporary file.

There is a close association between the primary file and the subsystem in use when the primary file is saved. Unless the null subsystem is active when the primary file is saved, or unless SS=NULL is specified in the SAVE command, the system sets an internal indicator called the subsystem flag indicating subsystem is in use. That subsystem is associated with the permanent file and is automatically selected each time the file is retrieved for use as the primary file. For example, if a user operating under the BASIC subsystem issues the command:

SAVE,ABC

where ABC is the current primary file name, the system associates the BASIC subsystem flag with resulting permanent file ABC. When the user is subsequently operating under another subsystem and enters the command:

OLD,ABC

the current primary file is released, the BASIC subsystem is selected automatically, and file ABC becomes the new primary file. To save the primary file without a subsystem association, the user must either enter the null subsystem before issuing the SAVE command or include SS=NULL in the SAVE command. Retrieving a file saved under the null subsystem does not affect the subsystem in use. The subsystem flag is set only when the user saves the primary file, or when he enters the SAVE or CHANGE command with the SS parameter. If the permanent file named in the command to select a new primary file does not have a subsystem associated with it, the subsystem currently being used remains in effect.

Before any operation on a primary file, the file is automatically rewound (positioned at BOI). However, the user is responsible for the position of all other local files because they are not automatically rewound before each operation. The REWIND command from a time-sharing terminal, the REWIND statement in FORTRAN, or the RESTORE statement in BASIC can position a file to BOI.

Information (program statements, text, or data) entered from the terminal keyboard is normally stored in the primary file. However, the user can also enter information into any local file, including the primary file, or create new local files using Text Editor (EDIT) or XEDIT. (The Text Editor and XEDIT commands are listed in appendixes E and F, respectively.) Refer to the Text Editor Reference Manual or the XEDIT Reference Manual for additional information.

FILE SORTING

IAF sorts a file by line numbers; a file without line numbers cannot be sorted. A sort places the lines of data in numerical order according to the first five digits of the line numbers. To sort a file using more than the first five digits, the user must specify the number of digits with the NC parameter of the SORT,ln command described in section 4.

In addition to ordering the line numbers, a sort packs the file. When entering information from the terminal, the user normally enters the line number and the data. After the user presses **CR**, the system places this text in central memory. Periodically, the system appends this text to the user's primary file on mass storage as new logical records. During a sort, the system packs these records by rewriting them into one logical record. Thus, the system stores the entire input as one logical record.

The user can enter line numbers manually or can use the AUTO command to generate them (refer to Terminal Control Commands in section 4). The line numbers generated by the AUTO command consist of only five digits. When creating a file that has line numbers, the following rules apply:

- Each line of information must have a space between the line number and the information, unless the user is entering a BASIC program. The space is optional in BASIC programs.
- Line numbers for FORTRAN and BASIC programs cannot be more than five digits; line numbers for all other lines of information can be up to 10 digits.
- Leading zeros on line numbers are permissible but are not required (00120 and 120 are interchangeable). Line numbers with leading zeros can be intermixed with line numbers containing no leading zeros.

The system normally sorts the primary file whenever it is possible that it will read the file. The system maintains an internal indicator called the sort flag which, when set, indicates that the primary file needs sorting. Entering a numbered line of information when not in text mode sets the flag. The system clears it just before it sorts the primary file. Before processing any command which might cause the reading of the primary file, the system examines the sort flag and initiates a sort if necessary.

If the user wishes to artificially cause or inhibit sorting, he may use the SORT command (which sets the flag) or the NOSORT command (which clears the flag). The user should enter the NOSORT command only when he does not want the primary file sorted and written into one logical record. The user can reset the sort flag by either entering a numbered line of data or by entering the SORT command.

The following example illustrates the effect of the NOSORT command.

```
old,a  
list  
79/11/02. 13.59.04.  
PROGRAM A  
  
00100 PROGRAM TEST(OUTPUT)  
00110 DO 1 J=1,10  
00120 PRINT 2,J  
00130 2 FORMAT (2X,I2)  
00140 1 CONTINUE  
00150 STOP  
00160 END
```

READY.

```
00120 k=j*j  
00130 print 2,k  
00140 2 format (2x,i2)  
00150 1 continue  
00160 stop  
00170 end
```

nosort
READY.

list, r

79/11/02. 14.03.09.
PROGRAM A

```
00100 PROGRAM TEST(OUTPUT)
00110 DO 1 J=1,10
00120 PRINT 2,J
00130 2 FORMAT (2X,I2)
00140 1 CONTINUE
00150 STOP
00160 END
--EOR--
00120 K=J*j
00130 PRINT 2,K
00140 2 FORMAT (2X,I2)
00150 1 CONTINUE
00160 STOP
00170 END
--EOR--
READY.

sort
READY.
list.r
```

The user obtains a copy of permanent file A.

The user lists file A.

The user enters changes.

The user enters the NOSORT command so that the system does not automatically sort the file.

The user enters the LIST command to list the file including all end of record indicators (—EOR—).

IAF lists the unsorted file, indicating the end of each logical record.

The user enters the SORT command so that sorting resumes.
The user lists the file.

79/11/02. 14.03.50.
PROGRAM A

IAF sorts the file and then lists it showing
that the changes have been incorporated.

```
00100 PROGRAM TEST(OUTPUT)
00110 DO 1 J=1,10
00120 K=J*j
00130 PRINT 2,K
00140 2 FORMAT (2X,I2)
00150 1 CONTINUE
00160 STOP
00170 END
--EOR--
```

READY.

When sorting, the user often encounters the following problems:

- The system attempts to sort an empty file or a file that has a line beginning with a nonnumeric character. The system issues the message:

FILE NOT SORTED.

The system does not complete the sort and clears the sort flag. The user may enter the SORT,lnf command for more informative diagnostics:

EMPTY SORT INPUT FILE.

or

NO LINE NUMBER ON SORT FILE.

- The system attempts to sort a file that is too long for the system to sort automatically. The system issues the message:

FILE TOO LONG TO BE SORTED.

The user should then enter the SORT,lnf command before proceeding.

- The user loses blank lines when the system sorts his file. During a sort, the system automatically deletes a line consisting of a line number followed only by a line terminator or a line number followed only by one trailing space and a line terminator. The system retains lines with more than one trailing space. (The RESEQ command retains at most one trailing space on a blank line. Consequently, a RESEQ command followed by a sort causes the system to delete all blank lines from the user's file.)

LOCAL FILE CONTROL

The OLD, NEW, and LIB commands change the status of temporary files by substituting another file for the current primary file. If the user enters the OLD command, a copy of an indirect access permanent file becomes the primary file. If the user enters the NEW command, the system creates a new primary file. The user can then write data on that file from the terminal. If the user enters the LIB command, a copy of an indirect access permanent file, saved under the user number LIBRARY, becomes the primary file. If the user enters the OLD, NEW, or LIB command without the ND parameter, the system releases the primary file. If the user specifies the ND parameter, the current primary file becomes a nonprimary local file, unless the specified file is already the primary file.

Entering the OLD, NEW, or LIB command affects other local files. The system releases these files unless the user specifies the ND parameter with the command. The following is a list of sample commands and their effects on files.

Files Present at Completion of Command

<u>Commands</u>	<u>Primary</u>	<u>Other Local Files</u>
BASIC,OLD,A	A	
GET,B	A	B
GET,C	A	B,C
ATTACH,D	A	B,C,D
FTNTS,OLD,A1 (local files are dropped)	A1	
RUN (program A1 creates TAPE1)	A1	
.		
RUN COMPLETE. (system response)	A1	TAPE1
.		
GET,B1	A1	TAPE1,B1
GET,C1	A1	TAPE1,B1,C1
SAVE,TAPE1	A1	TAPE1,B1,C1
FTNTS,OLD,A2/ND (local files are retained by using ND parameter; A1 becomes a nonprimary temporary file)	A2	TAPE1,A1,B1,C1
RUN (program A2 creates TAPE2)	A2	TAPE1,A1,B1,C1
.		
RUN COMPLETE. (system response)	A2	TAPE1,TAPE2,A1,B1,C1

Other commands that change the status of local files are:

BYE	GOODBYE	LOGIN	RETURN
CLEAR	HELLO	LOGOUT	

Each command releases some or all local files depending on the format of the command used. The logout commands (BYE, GOODBYE, HELLO, LOGIN, and LOGOUT) release all local files. Different forms of the CLEAR and RETURN commands allow the user to return specified files, all files, or all files except those specified in the command. Section 4 contains complete descriptions of these commands.

SYSTEM COMMANDS

4

INTRODUCTION

After the user has successfully logged into the system, he can enter the commands necessary to process his job. The four general categories of commands available to the user are:

- Terminal control commands.
- Subsystem selection commands.
- Time-sharing commands.
- Permanent file commands.

The first three categories are discussed in this section. Permanent file commands are described in section 5.

COMMAND PROCESSING

The minimum number of characters required to specify a command depends upon the specific command and the subsystem currently being used. Three characters are the minimum number checked when a command is entered in all subsystems except the batch subsystem. In the batch subsystem, commands cannot be abbreviated. In all subsystems if more than the minimum number of characters are entered, the system checks the number given, up to a maximum of seven characters.

For example, if a user is operating under the BASIC subsystem and wants to change to FTNTS, the following are legal and illegal commands for entering FTNTS.

<u>Legal Forms</u>	<u>Illegal Forms</u>
FTN	F
FTNT	FT
FTNTS	FTNx†

However, if the user is in the batch subsystem and wants to change to any other subsystem, the name of that subsystem must be entered without abbreviation.

In general, the system processes each command by checking each of the characters entered. If the user does not enter a sufficient number of characters to make the command unique, the system responds:

COMMAND NOT UNIQUE.

and the user must then reenter the command using a sufficient number of characters to make it unique. This only occurs with commands in which the first several characters are identical, such as HELP and HELLO. In this case, the entry of HEL would result in the command being declared not unique.

†x is any alphanumeric character except T.

If the command entered is not a valid command or is misspelled, the message:

ILLEGAL COMMAND.

is returned. The command entered can be up to 80 characters. A longer command causes the message:

COMMAND TOO LONG.

to be issued. If the system inserts a \$ at the beginning of the command, the character in the 80th position is truncated. The system adds a \$ to ensure that the command is processed as a command rather than executed as a local file with the same name as the command. When the user discovers an error in the command or other input before **(CR)** is pressed, he can either delete the entire line or backspace to the point of the error, correct it, and continue on the same line.

To delete a line, press the ESCAPE key or the CNTL [keys on ASCII code terminals. Press the ATTN key on correspondence code terminals. The system responds

DEL

and positions the carriage at the first character position of the next line.

To backspace and correct a line, count the number of characters and spaces from the error to the present position of the carriage (include the mistyped character) and press the backspace key an equal number of times. Then type the correct character and retype all characters and spaces after the error. The backspace key on ASCII code terminals is the BACK SPACE key or, on terminals without a BACK SPACE key, the CNTL H keys. The BACK SPACE key is used on all correspondence code terminals.

If the user attempts to perform an operation for which he is not validated or if he exceeds his validation limits, the system responds:

ILLEGAL USER ACCESS.

The system maintains validation controls for every user number (refer to LIMITS Command at the end of this section for a complete description).

Alphabetic characters entered in lowercase are automatically translated to uppercase unless the ASCII command is in effect. The ASCII command inhibits translation from lowercase to uppercase. All commands may be entered in uppercase or lowercase in either ASCII or normal character mode.

On all commands that require a numeric value, the number is treated as follows:

- If the number contains an 8 or 9, decimal base is assumed.
- The base may be specified by a postradix of B or D; that is, 562D is treated as 562 decimal and 562B is treated as 562 octal.
- Default conversion of numbers is normally decimal except for addresses and field lengths.

TERMINAL CONTROL COMMANDS

The terminal control commands allow the user to change the characteristics of the terminal and to vary the source and format of information given to and received from the system. These commands, which can be entered at any time after the user has successfully logged in, are as follows:

<u>Command</u>	<u>Description</u>
ASCII	<p>Selects the full ASCII or correspondence character set. This command causes subsequent characters entered from the terminal to be translated into 6/12 display code. On an ASCII code terminal, this code set represents the ASCII 128-character set. On a correspondence code terminal, the code set normally represents all graphics defined for that terminal. The standard character set contains only the first 63 or 64 of these 95 characters.</p> <p>The system recognizes all characters of the full ASCII code set except line feed (LF) and delete (DEL). The input control characters for return and backspace are recognized but are not translated. Characters of the standard character set are stored as 6-bit display code characters. The additional characters which make up the full ASCII code set are stored as 12-bit display code characters with an escape code convention. Refer to appendix A for an explanation of how the system interprets characters in both ASCII and normal character modes.</p> <p>The ASCII command must be entered if lowercase letters are to be interpreted by the system. In normal character mode, all lowercase letters are translated to uppercase. Commands may be entered in uppercase or lowercase in either ASCII or normal character mode.</p>
AUTO,nnnnn,iiii	<p>Automatically generates five-digit line numbers. The nnnnn parameter specifies the beginning line number; default value is 00100. The iiii parameter specifies the increment value added for each succeeding line number; default value is 10. The user can exit from auto mode by using either of the following methods.</p> <ul style="list-style-type: none">• Delete the current line by pressing the ESCAPE key or CNTL [keys (ASCII code terminals) or ATTN key (correspondence code terminals) and then enter a new command on the next line.• Backspace six character positions and then enter a new command on the same line (six backspaces are required to overwrite the five-digit line number and the blank that follows).

NOTE

If a correspondence code terminal is being used, the ASCII command must be entered to enable use of the full correspondence code set.

<u>Command</u>	<u>Description</u>
	The line numbering sequence can be altered by deleting the line or backspacing and then entering a new beginning line number rather than a new command (leading zeros are permissible but not required). The user should exercise caution when doing this since the AUTO command is still in effect and continues generating line numbers using the original increment value. Thus, if a line number is generated that already exists in the file, the original contents of that line are lost and must be reentered. The increment value cannot be altered unless a new AUTO command is entered.

In the following example of the AUTO command, the user backspaces over the line number and enters an LNH command (refer to Time-Sharing Job Commands in this section for a description of the LNH command).

```

auto
00100 this is a test
00110 for the auto command
00120 end
00130-----lnh†

00100 THIS IS A TEST
00110 FOR THE AUTO COMMAND
00120 END
READY.

```

In the following example, the user backspaces to change the numbering sequence and later deletes a line to change the numbering sequence.

```

new,kl

READY.
auto,10,10
00010 this is a test
00020 for the auto command
00030-----00100 test†
00110 test
00120 *DEL*
00200 test
00210 end
00220 *DEL*
lnh

00010 THIS IS A TEST
00020 FOR THE AUTO COMMAND
00100 TEST
00110 TEST
00200 TEST
00210 END
READY.

```

†← 's indicate number of backspaces.

<u>Command</u>	<u>Description</u>
BRIEF	Suppresses all full and partial headers such as those issued by the LIST or RUN command.
CSET,c	Sets the character set mode of the terminal to the specified mode. c Specifies terminal character set mode.
	ASCII ASCII mode (ASCII 128-character set or all correspondence code graphics)
	NORMAL Normal mode (standard 64- or 63-character set)
	The CSET command may also be entered through control statements included in a procedure file.

NOTE

The CSET,NORMAL command sets only the terminal character set. It does not have an effect on the AUTO, PARITY, or TAPE commands.

FULL	Selects full-duplex mode. Under this mode, each character received by the system is echoed to the terminal just as received. This mode is effective only for terminals which have a full-duplex capability. The system responds READY.
	If the HALF/FULL duplex switch on the terminal is in the HALF position when this command is entered, each subsequent character entered is double printed (initially when it is typed and again when it is echoed by the system). Placing this switch in the FULL position allows only the character echoed to the terminal to be printed.
HALF	Clears full-duplex mode (refer to FULL command). Characters received by the system after this command is issued are not echoed to the terminal. The system responds READY.

If switch is in the FULL position, characters entered at the terminal are not printed. Only system-generated output appears at the terminal. Placing this switch in the HALF position enables keyboard entry to be printed.

<u>Command</u>	<u>Description</u>
NORMAL	Reverses the effect of the ASCII, AUTO, BRIEF, CSET, NOSORT, PARITY, and TAPE commands on both input and output. The system initially assumes that this command has been entered. Normal mode uses the standard 64- or 63-character set. All lowercase letters are converted to uppercase (refer to ASCII command for additional information) and all command headers are printed (refer to BRIEF command).
PARITY,p	<p>Sets terminal parity where p is either</p> <p style="padding-left: 40px;">ODD to set odd parity</p> <p style="padding-left: 40px;">or</p> <p style="padding-left: 40px;">EVEN to set even parity</p> <p>If no parameters are supplied, odd parity is assumed.</p> <p>The system initially assumes that all information is transmitted in even parity to ASCII code terminals and in odd parity to correspondence code terminals.</p>
ROUT,nn	<p>Sets the amount of time required to perform the return function (carriage return and line feed) for an ASCII code terminal. This command is not valid for correspondence code terminals. The nn parameter specifies a character count delay. This is the amount of time required by the system to send nn characters to the terminal, where the value of nn can range from 0 to 30. If 0 is specified, a standard delay value is used. The system responds</p> <p style="padding-left: 40px;">READY.</p> <p>This command is necessary since the length of time required for the return operation varies depending upon the type of terminal being used. For example, if the system is transmitting output to a terminal, a fixed amount of time is allowed for the return function, after which the system sends the next line of output. If the amount of time is not sufficient, characters may be printed during carriage return function.</p> <p>The delay is accomplished by sending rubout characters for paper tape operations and null characters for all other I/O operations. The number of characters to be sent is initially set by the system at login or at recovery time. If a ROUT command is entered, the number of characters specified for the delay remains in effect until termination of the session (logoff), until recovery, or until another ROUT command is entered.</p>

<u>Command</u>	<u>Description</u>
TAPE	Permits subsequent information to be read from the paper-tape reader at an ASCII code terminal. The system sends an X-ON character to the terminal at the end of each program request for data and after execution of each command. The X-ON character turns on the paper-tape reader at the user's terminal. This mode also inhibits the output of header messages from the LIST command and the READY message after the execution of most commands. The system ignores carriage return characters entered on an empty input line while in tape mode. Refer to section 9 for complete information concerning paper-tape operations.
TERM,t	Allows the user to redefine the terminal characteristics that were established at login. The t parameter specifies the new terminal characteristics as follows:
TTY	ASCII code terminal with standard print.
COR	Correspondence code terminal with standard print.
CORAPL	Correspondence code terminal with APL print.
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print.
BLKEDT	Block transmission (ASCII code) terminal with full display screen editing capability (available only on certain terminals such as the Hazeltine 2000 terminal) and standard print.

This command is useful only for terminals with both standard and APL print capabilities. The user can change the print type and then indicate the change by means of the TERM command.

For example, a user operating a Memorex 1240 terminal changes the print type from standard to APL. To notify the system, he enters TERM, MEMAPL.

NOTE

All ASCII-code terminals with standard print are designated TTY. The Memorex 1240 is an ASCII code terminal and therefore is identified as TTY when using standard print.

Terminal characteristics may be selected by default. Enter the LIMITS command to obtain the default value.

<u>Command</u>	<u>Description</u>
TIMEOUT	Changes a no-timeout terminal to the standard timeout status. In standard timeout status, the user is automatically logged out after 10 minutes of inactivity. Standard status is in effect when bit 10 in access word AW is clear. (Refer to LIMITS Command in this section.) When the bit is set, the terminal remains connected until the user logs out. The TIMEOUT command clears this bit for the session in progress.

SUBSYSTEM SELECTION COMMANDS

The following commands select a specific subsystem. The user should always be aware of the subsystem that is currently active. For example, attempting to execute a FORTRAN program while operating under the BASIC subsystem causes meaningless diagnostic messages to be issued. To determine which subsystem is currently active, enter the ENQUIRE command.

NOTE

The subsystem may be automatically selected by the system at login if this has been set in the validation file.

A specific subsystem can be associated with an indirect access file so that whenever the user specifies that file as the primary file, the associated subsystem is selected automatically. A subsystem flag can be set when the user saves the file. If the user includes the SS=subsyst parameter on the SAVE command, he can specify any of the valid subsystems. If he enters SAVE without SS=subsyst and the file is the primary file, the subsystem flag that associates the current subsystem with the file is set. To save the primary file with no subsystem association, either the user can enter the null subsystem before he saves the file or he can enter the command.

SAVE,lfn/SS=NULL

Automatic subsystem association is made only when the primary file is saved. To associate a subsystem with a temporary file other than the primary file, the user must specify the SS=subsyst parameter on the SAVE command.

<u>Command</u>	<u>Description</u>
ACCESS	Selects the access subsystem for validated users. In this subsystem the user can communicate with another time-sharing user. When a validated user enters ACCESS, the system responds

READY.

<u>Command</u>	<u>Description</u>
BASIC,ccc	Selects the BASIC subsystem. The optional ccc parameter enables the user to specify one command in addition to the BASIC command. Any valid command is permitted, as well as all valid parameters for that command. The following example illustrates the use of the ccc parameter.
	BASIC,OLD,PRIME
	In this example, the user selects the BASIC subsystem and makes permanent file PRIME the primary file. The file name (PRIME) is a valid parameter with the OLD command.
	If ccc is omitted and no primary file is currently defined, the system responds
	OLD, NEW, OR LIB FILE:
	The user can then specify whether the primary file is to be a new file (NEW) or an existing indirect access permanent file (OLD or LIB). The file name (separated by a comma) also can be included in this entry. The entry is terminated by CR . If the file name is omitted in the reply, the system responds:
	FILE NAME:
	The user then specifies the file name.
BATCH,nnnnn	Selects the batch subsystem. Refer to section 8 for a description of this command.
EXECUTE,ccc	Selects the execute subsystem. This subsystem is used only to execute a previously compiled (object code) program. The RUN (or RNH) command must be entered to initiate execution. The optional ccc parameter enables the user to specify one command in addition to the EXECUTE command. Any valid command is permitted, as well as all valid parameters for that command. The following example illustrates the use of the ccc parameter.
	EXECUTE,OLD,OBJFILE
	In this example, the user selects the execute subsystem and makes permanent file OBJFILE the primary file. The file name (OBJFILE) is a valid parameter with the OLD command.
	If ccc is omitted and no primary file is currently defined, the system responds:
	OLD, NEW, OR LIB FILE:
	The user can then specify whether the primary file is to be a new file (NEW) or an existing indirect access permanent file (OLD or LIB). The file name (separated by a comma) also can be included in this entry. The entry is terminated by CR . If the file name is omitted in the reply, the system responds:
	FILE NAME:

<u>Command</u>	<u>Description</u>
	The user then specifies the file name.
NOTE	
Source language programs cannot be executed under the execute subsystem.	
	<p>The execute subsystem should be used whenever possible to conserve system resources. Efficiency can be achieved when certain programs are used frequently. For example, a source code program created under FORTRAN must first be compiled before it can be executed. This process is automatic. The user need only enter the RUN command. The program automatically goes through a compilation phase which produces an executable object code program. The object code program is then executed. Thus, greater efficiency is achieved by retaining the object code program in the permanent file system for later execution under the execute subsystem. This can be accomplished by using the following commands:</p>
RUN,B=lfn	Causes the object code program to be placed on temporary file lfn.
SAVE,lfn/ SS=EXECUTE	Retains file lfn as a permanent file and sets the execute subsystem flag.
	If this is done, subsequent requests to the file using the OLD command cause the execute subsystem to be selected. Entering the RUN command initiates execution of the object code program.
FORTRAN,ccc	Selects the FORTRAN subsystem, which uses the FORTRAN version 5 compiler. The optional ccc parameter enables the user to specify an additional command. Only one of the valid commands and its parameters can be specified. For example:
	FORTRAN,NEW,TAX
	selects the FORTRAN subsystem and creates a new primary file named TAX. The file name (TAX) is a valid parameter with the NEW command.
	If ccc is omitted and no primary file is currently defined, the system responds:
	OLD, NEW, OR LIB FILE:
	The user can then specify whether the primary file is to be a new file (NEW) or an existing indirect access permanent file (OLD or LIB). The file name (separated by a comma) also can be included in this entry. The entry is terminated by CR . If the file name is omitted in the reply, the system responds:
	FILE NAME:
	The user then specifies the file name.

<u>Command</u>	<u>Description</u>
FTNTS,ccc	Selects the FTNTS subsystem, which uses the FORTRAN Extended version 4 compiler. The optional ccc parameter enables the user to specify an additional command. Only one of the valid commands and its parameters can be specified. For example:
	FTNTS,OLD,DATAGEN
	selects the FTNTS subsystem and makes permanent file DATAGEN the primary file. The file name (DATAGEN) is a valid parameter with the OLD command.
	If ccc is omitted and no primary file is currently defined, the system responds:
	OLD, NEW, OR LIB FILE:
	The user can then specify whether the primary file is to be a new file (NEW) or an existing indirect access permanent file (OLD or LIB). The file name (separated by a comma) also can be included in this entry. The entry is terminated by $\textcircled{6}$. If the file name is omitted in the reply, the system responds:
	FILE NAME:
	The user then specifies the file name.
NULL	Selects the null subsystem. The system responds:
	READY.
	This command is entered before saving the primary file if the user does not want a specific subsystem flag to be associated with the file. Normally, when a user saves the primary file, an internal indicator called the subsystem flag is set to indicate the subsystem currently in use. That subsystem is automatically selected in each succeeding request for the file using the OLD command. An alternate way of saving the primary file with no subsystem association is to specify the SS=NULL parameter on the SAVE command (section 5). No automatic subsystem association occurs when saving temporary files other than the primary file.
	If the user does not specify a subsystem at login time, the system sets the subsystem specified in the validation file (refer to the IS field under LIMITS Command in this section). Null is the default.
TRAN,termnam	Selects the transaction subsystem for validated users. The termnam parameter specifies the name of the terminal. In this subsystem, the user can communicate with the Transaction Facility (TAF/TS). Operation of TAF/TS is detailed in the TAF/TS Reference Manual.

TIME-SHARING JOB COMMANDS

The time-sharing job commands allow the user to perform a variety of job-processing or job-related operations. These commands can be entered at any time after the login procedure is complete.

<u>Command</u>	<u>Description</u>
APL	Selects the APL (A Programming Language) interactive interpreter. This command also specifies translation to the full 128-character code set. All characters are recognized although input control characters are not processed. Rather, they are passed to the program as data. To leave APL control, the user must enter the following using all capital letters:)SYSTEM. Refer to APL Reference Manual for a list of terminal identifiers and information concerning the use of APL.
BEGIN,pname,pfile, P1,P2,...,Pn or -pname,pfile,p1, P2,...,Pn	Initiates processing of a CYBER Control Language (CCL) procedure. This allows the user to process a procedure file and remain interactive with his job. A procedure file is a source file that contains control statements and/or control language statements. It has the same structure as the control statement record in batch job, although the job statement and USER statement required in batch jobs cannot be included. Although the command format and short parameter descriptions are given here, refer to the NOS Reference Manual, volume 1, for information about CCL procedures and for a listing of CCL dayfile messages.
	The pname and pfile parameters are order dependent. If the user does not specify a parameter, a default value is assumed. Parameters pi are optional.
pname	Name of the procedure to be processed. The default is the next procedure on file pfile. If the default is used and pfile is at end-of-information, CCL rewinds pfile and calls the first procedure. An exception occurs if pfile is INPUT; in this case, pfile is not rewound.
pfile	Name of the file on which procedure pname is located. The default is an installation-defined file name (the released default is file PROCFILE). CCL first searches for a local file named pfile. If pfile is not local, CCL searches for an indirect access file named pfile and makes a local copy of it. CCL returns this copy of pfile at the end of the procedure pname unless it was local before the user entered the BEGIN command.
pi	A parameter having one of the following forms. Order dependent forms:
keyword	A keyword that is the same as a keyword used in the procedure header statement. This can be used to specify an alternate value substitution.
keyword=	Specifies null substitution for the keyword which appears in the procedure header statement.

<u>Command</u>	<u>Description</u>
value	A value of from 1 to 40 characters, or null. To include nonalphanumeric characters, except slash or leading minus, value must be a literal (\$-delimited).
Order independent form:	
keyword=value	The value is substituted for the keyword which appears in the procedure header statement.
To call procedure pname on file pfile, CCL searches for file pfile and then for procedure pname. If either pfile or pname is not found, the job step aborts.	
To create a CCL procedure file from the terminal, the user enters text mode (refer to the TEXT command), uses Text Editor (refer to the EDIT command), or uses XEDIT (refer to the XEDIT command). A CCL procedure file cannot contain line numbers. After leaving text mode, the user can retain the file for future reference by using the SAVE command. The user can make changes to the procedure file using Text Editor or XEDIT.	
NOTE	
Other relationships exist between the BEGIN command and the procedure header statement which can cause the command to abort. Refer to the NOS Reference Manual, volume 1, for a complete description.	
BINARY,nnnn,ccc	Selects binary input mode. Under control of this command, 8-bit characters entered from the terminal are stored, without translation, in the primary file. Normally, characters entered from the terminal are translated and stored as 6-bit or 12-bit display codes (refer to Central Memory Character Equivalents for Data Input in appendix A for complete information). Data entered is added to end-of-file.
nnnn	Number of characters (1 through 4095) to be entered.
ccc	Termination character; the three-digit octal code that corresponds to the character the user selects to terminate binary input mode. Refer to appendix A for the correct character set and corresponding codes. (If a correspondence code terminal is being used, a value of 200 must be added to the octal code listed in appendix A.)
The termination character enables the user to return to the normal input mode before the number of characters specified by nnnn are entered.	

<u>Command</u>	<u>Description</u>
	Example:
	BINARY,20,207
	In this example, binary input mode is terminated when 20 characters are entered or the ASCII character BELL (octal code 207) is typed.
	NOTE
	If the termination character (ccc) is omitted, the number of characters specified by nnnn must be entered to terminate binary input mode.
BYE	Logs the user out of the system and disconnects the terminal. The system responds:
	usernum LOG OFF hh.mm.ss.
	usernum SRU s.sss UNTS.
	usernum User number.
	hh.mm.ss. Time of day this command was entered.
	s.sss Measure of system resources used during this session. The SRU is a unit of measurement which represents the total usage of the system by the user. It includes all central processor unit (CPU) time, memory usage, and I/O activity.
	NOTE
	The user should ensure that all temporary files to be retained are made permanent before issuing this command. All temporary files which have not been saved are released when the BYE command is processed.
CALL,lfn	Initiates processing of KCL (a control language supported under previous operating systems) procedure file lfn. This allows the user to process a procedure file and remain interactive with his job.
	A procedure file is a source file which contains control statement images and/or control language statements. It has the same structure as the control statement record in a batch job although the job and USER statements required in batch jobs cannot be included.

<u>Command</u>	<u>Description</u>
Optional parameters may also be included in this command. The command format with parameters is:	
CALL,lfn,C,S=ccc(olnam ₁ =newnam ₁ ,olnam ₂ =newnam ₂ ..., olnam _n =newnam _n)	
lfn	Specifies name of the procedure file to be processed. The system searches for lfn according to the following indicated order.
	<ol style="list-style-type: none"> 1. Current local files (including primary file). 2. System-defined procedure files. This can be a procedure file created by the installation and included as part of the system. 3. User's permanent file catalog for indirect access permanent file.
If the specified file cannot be found, the system responds:	
lfn NOT FOUND.	
C	Replaces the control statement record after the CALL command with lfn.
S=ccc	Specifies control statement ccc as the first statement in the procedure file to be executed.
olnam _i = newnam _i	Directs that olnam be replaced with newnam for each occurrence of olnam in the procedure file. This allows use of the same procedure file to perform several tasks, depending on the parameters specified.
To create a procedure file from the terminal, the user enters text mode (refer to the TEXT command), uses Text Editor (refer to the EDIT command), uses XEDIT (refer to the XEDIT command), or enters auto mode (refer to the AUTO command). Lines within a procedure file may contain line numbers to make maintenance easier. However, the CALL command removes the line numbers before the procedure file is processed. The file can be retained for future reference as an indirect access permanent file by using the SAVE command. If the user wants to change the procedure file he must use Text Editor or XEDIT unless the file has line numbers. Editing files with line numbers is explained in section 6 under Program Editing.	
Refer to the NOS Reference Manual, volume 1, for information concerning the control statements and control language statements available.	

<u>Command</u>	<u>Description</u>						
CHARGE,chargeno, projectno	Allows the user to specify a valid charge number and project number to be billed for subsequent terminal operations. Certain users are required to enter a charge and project number during the login sequence; all users may issue the CHARGE command as long as the charge and project numbers entered are valid. This command can be entered at any time during the time-sharing session. Both charge number (chargeno) and project number (projectno) parameters are required. The number of system resource units (SRUs) the system used in processing the CHARGE command is printed at the terminal in the BASIC, FORTRAN, FTNTS, or execute subsystem. In the batch subsystem, the last entry in the usage summary is printed at the terminal. In all subsystems, the job SRU accumulator and usage summary are written on the user's dayfile, and the SRU accumulator is set to zero. The SRU is a unit of measurement which represents the total usage of the system by the user. It includes all CPU time, memory usage, and I/O activity. The system responds:						
	READY.						
	For greater security, the user can enter the CHARGE command with no parameters. Refer to Login Sequence in section 2 for more information on system response.						
CLEAR or CLEAR,*,lfn ₁ , lfn ₂ ,...,lfn _n	The first format of CLEAR releases all local files. The second format releases all local files except lfn ₁ , lfn ₂ ,...,lfn _n . If the user specifies no files (CLEAR,*), the command releases all local files. Neither format releases the CCL work files ZZZZZC0, ZZZZZC1, or ZZZZZC2.						
CONVERT	Converts input from one character set to another. For more information, refer to appendix A.						
(CR)	Causes the system to reply with an abbreviated status response.						
	<table> <tbody> <tr> <td>IDLE</td><td>System is waiting for user to enter command. (No command is currently being processed.)</td></tr> <tr> <td>EXECUTE</td><td>Command is currently being processed.</td></tr> <tr> <td>WAIT</td><td>Command is waiting for system resources to become available.</td></tr> </tbody> </table>	IDLE	System is waiting for user to enter command. (No command is currently being processed.)	EXECUTE	Command is currently being processed.	WAIT	Command is waiting for system resources to become available.
IDLE	System is waiting for user to enter command. (No command is currently being processed.)						
EXECUTE	Command is currently being processed.						
WAIT	Command is waiting for system resources to become available.						
DAYFILE,parameters	Lists the system record of the user's time-sharing activity. Each line of output is in the form: hh.mm.ss. message hh.mm.ss. Time of day the message was placed in the dayfile.						

<u>Command</u>	<u>Description</u>
The formats of the command are:	
DAYFILE,L=lfn,FR=string,OP=op,PD=pd,PL=pl,I=infile (order independent)	
	or
DAYFILE,lfn,string,op,pd,pl,infile (order dependent)	
L=lfn	Specifies file on which the dayfile is to be written. If the L=lfn parameter is omitted, L=OUTPUT is assumed and the dayfile output is listed at the terminal.
FR=string	Specifies the literal string† the system is to search for in the dayfile. If the string contains any characters other than letters and numbers, including leading and embedded blanks, it must be enclosed by \$ delimiters. Any \$ within the string must be replaced with two \$'s.
NOTE	
Time-sharing commands written to the dayfile are preceded by a \$. Hence, if the user is searching for a time-sharing command, he should replace the \$ preceding the command with two \$'s, and enclose the command with \$ delimiters (for example, \$\$\$OLD\$).	
The system searches for the string only at the beginning of the field specified by the OP parameter. The portion of the dayfile from the last occurrence of the requested string to the end of the dayfile is printed at the terminal or on the file named. The first character of the time field is a space.	
OP=op	Specifies the starting field position; op can be one of the following:
T	Searches the beginning of the time field for a matching string.
M	Searches the beginning of the message field for a matching string.
I	Lists the dayfile from the last occurrence of the message USER DAYFILE DUMPED. The FR=string parameter cannot be specified.

†A literal string is a sequence of characters.

<u>Command</u>	<u>Description</u>										
F	Lists the entire dayfile if the FR= string parameter is not specified; searches the beginning of the message field if FR=string is specified.										
	If the OP=op parameter is omitted and the FR=string parameter is specified, OP=M is assumed; otherwise, OP=F is assumed.										
PD=pd†	Print density in pd lines per inch (3, 4, 6, or 8). Default is PD=6.										
PL=pl†	Page length in pl lines per page. The default varies with the print density:										
	<table border="0"> <thead> <tr> <th><u>pd</u></th><th><u>default pl</u></th></tr> </thead> <tbody> <tr> <td>3</td><td>30</td></tr> <tr> <td>4</td><td>40</td></tr> <tr> <td>6</td><td>60</td></tr> <tr> <td>8</td><td>80</td></tr> </tbody> </table>	<u>pd</u>	<u>default pl</u>	3	30	4	40	6	60	8	80
<u>pd</u>	<u>default pl</u>										
3	30										
4	40										
6	60										
8	80										
I=infile	A local file containing a terminated dayfile to be used for input instead of the active dayfile. The default is the active dayfile.										
Examples:											
DAYFILE,TEMP,COMPASS	The system searches the beginning of the message field for the string, COMPASS, and lists the dayfile, starting at the last occurrence of the string, on file TEMP.										
DAYFILE,,,\$ 09.41.23.\$,T.	The system searches the beginning of the time field for 09.41.23. and lists the dayfile from that time at the terminal. There is a blank preceding the time in the time field.										
DAYFILE,OP=I	The system prints the portion of the dayfile not listed previously at the terminal.										
DAYFILE,DAY,\$\$\$FTN\$.	The system searches the beginning of the message field for \$FTN and lists the dayfile, from the last occurrence of \$FTN, on file DAY.										

†Meaningful only if the user has the file specified by the L parameter printed on a batch line printer.

<u>Command</u>	<u>Description</u>	
	DAYFILE, \$\$\$_\$BEGIN\$.	The system lists the dayfile from the last time the time-sharing command BEGIN was used.
DEBUG,parameter	Activates, terminates, or resumes CDC CYBER Interactive Debug. When activated, it aids the user in debugging compiled programs. One of the following parameters can be specified.	
	ON	Activates debug mode. The default is ON.
	OFF	Terminates debug mode.
	RESUME	Resumes suspended debug session.
	For information on the use of the debug mode, refer to the CYBER Interactive Debug Reference Manual.	
DIAL,nnnn,sss	Sends one-line message sss to the terminal specified by terminal number nnnn. DIAL can be used only by users validated to enter the access subsystem (refer to LIMITS Command in this section). A validated user can determine the terminal number of a person with a specific user number by entering the USER command.	
The possible system responses to this command are:		
	READY.	Message has been sent.
	TERMINAL NOT ACTIVE.	Terminal to which message is directed is inactive.
	TERMINAL BUSY.	Terminal to which message is directed is currently receiving output or has an input request outstanding. The message is lost; no further attempt to transmit the message to the terminal is made.
	ILLEGAL COMMAND.	User is not validated for the command.

<u>Command</u>	<u>Description</u>
EDIT,parameters	Selects Text Editor.
	The formats of the command are:
	EDIT, FN=lfn ₁ ,M=m,I=lfn ₂ ,L=lfn ₃ (order independent)
	or
	EDIT, lfn ₁ ,m,lfn ₂ ,lfn ₃ (order dependent)
FN=lfn ₁	Specifies the name of an existing local file to be edited. The FN=lfn ₁ parameter may also specify a new temporary file name. In this case, the new file is constructed using Text Editor. If the edit file name is omitted, the primary file name is assumed. [†] When necessary, the primary file is sorted prior to text editing.
M=m	Selects mode of file processing.
ASCII or AS	ASCII mode edit file. This option indicates that the user wishes to process the file in ASCII mode. The terminal is automatically put in ASCII mode. After editing is completed, the terminal is put back in its original mode.
NORMAL or N	Normal mode edit file.
	If omitted, normal mode is assumed. In a time-sharing session if a terminal is in ASCII mode, the system adds an AS to all EDIT commands when the EDIT statement is entered with no parameters or only lfn ₁ (not FN=lfn ₁). This does not apply to EDIT statements in procedure files or EDIT statements with more than one positional parameter.

[†]If two or more parameters are specified in the command, or if an equated parameter is used, the name of the file to be edited must be specified (the primary file is not assumed).

<u>Command</u>	<u>Description</u>
I=lfn ₂	Specifies file from which EDIT commands are to be read. If omitted, commands are input from the terminal.
L=lfn ₃	Specifies file on which output is to be written. If omitted, output is returned to the terminal.

NOTE

Before entering EDIT, the user should ensure that the file to be edited consists of no more than one logical record (refer to the PACK command for information on compressing files). If the file contains more than one logical record, all records but the first are lost. For additional information, refer to the Text Editor Reference Manual. To edit a multirecord file, the user can use XEDIT.

Entry of information using TEXT, AUTO, or line-numbered source statements is possible only to the primary file. However, Text Editor allows the user to enter or modify information in all files (including the primary file). This is important when the user wants to reference a direct access file. Text Editor is one method he can use to enter source code into a direct access file.

Text Editor commands are listed in appendix E. For a description of Text Editor parameters and commands, refer to the Text Editor Reference Manual.

ENQUIRE Requests the current job status. An important feature of this command is that it can be entered during job execution. The system responds:

TERMINAL:	nnn,iii.
SYSTEM:	Subsystem currently active.
FILE NAME:	Current primary file name.
STATUS:	IDLE No job being processed. or EXECUTE User job being processed. or WAIT User job waiting for computer resources to become available.

MESSAGE: For execute or wait status, the message line contains the current job status message.^f For idle status, no message is printed.

nnn Terminal number.

iii Identifier that describes type of terminal. These identifiers are the same as those used in the TERM command and are not repeated here.

^fMany programs and utilities (such as COMPASS and MODIFY) issue job status messages indicating their progress in processing, which may be of interest to the user. Some job status messages contain system information and may be ignored by the user.

<u>Command</u>	<u>Description</u>
ENQUIRE,parameters	Requests information about the system. The formats of the command are:
	ENQUIRE,OP=p ₁ p ₂ ...p _n ,JN=ccc,O=lfn ₁ ,FN=lfn ₂
	or
	ENQUIRE,p ₁ p ₂ ...p _n
OP=p _i	Specifies any of the following options. Up to seven options can be specified (for example, ENQUIRE,BDF).
<u>p_i</u>	<u>Description</u>
A	Causes all OP options except S and T to be executed (in the order B, D, R, U, J, L, F).
B	Requests user identification and job information (user number and index, job name and sequence number, family name, pack name, primary file name, subsystem, queue and CPU priorities, maximum and last assigned field lengths).
D	Requests resource demand information. Listing describes both resources (tape units and disk packs) demanded by job and resources currently assigned to the job.
F	Requests status of the user's local files. The information on each file is the same as that listed under the LENGTH command, described in this section.
J	Requests contents of control registers, error flag fields, and succeeding control statements.

<u>Command</u>	<u>Description</u>
<u>Pi</u>	<u>Description</u>
L	Requests loader information.
R	Requests amount of resources used (CPU time, mass storage activity, magnetic tape activity, permanent file activity, SRU adder constant value, total SRUs) in decimal.
S	Requests decimal number of SRUs used. The SRU is a unit of measurement which represents the total usage of the system by the user and includes CPU time, I/O activity, and memory usage.
T	Requests accumulated CPU time for this session in decimal.
U	Requests user's current limits on resource usage (seconds, job step SRUs, account block SRUs, dayfile messages, control statements, dispose files, mass storage) in decimal.
JN=ccc	Requests the status of a job that has the same user number as that of the terminal user. ccc is the last three characters of the seven-character job identification name (jobname) the system assigns to a job. If the user enters: ENQUIRE,JN the system returns the status of all of the jobs submitted under the current user number that are in queues (input, print, punch, rollout, or timed/event rollout) or are executing. The possible replies are: jobname NOT FOUND. jobname IN INPUT QUEUE. jobname IN PRINT QUEUE. jobname IN PUNCH QUEUE. jobname IN ROLLOUT QUEUE. jobname EXECUTING. jobname IN TIMED/EVENT ROLLOUT QUEUE.

<u>Command</u>	<u>Description</u>
	The jobname NOT FOUND message usually indicates that the job has been processed and is no longer in a queue or executing. The message can mean that the job output is currently printing. However, this message will also be issued if the JN=ccc parameter is entered incorrectly. Only the status of a job submitted under the current user number can be obtained.
O=lfn ₁	Requests that output be placed on alternate file lfn ₁ . If not specified, output is returned to the terminal. If O=lfn ₁ is the only parameter specified, the system prints option A on file lfn ₁ .
FN=lfn ₂	Requests the same information as the F option but only for the file name specified by lfn ₂ . The file name must be a local file. If the file name is misspelled or is not currently associated with the user's job, the system responds:
	FILE NOT FOUND.
GOODBYE	Same as BYE command.
HELLO	Logs the current user out of the system, as in the BYE command, and reinitiates the login sequence. Any temporary files to be retained must be made permanent before the HELLO command is entered. This command allows a new user to log in without first establishing communications with the system by dialing the computer.
HELP	Allows the time-sharing user to obtain assistance in the use of system commands. If the user at a terminal is not sure of the format of a command or its use, the HELP command provides an easily accessible on-line description of the system command set. The system responds:
	IF MORE INFORMATION NEEDED, TYPE YES. OTHERWISE, TYPE THE COMMAND NEEDED. ?
	The user either types YES in reply to the request or enters the command he wishes to have described. A minimum of three characters must be entered if a command is typed. The user must type END after a question mark to terminate the HELP command.
I key	Interrupts a program that is currently transmitting output to the terminal. The system responds:
	INTERRUPTED

<u>Command</u>	<u>Description</u>
	To continue, the user either presses CR or types P followed by CR . If the TAPE command is in effect (tape mode), CR cannot be entered as the first character of a new line; in this case, only P CR can be entered to continue. If the P option is used, some data is discarded. Refer to Job Suspension in section 6 for additional information.
	Using the I key to perform this function is effective only for ASCII code terminals. The ATTN key performs this function on correspondence code terminals. (Refer to table 2-1 for complete information concerning the use of special function keys for both ASCII and correspondence code terminals.)
	If pressing the I key accidentally disconnects the terminal from the system, the user can log in again and enter the RECOVER command in response to the RECOVER/SYSTEM or RECOVER/CHARGE request. He can then continue from the point at which he was disconnected. Refer to Recovery in section 6 for additional information.
LENGTH,lfn	Requests the file name, length, type, and last status of the local file specified by lfn. If the lfn parameter is omitted, the primary file is assumed. The system responds:

LOCAL FILE INFORMATION.

FILENAME LENGTH/PRUS TYPE STATUS

lfn	length	type	status
-----	--------	------	--------

lfn Local file name.

length File length in PRUs (each PRU contains 640 6-bit characters).

type File type:

IN. Input file.

LI. Library file.

LO. Any local file that is not the current primary file or a direct access permanent file.

PM. Direct access permanent file.

PT. Primary file.

SY. System file.

An * following the file type indicates a locked file (user cannot write on a locked file).

<u>Command</u>	<u>Description</u>	
	status	File status:
		EOI File is positioned at end-of-information.
		EOF File is positioned at end-of-file.
		EOR File is positioned at end-of-record.
	I/C	File is positioned somewhere between beginning- and end-of-information.
LIB	Retrieves a copy of a permanent file from the catalog of special user number LIBRARY. Refer to LIB Command in section 5.	
LIMITS	Provides a listing of the user's current validation limits. Refer to LIMITS Command later in this section.	
LIST,lnum,R, F=lfn	Lists at the terminal the contents of the primary file unless the F parameter is specified.	
	lnum	Specifies line number of the line where listing is to begin. This optional parameter can be specified only if the primary file is being listed. The file should be sorted to obtain correct results.
	R	Specifies that end-of-record and end-of-file marks, if present, are to be indicated in the listing. This optional parameter can be specified only if listing the primary file. The primary file is listed from the beginning-of-information to the end-of-information unless the lnum parameter is specified.
	F=lfn	Specifies listing of local file lfn (optional). The file is not sorted before listing. When this parameter is specified, neither the lnum nor the R parameter can be supplied. The local file specified is listed from the current position to the end-of-information.

The system replies:

yy/mm/dd.	hh.mm.ss.	}	header
PROGRAM nnnnnnnn			
.			
.			
(program lines)			
.			
READY.			

<u>Command</u>	<u>Description</u>
	The header information printed is
	yy/mm/dd. Current date (year/month/day).
	hh.mm.ss. Current time (24-hour format).
	nnnnnnn File name.
	This header is not displayed under the batch subsystem.
LNH,lnum,R, F=lfn	Optional form of the LIST command which lists the file without a header.
LOGIN	Same as HELLO command.
LOGOUT	Same as BYE command.
MONITOR,nnnn	Connects the validated user in the access subsystem with the terminal specified by terminal number nnnn. (For information on obtaining another user's terminal number, refer to the USER command.) All input from and output to terminal nnnn is received by the monitoring terminal. Binary input and output cannot be monitored.
	The possible responses to this command are
	TERMINAL NOT ACTIVE. The terminal the user requested is inactive.
	INVALID TERMINAL. The terminal the user requested is already being monitored by another user or is the terminal the user is at.
	ILLEGAL COMMAND. The user is not validated to enter this command (refer to LIMITS Command in this section.)
	When one user is monitoring another, he can communicate directly with the other user by entering text whenever the other user is not receiving output. When a carriage return is entered, the line of text is sent to the user being monitored. This allows easy two-way interaction between users. If the characteristics of the user's terminals are not the same, monitoring may not be successful.
	Monitoring of another user terminates when the user enters STOP.
	The system responds:
	END MONITORING.

<u>Command</u>	<u>Description</u>
NEW	<p>Allows the user to create a new primary file. The system responds:</p> <p>FILE NAME:</p> <p>The user enters a valid file name. When the system accepts the new file name, it replies:</p> <p>READY.</p> <p>The file name specified becomes the new primary file. The system releases all other local files, including the previous primary file.</p>
NEW,lfn/ND	<p>Creates a new primary file named lfn. The system converts the previous primary file to a nonprimary temporary file and retains all other local files. If the user omits the ND parameter, the system retains only the new primary file.</p>
NOSORT	<p>The system normally sorts the primary file whenever the current command causes it to read the file. NOSORT prevents the system from automatically sorting the primary file by clearing an internal indicator called the sort flag. The system responds to NOSORT by printing:</p> <p>READY.</p> <p>This command can be used when making additions and/or modifications to the primary file. When followed by the NOSORT command, the additions and/or modifications create a new logical record in the primary file. The NOSORT command remains in effect only until the next numbered line of source code is entered. This causes the sort flag to be turned on again. Refer to File Sorting in section 3 for an example which illustrates the effect of the NOSORT command.</p>
OLD	<p>Allows the user to access a copy of a file that was previously saved in the permanent file system as an indirect access file and to make it the primary file. The system responds:</p> <p>FILE NAME:</p> <p>The user then enters the selected file name. When the file is found, the system replies:</p> <p>READY.</p> <p>A copy of the selected file becomes the new primary file; any subsystem associated with the file is selected automatically (refer to Temporary and Local Files in section 3 for an explanation of subsystem association). All local files are released following entry of this command.</p>
OLD,lfn=pfn/ parameters	<p>Optional form of the OLD command which allows the user to specify the OLD command, the file name, and special parameters in one step. All local files are released following entry of this command unless the optional ND parameter is used. If the user specifies the ND parameter, the previous primary file becomes a nonprimary temporary file. Refer to the command description in section 5 for information concerning this command format.</p>

<u>Command</u>	<u>Description</u>
PACK,lfn ₁ , lfn ₂ ,NR	Compresses several logical records of a local file into one logical record. When the operation is complete, the system responds: READY.
	The PACK command is entered in one of the four following formats, depending upon the parameters selected:
PACK	Compresses the logical records of the primary file into one logical record. When the primary file is packed, it is not sorted automatically by subsequent operations that normally cause the file to be sorted (that is, LIST, RUN, and so on). Automatic sorting resumes when the user enters the next line of source code or the SORT command.
PACK,lfn	Rewinds local file lfn and compresses logical records of the file into one logical record. File lfn is rewound again after the pack.
PACK,lfn ₁ , lfn ₂	Rewinds local file lfn ₁ , compresses logical records of the file into one logical record, and then writes the file to lfn ₂ . If lfn ₂ currently exists, lfn ₁ is written at the current position of lfn ₂ . File lfn ₂ is rewound after the pack; lfn ₁ is not.
PACK,lfn ₁ , lfn ₂ ,NR	Same as the preceding format, except that the no rewind (NR) parameter inhibits file lfn ₁ from being rewound before the pack, unless it is the primary file.
	The primary file is rewound before all operations.
PASSWOR,oldpswd, newpswd	Allows validated users to change their password. The user must supply the correct current password (oldpswd) and an alphanumeric new password (newpswd). The maximum length for passwords is seven characters. The minimum length can be set by the installation to be from zero to seven characters; the released value is four. If the user wants secure entry of passwords, he can enter PASSWOR without parameters. The system responds:
	OLD PASSWORD: ? XXXXXXXXXX
	The user types the old password over the blacked-out characters and then presses CR . The system then requests the new password in the same manner. (These blacked-out characters are provided only on hardcopy terminals.)
	If any part of this command is in error, the system replies:
	ERROR IN PASSWOR ARGUMENTS.

<u>Command</u>	<u>Description</u>
	After the password has been changed, the system replies: READY.
	If the user is not validated to use this command, the system responds: ILLEGAL CONTROL CARD.
PRIMARY,lfn	Makes the temporary file specified by lfn the new primary file while changing the previous primary file into a nonprimary, local file. The system retains all other local files.
RECOVER,nnn	Enables the time-sharing terminal user to resume processing after an accidental disconnect from the system or after a system malfunction that requires the login sequence to be reinitiated. Refer to Recovery in section 6 for a complete description and examples of the use of this command.
RENAME,nlfni= olfn ₁ ,nlfni= olfn ₂ ,..., nlfni=olfn _n	Changes the name of temporary file olfn _i to nlfni. If file nlfni currently exists, it is released. This command cannot be used to change the name of a permanent file. However, a temporary file (including a copy of an indirect access permanent file) can be renamed and then saved in the permanent file system under the new name.
RESEQ	Enables the user to resequence or add line numbers to the current primary file. If the current primary file is a text file, line numbers are added. Statements that reference line numbers are updated when BASIC programs are resequenced in the BASIC subsystem. Refer to Resequencing Line Numbers in section 6 for a complete description and examples of the use of this command.
RETURN,lfn ₁ , lfn ₂ ,...,lfn _n	Releases local files lfn ₁ ,...,lfn _n . (When using tapes or packs refer to RESOURC control statement, NOS Reference Manual, volume 1.)
RETURN,*,lfn ₁ , lfn ₂ ,...,lfn _n	Releases all local files except lfn ₁ ,...,lfn _n . If the user specifies no files, the command releases all local files. This command does not release the CCL work files ZZZZZC0, ZZZZZC1, or ZZZZZC2. (When using tapes or packs refer to RESOURC control statement, NOS Reference Manual, volume 1.)
REWIND,lfn ₁ , lfn ₂ ,...,lfn _n	Positions local files lfn ₁ ,...,lfn _n at beginning-of-information. Refer to File Structure in section 3 for additional information.
REWIND,*	Positions all local files at beginning-of-information. Refer to File Structure in section 3 for additional information.
REWIND,*,lfn ₁ , lfn ₂ ,...,lfn _n	Positions all local files except lfn ₁ ,...,lfn _n at beginning-of-information. Refer to File Structure in section 3 for additional information.
RNH	Optional form of the RUN command which allows the user to run the job without a header being printed. The commands are identical in all other respects.

<u>Command</u>	<u>Description</u>	
RUN,parameters	Compiles and/or initiates execution of the primary file or another local file. The system responds:	

yy/mm/dd. hh.mm.ss.
PROGRAM nnnnnnnn } header

(data or error messages)

SRU ssss UNTS.
RUN COMPLETE.

The header information printed is:

yy/mm/dd.	Current date (year/month/day).
hh.mm.ss.	Current time (24-hour format).
nnnnnnn	File name.

This header is not displayed under the batch subsystem.

One or more of the following parameters can be supplied with the RUN command.

RUN,B=lfn or RUN,C=lfn	Compiles the source program and generates binary file lfn containing the resulting object code. The object code is not executed. File lfn can later be executed under the execute subsystem by entering the RUN command. File lfn can be retained as a permanent file using the SAVE (or REPLACE) command.
RUN,I=lfn	Compiles and/or initiates execution of local file lfn (lfn is assumed to be sorted). In the execute subsystem, no compilation occurs.
RUN,MA=nnnnn	Uses nnnnn as the octal field length for the current program execution.†
RUN,MI=nnnnn	Increments the system default field length by nnnnn octal.

† If the user exceeds the field length for which he is validated, the message ILLEGAL USER ACCESS is returned to the terminal.

<u>Command</u>	<u>Description</u>
RUN,T,q ₁ ,q ₂ , ...,q _n	This command applies only when running a previously compiled (object code) FORTRAN program in the execute subsystem. It allows the user to rename local files specified in the PROGRAM statement without recompiling the program. Parameters q ₁ through q _n specify new local file names to be used in place of those that currently exist in the PROGRAM statement (the parameters are order-dependent and must correspond to those they replace).
The following example illustrates the use of the RUN,T command. Assume a FORTRAN program contains the following PROGRAM statement.	
<pre>PROGRAM TEST (INPUT,OUTPUT,AAA,BBB,TAPE1=AAA, TAPE2=BBB)</pre>	
Normally, to execute a program containing this statement, the user must define local file AAA. Local file BBB results from the execution of the program. However, the user could change the names of local files AAA and BBB without recompiling the program (if the object program exists) by using the RUN,T command under the execute subsystem. For example:	
<pre>RUN,T,INPUT,OUTPUT,CCC,DDD</pre>	
S key	Terminates a job that is currently transmitting output to the terminal. This key is effective only for ASCII code terminals. The system responds:
<pre>*TERMINATED*</pre>	
If pressing this key accidentally disconnects the terminal from the system, dial the computer and enter the RECOVER command in response to the request RECOVER/SYSTEM or RECOVER/CHARGE during the login sequence.	
To terminate a job currently transmitting output to a correspondence code terminal, the user presses the ATTN key. The system responds:	
<pre>*INTERRUPTED*</pre>	
The user must then enter the STOP command, which terminates the job and prints the message:	
<pre>*TERMINATED*</pre>	
Refer to section 2 for complete information concerning the use of special function keys for both ASCII and correspondence code terminals.	

<u>Command</u>	<u>Description</u>
SETASL,nnnnn	Sets SRU account block limit; allows the user to specify an SRU limit (nnnnn decimal units†) for all operations prior to logout or entry of another CHARGE command. The maximum depends on the user's validation limits and the validation limit associated with his charge number (refer to LIMITS Command at the end of this section). If nnnnn is larger than the maximum value, the system uses the maximum value allowed for the user. The user can enter: SETASL,* to set the SRU limit at its maximum.
SETJSL,nnnnn	Sets SRU job step limit; allows the user to specify an SRU limit (nnnnn decimal units†) for each subsequent individual operation (job step). The maximum value of nnnnn is the same as for SETASL, except nnnnn cannot exceed the current value of the account block SRU limit. If nnnnn is larger than the maximum value, the system uses the maximum value allowed for the user. The user can enter: SETJSL,* to set the SRU job step limit at its maximum. If a job exceeds its SRU limit, the system issues the SRU limit message. To continue job execution, the user must enter the S command (refer to Job Suspension in section 6).
SETTL,nnnnn	Sets central processor time limit; allows the user to specify a time limit (nnnnn decimal seconds†) for each subsequent job step requiring the central processor. The maximum time limit depends upon the user's validation limits (refer to LIMITS Command at the end of this section). If nnnnn is larger than the maximum value, the system uses the maximum value allowed for the user. The user can enter: SETTL,* to set the time limit at its maximum. If a job step exceeds its time limit, the system issues the time limit message. To continue job execution, the user must enter the T command (refer to Job Suspension in section 6).
SORT	This command is useful when debugging a program that is cycling in a loop. Enables automatic primary file sorting by setting an internal indicator called a sort flag. The user enters this command to reverse the effects of the NOSORT command. Refer to File Sorting in section 3 for additional information.

† To enter octal units, use nnnnnB, where nnnnn is an octal number.

<u>Command</u>	<u>Description</u>
SORT,lfn,NC=n	Sorts local file lfn by arranging the statements in order using the first n digits of the line number (where $1 \leq n \leq 10$). If the NC=n parameter is omitted, the sort is performed using the first five digits of the line number. The logical records of file lfn are also compressed into one logical record when the sort is performed. When the sort is complete, file lfn is positioned at end-of-information unless it is the primary file, which is automatically rewound. This command forces immediate sorting of the specified file and is commonly used to sort the primary file when it is too long to be sorted automatically by the system. Refer to the description of File Sorting in section 3 for additional information.
STATUS	Same as the ENQUIRE command.
STOP	Terminates any program that is currently in execution or waiting for input from the terminal, unless the program has disabled terminal control (for example, APL). The system responds:
	TERMINATED
SUBMIT	Allows a validated user to create a batch job deck image and submit it to the local batch queue as a deferred batch job. The user cannot interact with a job submitted for processing using this command. Rather, the job is treated in the same manner as a job submitted from the card reader at the local computer site or from a remote batch terminal. Refer to section 7 for a complete description and examples of the use of this command. It is recommended that the user become familiar with all information in section 7 before attempting remote job entry.
SUMMARY	Same as the ENQUIRE command, except default for SUMMARY lists the resources used during the session rather than current job status (refer to description of ENQUIRE command in this section).
TEXT	Selects text mode. This command allows direct entry of information (program statements, data, or text) into the primary file without specifying line numbers. Commands entered in text mode are interpreted as text and are not processed by the system. The system responds:
	ENTER TEXT MODE.
	The input line can consist of a maximum of 150 6-bit characters. The system appends data to the end of the file.
	To terminate text mode from an ASCII code terminal, press the BREAK key or, if provided, the ETX (control C) key. Some terminal interfaces cannot distinguish interrupt from the character NUL, which is not recognized in tape mode; therefore, for these terminals, the ETX character must be used to terminate text mode when in tape mode. If a correspondence code terminal is being used, press the ATTN key. For both terminal types, the carriage must be positioned at the beginning of a new line (empty input line) to be effective. The system responds:
	EXIT TEXT MODE.

CommandDescription

The system automatically writes the text file in one logical record.

The following example uses text mode to create a FORTRAN program, a CCL procedure file, and a data file. The FORTRAN program calculates the grade average, highest grade, and lowest grade from periodic input by classroom instructors. The user creates it in the following manner:

```
new,exam  
READY.  
text  
ENTER TEXT MODE.  
  
program exam(input,output,rundat,tape1=rundat)  
dimension i(50)  
imin=100  
imax=0  
itotal=0  
read(1,10)(i(j),j=1,30)  
10 format(14(i3,1x))  
do 20 j=1,30  
itotal=itotal+i(j)  
if i(j) .lt. imax go to 15  
imax=i(j)  
15 if (imin .gt. i(j))imin=i(j)  
20 continue  
avg=itotal/30  
print 30, avg  
30 format(17h grade average = ,f5.2)  
print 40, imin  
40 format(16h lowest grade = ,i3)  
print 50, imax  
50 format(17h highest grade = ,i3)  
end
```

To exit from text mode, the user presses the BREAK key or, if provided, the ETX key.

EXIT TEXT MODE.

The user makes this program an indirect access permanent file.

save

Command

Description

To correlate this program with input data, the user creates a procedure file under text mode:

```
new,grades

READY.
text
ENTER TEXT MODE.

.proc,average.
get(exam,result)
copycf(result,rundat)
rewind(rundat)
rf1,50000.
ftn(i=exam,l=list)
lgo.
revert. ok.
exit.
replace(list)
dayfile(myday)
replace(myday)
revert,abort. problems.
```

The user exits from text mode as described previously:

```
EXIT TEXT MODE.
```

The user makes GRADES an indirect access permanent file:

```
save
```

The user enters the following data under text mode into a file called RESULT. The system releases all other files:

```
new,result

READY.
text
ENTER TEXT MODE.
```

54	87	90	68	51	87	88	92	73	76	76	81	83	67
78	55	97	68	82	84	94	79	88	77	66	78	98	67
87	99												

<u>Command</u>	<u>Description</u>
----------------	--------------------

The user exits from text mode as described previously:

EXIT TEXT MODE.

The user makes the data file an indirect access permanent file:

save,result

To make use of the procedure file, the user enters:

begin,average,grades

This gives the printout:

```
GRADE AVERAGE = 79.00
LOWEST GRADE = 51
HIGHEST GRADE = 99
```

USER,usernum Allows a validated user in the access subsystem to determine the terminal number(s) to which a specified user (usernum) is currently connected. A list of terminal numbers is returned when multiple terminals are logged in under the same user number. (Refer to LIMITS Command in this section for validation limits.)

XEDIT,parameters Selects XEDIT.

The command format is:

XEDIT,lfn₁,p₁,p₂...,p_n.dcs

If lfn₁ is specified, it must be the first parameter listed; if omitted, the separators which would precede and follow lfn₁ must be included when other parameters are used. The delimited command sequence (dcs), if used, must follow the period. All other parameters are order independent. A brief description of the XEDIT parameters follows.

lfn₁ Name of the local or permanent file to be edited. If lfn₁ is omitted, the primary file is edited.

p_i One or more of the following parameters.

AS Process the file in ASCII mode. Upon exiting XEDIT, the terminal is returned to the mode in effect before the editing session. If the user omits the AS parameter, the mode that the terminal is in before he enters the XEDIT command remains in effect.

<u>Command</u>	<u>Description</u>
B	Process the job as a batch origin job.
C	Create a new file lfn ₁ (creation mode).
FR	Take the first editing commands from the first line of file lfn ₁ . If FR and I=lfn ₂ parameters are both specified, XEDIT executes FR and then I=lfn ₂ .
I=lfn ₂	Take editing commands from file lfn ₂ . If I=0, commands are taken from the dcs field. If I is omitted, commands are taken from file INPUT.
L=lfn ₃	Place XEDIT output on file lfn ₃ . If L=0, no output is generated. The default is L=OUTPUT.
NH	Suppress printing of the XEDIT header.
P	Retrieve and edit permanent file lfn ₁ . Direct access files are attached in write mode. If P is omitted, the file lfn ₁ is assumed to be a local file.
des	Delimited command sequence that is processed before XEDIT takes commands from file INPUT or file lfn ₂ .
Entry of information using TEXT, AUTO, or line-numbered source statements is possible only to the primary file. However, XEDIT allows the user to enter or modify information in all files (including the primary file). This is important when the user wants to reference a direct access file. XEDIT is one method he can use to enter source code into a direct access file.	
XEDIT commands are listed in appendix F. For a description of the XEDIT parameters and commands, refer to the XEDIT Reference Manual.	
-pname,pfile,p ₁ , p ₂ ,...,p _n	Optional form of the BEGIN command. Refer to the description in this section or the BEGIN command for more information.

LIMITS COMMAND

Entry of the LIMITS command provides a list of the user's validation limits. Validation limits are the internal system controls associated with each user number which govern use of certain system resources. All limits are set by the site when the user registers. The listing provided describes both the resources available to the user and the extent to which they may be used. All numeric values listed are decimal unless the postradix B appears, signifying an octal value. The following information is listed.

<u>Field</u>	<u>Description</u>
AB	Answerback identifier (1 to 10 alphanumeric characters). Terminals with answerback capability automatically send this identifier to the system during login. The system uses the information to determine if it is a legal terminal. Each user number can have up to four terminal answerback identifiers associated with it.
MT†	Maximum number of magnetic tapes the user is allowed to have assigned to his job concurrently.
RP†	Maximum number of auxiliary devices the user is allowed to have assigned to his job concurrently.
TL	Maximum amount of central processor time in seconds allowed for the user's job step (refer to SETTL command). The value shown is the actual time limit divided by 10^8 .
CM†	Maximum number of central memory words the user is allowed to request. This is the job's maximum field length, expressed as multiples of 10^8 words.
NF	Maximum number of files the user is allowed to have associated with his job concurrently.
DB	Maximum number of deferred batch jobs the user is allowed to have in the system concurrently (refer to description of SUBMIT command in section 7).
FC	Maximum number of permanent files the user is allowed to have in each catalog. This limit applies for each catalog being accessed (main, public auxiliary, or private auxiliary catalogs).
CS	Maximum number of PRUs available to the user for indirect access permanent files.
FS	Maximum number of PRUs available to the user for any given indirect access permanent file.
PA	Odd or even parity is associated with the terminal (refer to PARITY command in Terminal Control Commands in this section).
RO	Number of rubout characters required for carriage return delay (refer to ROUT command in Terminal Control Commands in this section).
PX	Full- or half-duplex transmission mode is associated with terminal (refer to FULL or HALF commands in Terminal Control Commands in this section).

† For further information about this field, refer to the NOS Reference Manual, volume 1.

<u>Field</u>	<u>Description</u>
TT	Type of terminal:
	TTY ASCII code terminal with standard print.
	COR Correspondence code terminal with standard print.
	CORAPL Correspondence code terminal with APL print.
	MEMAPL Memorex 1240 (ASCII code) terminal with APL print.
	BLKEDT Block transmission (ASCII code) terminal with full display screen editing capability with standard print (available only on certain terminals such as the Hazeltine 2000 terminal).
TC = xxxxx	Initial character set to be used by the terminal. For each account number, xxxxx is one of the following:
	STANDARD 64-character set.
	ASCII ASCII 128-character set or full correspondence code character set.
IS = xxxxx	Initial subsystem for the terminal. For each account number, xxxxx is one of the following:
	BASIC BASIC subsystem.
	BATCH Batch subsystem.
	EXECUTE Execute subsystem.
	FORTRAN FORTRAN subsystem.
	FTNTS FTNTS subsystem.
	NULL Null subsystem.
	TRANACT Transaction subsystem (TAF/TS).
MS	Maximum number of mass storage PRUs the user is allowed to additionally allocate to his job.
DF	Maximum number of dayfile message requests the user is allowed to issue to the system and/or job dayfiles.
CC	Maximum number of batch control statements processed for a user (excludes time-sharing processed control statements).
OF	Maximum number of job print and punch files the user is allowed to dispose to the output queues.
CP	Maximum number of cards that can be punched from a user's disposed punch file.

<u>Field</u>	<u>Description</u>
LP	Maximum number of lines that can be printed from a user's disposed print file.
EC	Maximum number of extended core storage (ECS) memory words that the user is allowed to request.
SL	Maximum number of SRUs allowed for the user's job (refer to SETASL and SETJSL commands).
CN	Not currently used by the system, but is provided for future expansion of validation control.
PN	Project number to which the user is assigned (not the project number associated with the charge number entered at login with the CHARGE command).
DS	Maximum number of PRUs available to the user for any given direct access permanent file.
AW	Access word; the octal value listed corresponds to the following access options (bit 0 is option 0, bit 1 is option 1, and so forth).
<u>Option</u>	<u>Signifies</u>
0	User can change his password (refer to PASSWOR command).
1	User can use the access subsystem (ACCESS command).
2	User can create direct access permanent files.
3	User can create indirect access permanent files.
4	User can have system origin status from any job origin if the system console is in debug mode (refer to NOS Reference Manual, volume 1).
5	User can create and access library type files (refer to NOS Reference Manual, volume 1).
6	User can assign nonallocatable devices (magnetic tape units) (refer to NOS Reference Manual, volume 1).
7	User can access system without supplying his assigned charge and project numbers.
8	User can create and/or replace files on auxiliary devices.

<u>Field</u>		<u>Description</u>
	<u>Option</u>	<u>Signifies</u>
	9	User can access special transaction functions (refer to TAF/TS Reference Manual).
	10	User's terminal is designated as a no-timeout terminal. If this bit is not set, the terminal automatically logs off after 10 minutes of inactivity. With this bit set, the terminal remains connected until the user logs out. The TIMEOUT command (Terminal Control Commands in section 4) clears this bit for the session in progress.
	11	User can use the system control point (SCP) facility.
	12	User has special accounting privileges.
	13	User has special privileges for batch jobs.
	14	User can use PROTECT statements.
	15-23	Reserved for the operating system.
	24-28	Reserved for network applications.
	29-59	Reserved for the operating system.

For example, if the access word listed is

AW=000000000000000000000000215

the user is validated for options 0, 2, 3, and 7, since the binary representation for 215 (octal) is 10001101.

Options ...	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AW (binary) ...	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1
AW= ...	0				0				2		1		5		

If any parameters are included with the LIMITS command, the system responds:

ERROR IN LIMITS ARGUMENTS.

A user enters the LIMITS command and receives the following listing:

LIMITS.	76/02/20. 13.44.12.	PAGE	1
ABC 2435	2435 76/01/06. 76/01/06.		
AB = ,			
MT = 3,			
RP = 2,			
TL = UNLIMITED,			
CM = 2037B,			
NF = 40,			
DB = 10,			
FC = SYSTEM,†			
CS = 32768,			
FS = SYSTEM,†			
PA = EVEN ,			
RO = SYSTEM,†			
PX = HALF ,			
TT = TTY ,			
TC = STANDARD ,			
IS = NULL ,			
MS = 12800,			
DF = 464,			
CC = 464,			
OF = 4,			
CP = 2112,			
LP = 31232,			
EC = 0B,			
SL = 8,			
CN = ,			
PN = ,			
DS = 512,			
AW = 000000000000000000001555			
READY.			

† When the field is equal to system, the user should check with the site for the limit since it is an installation set limit.

PERMANENT FILE COMMANDS

5

INTRODUCTION

Permanent file commands are used in the following ways:

- To create permanent files.
- To access permanent files.
- To change the attributes of permanent files.

They are processed in the same manner as other commands (refer to Command Processing in section 4). Permanent files are described in section 3.

PERMANENT FILE COMMAND FORMAT

The following is the general format of a permanent file command. The optional keywords (kw) and corresponding options must follow the specified file name and the slash character; they are order independent.

pfcmd,lfn=pfn/kw₁=option₁,...,kw_n=option_n

pfcmd Permanent file command name.

lfn Local file name used when creating or accessing permanent file pfn.

pfn Name under which the file is cataloged in the permanent file directory. If the pfn option is omitted, lfn is assumed to be the permanent file name.

The following are the keywords and corresponding options which can be included.

<u>kw</u>	<u>option</u>	<u>Description</u>
BR=	br	Backup requirement. With this parameter, the user specifies whether the system should maintain a backup copy of a file. The user enters the parameter on the CHANGE, DEFINE, or SAVE command.
	Y	Maintain a backup copy. Y is the released default.
	MD	Maintain a backup copy unless the file resides on MSF.
	N	Do not maintain a backup copy.
CE		Clear error flag. The use of the CE parameter in the CHANGE command clears the error flag which had previously been set. (Refer to the NOS Reference Manual, volume 1, for more information on file error status.)
CT=	n	Permanent file category. The file category determines which users can access a file. There are three categories of permanent files.

<u>kw</u>	<u>option</u>	<u>Description</u>
	P, PR, or PRIVATE	Private file. Files available for access only by the originating user, by those explicitly granted permission (refer to PERMIT command in this section), or by those with automatic read-only permission.
	S OR SPRIV	Semiprivate file. Files available for access by any user knowing the file name, password, and user number and whose permitted mode of access is not NULL (refer to description of permission mode under keyword M). Accesses by alternate users are recorded for the originator of the file. This includes the user number of the alternate user, the number of accesses made, and the date and time of the last access (refer to CATLIST Command in this section).
	PU or PUBLIC	Public file. Files available for access by all users knowing the file name, password, and user number. Only the total number of accesses are recorded for the originator of the file.
M=	m	File or user permission mode.
	W or WRITE	Write mode. Allows the user to write, modify, append, read, execute, or purge the file (modify permission applies only to direct access files).
	M or MODIFY	Modify mode. Allows user to modify information within a direct access file and/or append information at the end of the file. The user can also read or execute the file. This mode applies only to direct access files.
	A or APPEND	Append mode. Allows user to append information at the end of the file.
	R or READ	Read mode. Allows user to read and/or execute the file.
	RM or READMD	Read modify mode. Allows user to read a direct access file with the implication that another user may currently be accessing the file in modify mode. The file can also be executed in this mode. This does not apply to indirect access files.
	RA or READAP	Read append mode. Allows user to read a direct access file with the implication that another user may currently be accessing the file in append mode. The file can also be executed in this mode. This mode does not apply to indirect access files.
	E or EXECUTE	Execute mode. Allows the user to execute the file.
	N or NULL	Null mode. Removes previously granted permission.

<u>kw</u>	<u>option</u>	<u>Description</u>
NA		<p>No abort. The NA parameter specifies that if a requested resource is not available, the command is not aborted. NA can be used as follows:</p> <ul style="list-style-type: none"> ● It can be specified on the ATTACH command to indicate that the user wants to wait for the direct access file to become available. If the file is currently being accessed in a mode that conflicts with that specified in the command (interlocked), the user's job is suspended and subsequent terminal entry is not accepted. Entering the STOP command terminates the request. ● It can be used to prevent a permanent file command from aborting if a specified file does not exist or if the user does not have permission to access the file. For the time-sharing user, this is useful for procedure files which perform their own error processing. ● It can be used in conjunction with the PN=packnam and R=r parameters on any permanent file request (including the ATTACH command) for a file that resides on an auxiliary device. If the device is not currently available, the NA parameter directs the system to make the desired auxiliary device available. The system does not make the device available unless the user is validated to have at least one auxiliary device assigned to his job. <p>Use of the NA parameter prevents the job step from aborting when errors occur. Since the system does not indicate when errors occur, the user should enter the DAYFILE or ENQUIRE,F command to determine if a command with the NA parameter has been processed without error.</p>
ND		<p>No drop. The use of the ND parameter in an OLD or LIB command prevents local files from being released and converts the previous primary file to a nonprimary temporary file.</p>
PN= packnam		<p>One- through seven-character pack name used in conjunction with the R=r parameter to identify an auxiliary device to be accessed in the permanent file request. This parameter is specified only when the file to be accessed resides on an auxiliary device. If the device is currently not available and the NA parameter is not specified in the request, the system responds:</p> <p style="text-align: center;">DEVICE UNAVAILABLE.</p>

<u>kw</u>	<u>option</u>	<u>Description</u>
		An auxiliary device is not necessarily a disk pack that can be physically removed as the option name implies. Rather, an auxiliary device can be any mass storage device supported by the system and defined as such by the service vendor. Auxiliary devices are used to supplement the normal mass storage permanent file devices associated with the system (family devices). If two or more auxiliary devices are to be in use concurrently, the user must enter the batch subsystem and issue the RESOURC control statement (refer to the NOS Reference Manual, volume 1, for information concerning the RESOURC statement). The user must be validated to create or replace files on an auxiliary device.
PR=	pr	Preferred residence. The user can indicate that he would like a direct access file to reside on MSF or that he has no preference as to file residence.
	M	The user prefers that the file reside on MSF. However, the file may not reside on MSF even though the user prefers MSF.
	N	The user has no preference as to file residence. N is the released default.
PW=	passwd	One- through seven-character password for a file. If specified by file originator, this password must then be given whenever alternate users access the file. For secure entry of the password, user should specify PW without entering the password in the command. The system responds with:
		PASSWORD: ? XXXXXXXXXX
		The user types the password over this blacked-out area.
R=	r	Type of device on which the permanent file resides or is to reside. Refer to appendix D for a list of device codes. When a direct access file is moved from MSF to disk, it might not be put on the device type originally specified by the R=r parameter on the DEFINE command.
		The R=r parameter can be used in two ways.
		<ul style="list-style-type: none"> It can be specified on the DEFINE command to indicate the type of family device on which the direct access file is to reside. The user should consult the service vendor to determine which devices are available. Appendix D contains a list of the block sizes allocated for each device type. If the file already exists on another device, R is ignored. If an illegal device is specified, the system responds:
		DIRECT ACCESS DEVICE ERROR.

<u>kw</u>	<u>option</u>	<u>Description</u>
		If the number of units specified does not agree with the actual number for device types DI, DJ, DK, DL, DM, or DQ, the system issues the message: ILLEGAL DEVICE REQUEST.
		• It can be used in conjunction with the PN=packnam and NA parameters on any permanent file command (including DEFINE) to identify an auxiliary device on which the file resides or is to reside. If the PN=packnam and NA parameters are specified but the R=r parameter is not, the system default device type is assumed. If the device type of the auxiliary device packnam conflicts with the device specified by the R=r parameter, the system issues the message: ILLEGAL DEVICE REQUEST.
RT		For example, if auxiliary device PN has device type DI and the device type specified by the R=r parameter is DK, a conflict exists and the error message is issued. Real time. Allows the user to enter commands while the system is moving a direct access file from MSF to disk. The RT parameter can be specified only on the ATTACH command. To determine if a direct access file resides on MSF, the user can enter CATLIST,LO=F,FN=pfn. For more information, refer to the ATTACH command.
S=	space	Amount (in decimal PRUs) of space wanted for a new direct access permanent file (refer to DEFINE command). There are 640, decimal, 6-bit characters in a PRU. The system then creates the file on a device with the specified amount of space available. Unused space is not guaranteed to be available if the user attempts to expand the file at a later date. If no device currently has the specified amount of space available, the system responds: PRUS REQUESTED UNAVAILABLE. If the S=space parameter is used in conjunction with the R=r parameter, only devices of the type specified are checked for adequate space.
SS=	subsys	Subsystem to be associated with a file. This parameter can be specified on only the SAVE and CHANGE commands. If the user specifies just SS, the current subsystem is associated with the file. If the SS=subsyst parameter is omitted, the null subsystem is associated with the file, unless the file is the primary file, in which case, the current subsystem is associated with the file.

<u>kw</u>	<u>option</u>	<u>subsys</u>	<u>Description</u>
			<u>Description</u>
		NULL	Null subsystem
		BASIC	BASIC subsystem
		FTNTS	FTNTS subsystem
		FORTAN	FORTRAN subsystem
		EXECUTE	Execute subsystem
		BATCH	Batch subsystem

The user can enter the first three characters of the subsystem name as an abbreviation for the subsystem.

UN= usernum	Alternate user number. This parameter is necessary only if the permanent file involved resides in another user's catalog. To access a file in another user's catalog, the requesting user must have explicit permission (refer to PERMIT command in this section), must have automatic read-only permission (user number contains asterisks for all characters that do not match in the alternate user number), or the file must be a semiprivate or public file.
-------------	---

PERMANENT FILE PROCESSING COMMANDS

The following permanent file commands are available to the user. The parameters specified have been described in Permanent File Command Format in the preceding pages.

<u>Command</u>	<u>Description</u>
APPEND,pfn,lfn ₁ ,...,lfn _n /UN=usernum,PW=password,PN=packnam,R=r,NA	Appends temporary files lfn ₁ ,...,lfn _n at the end of the specified indirect access permanent file pfn. The logical structure of the files is retained; that is, EORs and EOFs are appended as well as data. Each temporary file is appended in the order specified in the command. This command can only be used for indirect access permanent files.
ATTACH,lfn ₁ =pfn ₁ ,lfn ₂ =pfn ₂ ,...,lfn _n =pfn _n /M=m,UN=usernum,PW=password,PN=packnam,R=r,NA,RT	Establishes a link to permanent file pfn _i for direct access usage. If pfn _i is omitted, the system assumes lfn _i =pfn _i .
	A temporary file is not created since user access is directly to the permanent file. The lfn _i option is used to reference the attached file by a name other than its permanent file name (pfn _i). If lfn _i specifies the name of a temporary file, the contents of that file are lost when the permanent file is attached. If lfn _i specifies the name of a direct access permanent file that is already attached, that file is released, and the file being attached is referred to by the name lfn _i . In addition, lfn _i cannot specify the primary file name. A direct access file cannot become the primary file.

<u>Command</u>	<u>Description</u>
	If the M=m parameter (permission mode) is omitted, the system assumes read permission for the attached file. The M=m parameter must be specified by all users, including the file originator, if the file is to be modified or if new information is to be added.
	The RT parameter is specified when the user wants to attach a direct access file that might be on MSF, but does not want to wait until the file is copied to disk before entering other commands. If the user specifies the RT parameter and if the file is on MSF, the file is not attached but is copied to disk. If the file is already on disk, it is attached. The message
	pfn STAGE INITIATED.
	is placed in the dayfile if the file is on MSF and the user specifies the RT parameter. This means the system has started the process of copying the file to disk. If the user is in the batch subsystem, pfn STAGE INITIATED is also returned to the terminal. In other subsystems, the user can determine the status of the file by entering any of the following:
DAYFILE command	This shows what the system did in response to the ATTACH command.
ENQUIRE,F	The system lists the names of the user's local files. If the file is attached, its name will be in the list.
LENGTH,lfn or ENQUIRE, FN=lfn	The system lists local file information on file lfn if it is attached.
ATTACH command without RT	When the system responds READY, the file is attached.
If the file was copied to disk in response to an ATTACH command with the RT parameter, the user must enter another ATTACH command to actually attach the file.	
If the RT parameter is not specified when the user enters the ATTACH command to attach a direct access file that is on MSF, the user must wait while the file is copied to disk and then attached to his job. While waiting, the user can enter the ENQUIRE command to obtain information relating to the file. One of the following job status messages can appear in response to the ENQUIRE command.	

<u>Command</u>	<u>Message</u>	<u>Description</u>
	PF STAGING INITIATED.	Processing of the ATTACH command is delayed until the file is copied from MSF to disk.
	PF UTILITY ACTIVE.	The system is performing a permanent file maintenance procedure. The user should wait until the procedure is finished.
	WAITING FOR MSSEEXEC.	The system operator must activate the MSSEEXEC subsystem before the system can copy the file from MSF to disk.

The user can terminate the ATTACH command by entering STOP. If the user enters STOP, the system copies the file to disk but does not attach it to the user's job. However, if the system is waiting for the operator to activate MSSEEXEC and the user enters STOP, then the system does no copying.

If the NA parameter is not specified and another user is currently accessing the file in a mode that conflicts with that specified in the command (M=m parameter), the system responds

pfn BUSY,AT addr.

When a direct access permanent file is attached in the write mode, it should be released after use so that it can be attached by other users. When a direct access permanent file is attached in the write mode, the last modification date is updated even if the file is not altered.

A direct access permanent file can be released with any of the following commands (refer to section 4 for a description of these commands).

- OLD, NEW, or LIB (all local files also released).
- GOODBYE, BYE, LOGOUT, HELLO, or LOGIN (all local files also released).
- CLEAR or CLEAR,* (all local files also released).
- RETURN, lfn (local file specified released) or RETURN,* (all local files also released).

A direct access file is also released if the lfn option on subsequent GET or ATTACH commands specifies the name of the attached file.

<u>Command</u>	<u>Description</u>
CATLIST	Refer to CATLIST Command later in this section.
CHANGE,nfn=ofn/PW=password, CT=n,M=m,BR=br,PR=pr SS=subsyst,PN=packnam, R=r,NA,CE	Allows the originator of a direct or indirect access permanent file to alter any of several parameters without having to attach and redefine the file or retrieve and save it. This command is valid only for the originator of the file (UN=username parameter is illegal).
	The nfn parameter specifies the new permanent file name to be assigned; ofn is the current file name. If no name change is desired, only ofn is specified. To clear the password currently assigned to a file, set the new password equal to zero (PW=0). The PW=password, CT=n, M=m, BR=br, PR=pr, and SS=subsyst parameters should be specified only when the user wants to change the option associated with that parameter. The PN=packnam and R=r parameters cannot be used to specify a new auxiliary device. They are used only to specify the auxiliary device on which ofn resides. The use of the CE parameter clears the file error status and allows the change in parameters to be made.
	If the current permanent file name, ofn, is not in the user's catalog, the system responds:
	ofn NOT FOUND.
	If the new permanent file name, nfn, already exists in the user's catalog, the system responds:
	nfn ALREADY PERMANENT.
DEFINE,lfn ₁ =pfni ₁ ,lfn ₂ = pfni ₂ ,...,lfn _n =pfni _n / PW=password,CT=n,M=m,BR=br, PR=pr,PN=packnam,R=r, S=space,NA	Allows a validated user to create a direct access permanent file (pfni _i) and attach it in write mode. If pfni _i is omitted, the system assumes lfn _i =pfni _i .
	If pfni _i is specified, the lfn _i option can be used in two ways.
	<ul style="list-style-type: none"> ● To reference the direct access file by a name other than its permanent file name (pfni_i). In this case, the direct access file created contains no information initially. Data is placed on the file in succeeding write operations. ● To define a temporary file as a direct access permanent file, the temporary file specified must reside on a mass storage device that also supports the residence of direct access permanent files. If the file specified by lfn_i does not reside on such a device, the system responds:
	DIRECT ACCESS DEVICE ERROR.

<u>Command</u>	<u>Description</u>
	NOTE
	Since time-sharing users are generally unaware of the type of device that contains their temporary files, this use of the lfn _i parameter is not recommended.
	If the optional parameters are omitted, the system assumes the following:
PW	No password.
CT	Private file.
M	Write mode.
BR	Maintain a backup copy.
PR	No preferred residence.
PN	The file will be placed on the family device or on the auxiliary device specified by the last PACKNAM command.
R }	File placed on device with most space available.
S	
NA	Not applicable unless PN specified.

If the user releases the file and then wants to access it at some time in the future, the ATTACH command must be entered (refer to the ATTACH command in this section).

For information regarding creation of indirect access permanent files, refer to the SAVE command in this section.

GET,lfn₁=pfni₁,lfn₂=
pfni₂,...,lfn_n=pfni_n/
UN=usernum,PW=password,
PN=packnam,R=r,NA

Retrieves a copy of the specified indirect access permanent file (pfni_i) for use as a temporary file (lfn_i). This command can only be used for indirect access permanent files. If pfni_i is omitted, the system assumes lfn_i=pfni_i.

To reference a temporary file by a name other than its permanent file name (pfni_i), the user specifies the lfn_i option. However, if lfn_i is the name of a current local file, the current file is released and a copy of file pfni_i becomes the temporary file named lfn_i. The current primary file and subsystem remain the same unless lfn_i is the name of the current primary file. In this case, the current primary file is released and a copy of pfni_i becomes the primary file named lfn_i. Furthermore, if pfni_i has a subsystem flag associated with it

<u>Command</u>	<u>Description</u>
	(refer to SAVE command in this section), that subsystem is selected automatically. For example, if file A is the current primary file and the user enters:
GET,A=B	the copy of permanent file B becomes the new primary file. If a subsystem other than the one currently active is associated with file B, it is selected.
	In order for the user to access a file in another user's catalog (UN=usernum parameter specified), the permission mode must allow the user to read the file.
LIB,lfn=pfn/PW=password, PN=packnam,R=r,NA,ND	Retrieves a copy of the specified indirect access permanent file from the catalog of special user number LIBRARY and makes it the primary file.
	The permission mode is that which has been granted for private files or specified in the catalog for semiprivate and public files. All local files are released unless the ND parameter is specified.
OLD,lfn=pfn/UN=usernum, PW=password,PN=packnam,R=r, NA,ND	The ATTACH, GET, or OLD command can also be used to access permanent files in the catalog of user number LIBRARY. For additional information, refer to LIB Command at the end of this section.
PACKNAM,PN=packnam or PACKNAM,packnam	Retrieves a copy of the specified indirect access permanent file for use as the primary file. If a specific subsystem is associated with the file, it is selected automatically. Refer to the description of the SAVE command in this section for additional information on subsystem association. All local files are released unless the ND parameter is specified.
	Enables validated users to direct subsequent permanent file requests to the auxiliary device specified by the PN option. This command allows the user to omit the PN=packnam parameter in requests for files that reside on that auxiliary device. However, if permanent files on another auxiliary device are to be accessed, the PN=packnam parameter can be included in the request or a new PACKNAM command can be entered.
	The user cannot access permanent files residing on the normal system devices (family devices) while the PACKNAM command is in effect. To access these files again, he must enter one of the following:
	PACKNAM,PN=0 or PACKNAM

<u>Command</u>	<u>Description</u>
	An auxiliary device is not necessarily a disk pack that can be physically removed as the command name implies. Rather, an auxiliary device can be any mass storage device defined as such by the site. Auxiliary devices are used to supplement the normal mass storage permanent file devices associated with the system (family devices).
	Examples:
	OLD,FILEA ← Request to normal family device for file FILEA.
	PACKNAM, ← Permanent file requests that follow are made to auxiliary device ABC, unless PN option is specified in the request or a new PACKNAM command is entered.
	GET,DATA ← Request to auxiliary device ABC for file DATA.
	PACKNAM, ← Permanent file requests that follow are made to auxiliary device XYZ.
	GET,TAPE1 ← Request to auxiliary device XYZ for files TAPE1 and TAPE2.
	GET,TAPE2
	PACKNAM, ← Clear PACKNAM command. Permanent file requests that follow are made to the normal family devices.
PERMIT,pfn,usernum ₁ =m ₁ , usernum ₂ =m ₂ ,..., usernum _n =m _n /PN=packnam, R=r,NA	Grants user usernum _i permission to access private or semiprivate file pfn. If the permission mode, m _j , is omitted, the system assumes read permission.
PURGE,pfn ₁ ,pfn ₂ ,..., pfn _n /UN=usernum,PW=password, PN=packnam,R=r,NA	Removes the specified permanent files from permanent file storage. If the permanent file (pfn _i) does not exist, the system responds: pfn NOT FOUND. Write permission is required to purge a permanent file in an alternate user's catalog (UN=usernum parameter specified). The PW=password parameter must also be specified if the file has a password. Direct access files are not actually purged until the last user accessing the file has released it. However, subsequent attempts to access the file are rejected.

<u>Command</u>	<u>Description</u>
REPLACE,lfn ₁ =pfni, lfn ₂ =pfni ₂ ,..., lfn _n =pfni _n /UN=usernum, PW=password,PN=packnam, R=r,NA	Allows validated users to replace the contents of an indirect access permanent file (pfni) with the contents of a temporary file (lfni). If the file name for pfni is not specified or does not exist, a new permanent file is created. The REPLACE command does not affect the subsystem associated with the permanent file (if any).
SAVE,lfn ₁ =pfni, lfn ₂ =pfni ₂ ,..., lfn _n =pfni _n /PW=password, CT=n,M=m,SS=subsyst,BR=br, PN=packnam,R=r,NA	Creates one or more indirect access permanent files. This command allows a validated user to retain a copy of the specified temporary file (lfni) in the permanent file system. If the optional parameters are omitted, the system assumes the following values.
PW	No password.
CT	Private file.
M	Write mode.
SS	Current subsystem when lfn _i is the primary file. Null subsystem when lfn _i is not a primary file.
BR	Maintain a backup copy.
PN	The file will be placed on the family device or on the auxiliary device specified by the last PACKNAM command.
R NA	Not applicable unless PN is specified.

Unless the null subsystem is active when the primary file is saved, an internal indicator called the subsystem flag is set to indicate which subsystem is being used. That subsystem becomes associated with the permanent file (pfni) and is reselected automatically each time the file is retrieved in subsequent requests using the OLD command. To associate the primary file with a subsystem different from the current subsystem, the user can enter the SS=subsyst parameter. To save the primary file without a subsystem association, the user can enter the SS=NULL parameter. The user can also use the SS parameter to associate a subsystem with a temporary file that is not the primary file.

CATLIST COMMAND

Users can obtain specific information about their permanent files with the CATLIST command. In addition, this command can be entered to obtain specific information about permanent files the user can access in the catalogs of alternate users.

COMMAND FORMAT

The following is the format of the CATLIST command; the optional keywords (and corresponding parameters) included must follow the comma and are order independent.

CATLIST,LO=option,FN=pfn,UN=usernum,PN=packnam,R=r,DN=dn,NA,L=lfn

<u>Keyword</u>	<u>Option</u>	<u>Description</u>
LO=	F	Selects a listing of pertinent information about each file in the user's catalog (refer to example 1 under Examples in this section). If an alternate user number is specified (UN parameter), it selects a listing of the names of all files the user can access in the alternate user's catalog (refer to example 2 under Examples in this section). The passwords for files in an alternate user's catalog are not included in the listing; they must be obtained directly from that user.
	FP	Selects a listing of permission information recorded for each alternate user that accessed a specified file in the user's catalog (refer to Alternate User Access in this section). This option requires that a file name be specified (FN=pfn parameter). If an alternate user number is also specified (UN=usernum parameter), only the permission information recorded for that user of the specified file is listed.
		The user numbers listed include <ul style="list-style-type: none">● Those that have explicit permission to access the file (private or semiprivate file).● Those that have implicit permission to access the file (semiprivate files only). Permission information for alternate users that have accessed public files is not recorded.
0		Selects a short list that gives only the names of the files in the user's catalog (refer to example 3 under Examples in this section). Files are listed alphabetically and are grouped as indirect access files and direct access files. An asterisk preceding a file name indicates that an error flag is set in the catalog entry for the file. The cause of the error may be any of the following: <ul style="list-style-type: none">● Alteration of EOI by the recovery process.● Error in verifying BOI/EOI position.● Error in data and/or permit entries. If the user attempts to attach such a file without specifying the NA parameter, the system issues an error message stating the reason for the error status.

<u>Keyword</u>	<u>Option</u>	<u>Description</u>
		If the user specifies an alternate user number also (UN=usernum), the user obtains only the names of the files he can access in the alternate user's catalog.
		If the user specifies a file name (FN=pfn), the user receives a message that tells him if the system found the file.
		If the user does not specify a value for LO, the system assumes LO=0.
P		Selects a short list that gives only the user numbers of alternate users who accessed the specified private or semiprivate file (refer to example 4 under Examples in this section). This parameter requires a value for parameter FN (FN=pfn).
		If the user specifies UN=usernum, the system tells the user if it has granted permission to the named file to that user number:
		usernum FOUND. or usernum NOT FOUND, AT nnnnnn.
		nnnnnn is an internal address the user may ignore.
FN=	pfn	The user obtains catalog information only for permanent file pfn. The user must specify this parameter when listing permit information (LO=P, LO=FP).
		If a user has several files with similar names, he can list information about all of these files using only one FN=pfn parameter: when specifying pfn, the user inserts an asterisk in place of any letter that is not the same in all of his file names.
		Examples:
		FN=***OPL Lists information about all files having a six-character file name ending with the letters OPL.
		FN=M***** Lists information about all files having a file name beginning with the letter M.
		Use of the asterisk is not allowed when listing permit information (LO=FP, LO=P).
UN=	usernum	This parameter has two purposes. <ul style="list-style-type: none"> ● For LO=F and LO=0. Specifies the alternate catalog for which the user wants catalog information (refer to example 2 under Examples in this section).

<u>Keyword</u>	<u>Option</u>	<u>Description</u>
		<ul style="list-style-type: none"> ● For LO=FP and LO=P. Specifies the alternate user for which the user wants recorded permission information.
PN=	packnam	<p>Identifies an auxiliary device that contains catalog information for all users with files on that device. Unless the PACKNAM command is in effect, the user must specify the PN parameter to obtain the following information from his catalog on an auxiliary device.</p> <ul style="list-style-type: none"> ● Pertinent information about each file that resides on the device (LO=F). ● Only the name of each file (LO=0). ● Permission information for each alternate user that accessed a specific file (LO=FP). ● Only the user number of each alternate user that accessed a specific file (LO=P). <p>If two or more auxiliary devices are in use concurrently, the user must enter the batch subsystem and issue the RESOURC control statement (refer to the NOS Reference Manual, volume 1). The PN=packnam parameter can also be specified by alternate users to obtain a list of files and pertinent information about each file they can access on an auxiliary device.</p>
R=	r	<p>This parameter is used in conjunction with the PN=packnam and NA parameters to specify the type of auxiliary device that contains the desired catalog information. If the PN=packnam and NA parameters are specified but the R=r parameter is not, the system default device type is assumed. If the device type of auxiliary device packnam conflicts with the device specified by the R=r parameter, the system issues the message:</p> <p style="text-align: center;">ILLEGAL DEVICE REQUEST.</p> <p>For example, if auxiliary device packnam is device type DI and the device type specified by the R=r parameter is DK, a conflict exists and the error message is issued.</p> <p>If the number of units specified does not agree with the actual number for device types DI, DJ, DK, DL, DM, and DQ, the system issues the message:</p> <p style="text-align: center;">ILLEGAL DEVICE REQUEST.</p> <p>Refer to appendix D for a list of device codes.</p>
DN=	dn	The device number (1 to 77) specified is searched for files that are specified by the CATLIST command options.

<u>Keyword</u>	<u>Option</u>	<u>Description</u>
NA		<p>Directs the system to make a device available if it is currently not available. NA is used in conjunction with the PN=packnam parameter in requests for catalog information on auxiliary devices. If the NA parameter is omitted and the device is not currently available, the system responds:</p> <p style="text-align: center;">DEVICE UNAVAILABLE.</p> <p>The system does not make the device available unless the user is validated to have at least one auxiliary device assigned to his job (refer to LIMITS Command in section 4).</p>
L=	lfn	<p>Specifies the name of a local file to which the CATLIST information is written. If this parameter is omitted, the system assumes L=OUTPUT and the data is returned to the terminal. If an alternate file is specified, the CATLIST information is written at the current position in the file. For example, if lfn is positioned at BOI, the contents of that file are replaced with the CATLIST information. If lfn is positioned at EOI, the CATLIST information is appended to the file as a new logical record.</p>

INFORMATION FORMAT

The format of the first line of information returned for all CATLIST commands is:

CATALOG OF usernum	FM/family	yy/mm/dd.	hh.mm.ss.
usernum	The user number entered during login.		
family	The name of the set of permanent file devices on which the user's permanent files are stored. If the user specifies the PN parameter in the CATLIST command, the system prints PN=packnam instead of FM/family, where packnam is the name of the device the user specified with the PN parameter.		
yy/mm/dd	The date in year/month/day.		
hh.mm.ss	The time in hours, minutes, and seconds.		

The general format of the information returned for a full CATLIST command (LO=F option) is:

FILE NAME	ACCESS	FILE-TYPE	LENGTH	DN	CREATION	ACCESS	DATA MOD
PASSWORD	MD/CNT	INDEX	PERM.	SUBSYS	DATE/TIME	DATE/TIME	DATE/TIME
PR	BR	RS					

list of files and information

xxxxx INDIRECT ACCESS FILE(S), TOTAL PRUS = nnnnn.

xxxxx DIRECT ACCESS FILE(S), TOTAL PRUS = nnnnn.

The following are descriptions of the various fields.

FILE NAME	Permanent file name.
ACCESS MD/CNT	The access mode field contains the permanent file type, either direct access (DIR) or indirect access (IND). The access count field contains the number of times users have accessed the file.
FILE-TYPE	Method of access or category. This field can be either private (PRIVATE), semiprivate (SEMI-PR), or public (PUBLIC).
LENGTH	Length of the file in decimal PRUs.
DN	Device number of the mass storage device on which a direct access file is stored. If the file resides on the master device, this field is replaced by an *.
CREATION DATE/TIME	Time and date of file creation. The format is: yy/mm/dd. hh.mm.ss.
ACCESS DATE/TIME	Time and date of the last access to the file.
DATA MOD DATE/TIME	Time and date of the last modification to the file.
PASSWORD	Password associated with the file (field not present if catalog of alternate user name).
INDEX	User index (reserved for system utilities).
PERM.	Permission mode. Entry can be WRITE, MODIFY, APPEND, READ, READMD, READAP, or EXECUTE.
SUBSYS	Subsystem under which the file was saved. Possible entries include FORT, FTNTS, BASIC, EXEC., or BATCH. If this field contains no entry, a subsystem is not associated with the file.

PR	Preferred residence. This field can be M for MSF preference or N for no preference (refer to the PR=pr parameter description in this section).
BR	Backup requirement. This field can be Y for backup copy maintained, MD for backup copy maintained if file is not on MSF, or N for no backup copy (refer to the BR=br parameter description in this section).
RS	Actual residence. This field can be A for MSF, D for disk, or B for both MSF and disk. If the user attaches an MSF file in write, modify, or append mode, the actual residence will show D. If the user attaches an MSF file in a mode other than write, modify, or append, the actual residence will show B.
xxxxx	Total of each file type.
nnnnn	Total of PRUs allocated to each file type.

EXAMPLES

The following examples illustrate some of the options available to the user when entering the CATLIST command.

Example 1. Listing of pertinent information about each file in the catalog of BJK2201. The command is entered by user BJK2201 in the form: CATLIST,LO=F.

CATALOG OF BJK2201

FM/NOSCLSH 79/09/28. 15.28.15.

FILE NAME	ACCESS	FILE-TYPE	LENGTH	DN	CREATION	ACCESS	DATA	MOD
PASSWORD	MD/CNT	INDEX	PERM.	SUBSYS	DATE/TIME	DATE/TIME	DATE/TIME	
PR	BR	RS						

1 XX	IND.	PRIVATE	66	WRITE	1	78/10/10.	79/09/27.	78/10/10.
						10.04.59.	12.54.50.	11.24.03.
N Y D								
2 FIND	IND.	PRIVATE	11	WRITE BASIC	1	79/02/14.	79/09/27.	79/02/14.
1234						14.02.49.	12.54.51.	14.02.49.
N Y D								
3 ID	IND.	PUBLIC	9	EXEC EXEC.	3	79/03/06.	79/09/27.	79/03/28.
NUMBER						15.25.35.	12.54.51.	12.48.53.
N Y D								
4 CAPITAL	IND.	SEMI-PR	6	READ FORT.	1	79/03/06.	79/09/27.	79/03/06.
						15.19.03.	12.54.51.	15.19.03.
N Y D								
5 MODIFY2	IND.	PRIVATE	3	WRITE FTNTS	2	79/02/14.	79/09/27.	79/02/14.
						13.47.58.	12.54.51.	13.47.58.
N Y D								
6 TV	DIR.	PRIVATE	9	WRITE	1	* 79/03/06.	79/09/27.	79/05/03.
						08.25.21.	12.54.51.	09.43.13.
N Y D								
7 PRIME	IND.	SEMI-PR	6	READMD	1	79/03/06.	79/09/27.	79/03/06.
						08.24.16.	12.54.52.	08.24.16.
N Y D								
8 DIRFILE	DIR.	PUBLIC	5	READAP	3	* 79/03/06.	79/09/27.	79/03/06.
LARGE						14.58.45.	12.54.52.	14.58.45.
N Y D								
6 INDIRECT ACCESS FILE(S),					TOTAL PRUS =			9.
2 DIRECT ACCESS FILE(S),					TOTAL PRUS =			4.

Example 2. Listing of pertinent information about each file that JLC2016 can access in the catalog of BJK2201. The command is entered in the form:
 CATLIST,LO=F,UN=BJK2201.

CATALOG OF JLC2016			FM/NOSCLSH 79/09/28. 15.31.16.		
			ALTERNATE CATALOG BJK2201		
	FILE NAME	ACCESS	FILE-TYPE	LENGTH	DN CREATION ACCESS DATA MOD
			MD/CNT	INDEX	PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
	PR BR RS				
1	FIND	IND. PRIVATE	11	WRITE BASIC	79/02/14. 79/09/27. 79/02/14. 14.02.49. 12.54.51. 14.02.49.
		N Y D			
2	ID	IND. PUBLIC	9	EXEC EXEC.	79/03/06. 79/09/27. 79/03/28. 15.25.35. 12.54.51. 12.48.53.
		N Y D			
3	CAPITAL	IND. SEMI-PR	6	READ FORT.	79/03/06. 79/09/27. 79/03/06. 15.19.03. 12.54.51. 15.19.03.
		N Y D			
4	PRIME	IND. SEMI-PR	6	READMD	79/03/06. 79/09/27. 79/03/06. 08.24.16. 12.54.52. 08.24.16.
		N Y D			
5	DIRFILE	DIR. PUBLIC	5	READADP	79/03/06. 79/09/27. 79/03/06. 14.58.45. 12.54.52. 14.58.45.
		N Y D			
4	INDIRECT ACCESS FILE(S),				TOTAL PRUS = 6.
1	DIRECT ACCESS FILE(S),				TOTAL PRUS = 3.

DN indicates device number for direct access files. An * in this column indicates that file resides on master device (refer to appendix C).

Example 3. Listing of current files in the catalog of BJK2201. The command is entered by user BJK2201 in the form: CATLIST,LO=0 (it is not necessary to specify the LO=0 option since it is the default value).

```
CATALOG OF BJK2201          FM/NOSCLSH 79/03/14. 09.10.47.  
INDIRECT ACCESS FILE(S)  
ADD      EXAM      GRADES     ID      MODIFY2    RESEQ    XX  
CAPITAL   FIND      *HEROFTN   LIST      PRIME      T  
DIRECT ACCESS FILE(S)  
DIRFILE   DRFILE    TV  
13 INDIRECT ACCESS FILE(S),   TOTAL PRUS =        14.  
3 DIRECT ACCESS FILE(S),   TOTAL PRUS =        2.
```

An asterisk preceding a file name indicates error flag set.

Example 4. Listing of alternate users that have accessed file PRIME in the catalog of BJK2201. The command is entered by user BJK2201 in the form: CATLIST,LO=P,FN=PRIME.

```
CATALOG OF BJK2201          FM/NOSCLSH 79/03/08. 07.48.55.  
FILE NAME PRIME  
USER NUMBER(S)  
  
CML2011   JLC2016   KXK4277  
3 USER(S)
```

ALTERNATE USER ACCESS

The following examples illustrate the alternatives available to users for accessing files in alternate catalogs.

PRIVATE FILE CATEGORY

Temporary file FIND is retained by BJK2201 as a private indirect access permanent file using the following command.

```
SAVE,FIND/PW=1234
```

The following is the entry for file FIND in BJK2201's catalog. The listing is obtained by entering the CATLIST command in the form: CATLIST,LO=FP,FN=FIND.

```
CATALOG OF BJK2201          FM/NOSCLSH 79/09/28. 15.27.16.  
FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION ACCESS DATA MOD  
PASSWORD MD/CNT INDEX PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME  
PR BR RS  
  
1 FIND      IND. PRIVATE      1    79/02/14. 79/09/27. 79/02/14.  
1234        11           WRITE BASIC 14.02.49. 12.54.51. 14.02.49.  
N Y D  
  
1 INDIRECT ACCESS FILE(S),   TOTAL PRUS =      1.
```

User BJK2201 issues the following command to permit alternate user access.

```
PERMIT,FIND,KXK4277=R,JLC2016=W
```

The following is the permission information recorded in the catalog of BJK2201. The listing is obtained by entering the CATLIST command in the form: CATLIST,LO=FP,FN=FIND.

```
CATALOG OF BJK2201          FM/NOSCLSH 79/03/08. 07.50.41.  
FILE NAME FIND  
USER NUMBER PERM. ACCESSES DATE      TIME  
  
1. KXK4277  READ*      1 79/03/06. 15.31.58.  
2. JLC2016  WRITE*     2 79/03/06. 15.40.02.
```

An asterisk indicates explicit permit flag.

The format of the command required of KXK4277 and JLC2016 to access the file is:

GET,FIND/UN=BJK 2201,PW=1234 or OLD,FIND/UN=BJK 2201,PW=1234

Only the following users can access this private file.

User	Information Necessary for Access	
BJK2201	File name	FIND
KXK4277	File name File password User number	FIND 1234 BJK 2201
JLC2016	File name File password User number	FIND 1234 BJK 2201

SEMIPRIVATE FILE CATEGORY

Temporary file CAPITAL is retained by BJK2201 as a semiprivate indirect access permanent file using the following command.

SAVE,CAPITAL/CT=SPRIV,M=READ

The following is the entry for file CAPITAL in the catalog of BJK2201. The listing is obtained by entering the CATLIST command in the form: CATLIST,LO=F,FN=CAPITAL.

CATALOG OF BJK2201 FM/NOSCLSH 79/09/28. 15.26.31.

FILE NAME	ACCESS	FILE-TYPE	LENGTH	DN	CREATION	ACCESS	DATA	MOD
PASSWORD	MD/CNT	INDEX	PERM.	SUBSYS	DATE/TIME	DATE/TIME	DATE/TIME	DATE/TIME
PR	BR	RS						

1	CAPITAL	IND.	SEMI-PR	6	1	79/03/06.	79/09/27.	79/03/06.
					READ FORT.	15.19.03.	12.54.51.	15.19.03.
N	Y	D						

1 INDIRECT ACCESS FILE(S), TOTAL PRUS = 1.

The following is the permission information recorded in the catalog of BJK2201. The listing is obtained by entering the CATLIST command in the form: CATLIST,LO=FP,FN=CAPITAL.

CATALOG OF BJK2201 FM/NOSCLSH 79/03/08. 07.52.07.

FILE NAME	CAPITAL	USER NUMBER	PERM.	ACCESES	DATE	TIME
-----------	---------	-------------	-------	---------	------	------

1. KXK4277	READ	3	79/03/06.	15.34.51.
2. JLC2016	READ	1	79/03/06.	15.39.19.
3. CML2011	READ	1	79/03/06.	15.43.27.

The listing shows that three users have accessed file CAPITAL the indicated number of times. When an alternate user accesses file CAPITAL the first time, the permission information is recorded in the catalog of BJK2201. Each succeeding access increases the access count (ACCESSES). The absence of an asterisk after the permission mode indicates that this is an accounting permit and was not created by a PERMIT command.

The format of the command required of all users, except BJK2201, to access the file is:

GET,CAPITAL/UN=BJK2201 or OLD,CAPITAL/UN=BJK 2201

Users who can access this file are:

<u>User</u>	<u>Information Necessary for Access</u>	
BJK2201	File name	CAPITAL
All others	File name Password User number	CAPITAL None required BJK 2201

PUBLIC FILE CATEGORY

Temporary file ID is retained by BJK2201 as a public indirect access permanent file using the following command.

SAVE,ID=NUMBER,CT=PUBLIC,M=EXECUTE

The following is the entry for file ID in the catalog of BJK2201. The list is obtained by entering the CATLIST command in the form: CATLIST,LO=F,FN=ID.

CATALOG OF BJK2201 FM/NOSCLSH 79/09/28. 15.25.41.

FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION ACCESS DATA MOD
PASSWORD MD/CNT INDEX PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
PR BR RS

1 ID IND. PUBLIC 3 79/03/06. 79/09/27. 79/03/28.
NUMBER 9 EXEC EXEC. 15.25.35. 12.54.51. 12.48.53.
N Y D

1 INDIRECT ACCESS FILE(S), TOTAL PRUS = 3.

Permission information is not recorded for accesses to a public file; only the total number of accesses is recorded for the originator of the file.

The format of the command required of all users, except BJK2201, to access file ID is:

GET,ID/UN=BJK 2201,PW=NUMBER or OLD,ID/UN=BJK 2201,PW=NUMBER

Users who can access this file are:

<u>User</u>	<u>Information Necessary for Access</u>	
BJK2201	File name	ID
All others	File name Password User number	ID NUMBER BJK 2201

LIB COMMAND

The LIB command allows access to indirect access permanent files in the catalog of special user number LIBRARY. Direct access files can also reside in the catalog of LIBRARY although an alternate method must be used to access these files (described later in this section). The file category is private, semiprivate, or public. The permission mode is that which has been granted for private files or specified in the catalog for semiprivate and public files.

When the LIB command is entered, a copy of the specified indirect access permanent file is retrieved. This copy becomes the new primary file and all other local files are dropped unless the ND parameter is used. For example:

```
LIB,ABC75/PW=13479,ND
```

In this example, a copy of indirect access permanent file ABC75 is retrieved and becomes the new primary file; all local files remain associated with the job. The previous primary file becomes a nonprimary temporary file. Only indirect access permanent files can be accessed with the LIB command.

The following commands can also be entered to access permanent files in the catalog of LIBRARY (differences in command function are described).

```
ATTACH,lfn=pfn/UN=LIBRARY,PW=password,M=m
```

This command must be entered to use any direct access permanent files in the catalog of LIBRARY. A temporary file is not created since all I/O operations are performed directly on the permanent file itself. However, if the user wishes to reference the file by a name other than its permanent file name, he can assign a local file name (lfn) in the command. The primary file name remains unchanged.

```
GET,lfn=pfn/UN=LIBRARY,PW=password
```

This command retrieves a copy of the specified indirect access permanent file (from catalog of LIBRARY) for use as a temporary file. The primary file remains unchanged unless lfn specifies the name of the current primary file (refer to description of GET command).

```
OLD,lfn=pfn/UN=LIBRARY,PW=password
```

The function of this command is identical in all respects to that of the LIB command.

The only files that can be accessed from the catalog of LIBRARY are those that allow alternate user access. All users, except user number LIBRARY, are considered to be alternate users. The rules for alternate user access are summarized as follows:

- Only those users with permission can access private files.
- All users can access semiprivate and public files providing they know the file name, password (if one exists for the file), type of access (direct or indirect), and user number (LIBRARY).

The following forms of the CATLIST command can be entered to obtain a listing of files that can be accessed in the catalog of LIBRARY.

```
CATLIST,LO=F,UN=LIBRARY (full listing)
```

```
CATLIST,UN=LIBRARY (file names only)
```

ADDITIONAL SYSTEM FEATURES

6

This section contains descriptions of the program editing, line resequencing, job suspension, and recovery features provided for the terminal user.

PROGRAM EDITING

To allow ease of editing the primary file when working with source programs, the system provides a line editing capability that can be used under any subsystem. For more detailed editing, Text Editor or XEDIT should be used (refer to the Text Editor Reference Manual or the XEDIT Reference Manual). The line editing capability allows the user to:

- Insert statements into a program.
- Delete statements from a program.
- Correct statements in a program.

The following example illustrates this program editing capability. The user enters the following in a new primary file:

```
10 LET A=1
20 LET B=50
30 FOR I=1 TO B
40 A=A*I
50 PRINT "FACTORIAL" ; I,A
60 NEXT I
70 END
```

If the user wishes to insert a statement between line numbers 10 and 20, he types:

```
15 LET Z=20
```

If the user wishes to delete the statement at line 20, he types:

```
20      (CR)
```

Finally, to change the statement at line number 30, the user types:

```
30 FOR I=1 TO Z
```

Leading zeros are not required even if the statement being deleted or corrected contains line numbers with leading zeros. For example, a line numbered 30 replaces, and is interchangeable with, a line numbered 00030.

When the file is listed, it has the following form.

```
10 LET A=1
15 LET Z=20
30 FOR I=1 TO Z
40 A=A*I
50 PRINT "FACTORIAL" ; I,A
60 NEXT I
70 END
```

To retain the latest (corrected) version of the primary file, the user enters:

SAVE

If the primary file is a copy of an indirect access permanent file, the user can replace the version in the permanent file system with the corrected version by entering:

REPLACE

To save the corrected file under a new name, the user enters:

SAVE,lnfn=pfn

lnfn Temporary file name.

pfn Permanent file name.

RESEQUENCING LINE NUMBERS

The RESEQ command enables the user to resequence or add line numbers to the primary file. The user should resequence only sorted files. This is done using the following command.

RESEQ,nnnnn,iiii,t

nnnnn New line number of the first statement (five digits maximum); if omitted, the system assumes nnnnn is 00100.

iiii Increment to be added to nnnnn; if omitted, the system assumes iiii is 0010.

t Type of file to be resequenced:

B Files that contain BASIC source code. BASIC statements that contain references to line numbers are updated when the line numbers are resequenced.

T Files that contain text source information. A five-digit line number, plus a blank, is added at the beginning of each line. No inspection is made for existing line numbers. Thus, if line numbers currently exist, they are not resequenced or deleted; two sets of line numbers result.

Other Any number at the beginning of a line is considered a line number and is resequenced according to the nnnnn and iii parameters; line numbers are added to the beginning of lines where none appear.

Default The user must change to the correct subsystem before entering the RESEQ command.

BASIC programs are resequenced under the BASIC subsystem. If the user resequences a BASIC source file, all line number references are updated. The format of each BASIC statement remains the same unless a new line number reference contains more digits than the original line number reference. Then, any multiple blanks surrounding the line number reference are used for the expanded number. In the following example, the user enters the RESEQ command while in the BASIC subsystem.

```
basic
READY.
lnh

95 ON SGN(A)+2 GOTO 100,110,120      'COMMENT
100 PRINT "A IS NEGATIVE"
105   GOTO    130                      'COMMENT
110 PRINT "A IS ZERO"
115   GOTO    130                      'COMMENT
120 PRINT "A IS POSITIVE"
130 LET B=A+1
135 END

READY.
reseq

READY.
lnh

00100 ON SGN(A)+2 GOTO 00110,00130,00150 'COMMENT
00110 PRINT "A IS NEGATIVE"
00120   GOTO  00160                   'COMMENT
00130 PRINT "A IS ZERO"
00140   GOTO  00160                   'COMMENT
00150 PRINT "A IS POSITIVE"
00160 LET B=A+1
00170 END

READY.
```

FORTRAN programs are resequenced under FORTRAN or FTNTS subsystem; statement numbers and references to statement numbers are not changed.

All parameters supplied in the RESEQ command are order-dependent; that is, they must be entered in the order they appear in the command. Omitted parameters must be indicated by a comma if other parameters are to follow. For example, the command:

RESEQ,50,,T

adds line numbers to the primary (text) file without checking for existing line numbers. Thus, the first line is numbered 00050 and each succeeding number is increased by 10 (default value). If the T parameter is not specified in the command, the two preceding commas are not necessary.

If the starting line number (nnnnn parameter) is longer than five digits or if the increment (iiii parameter) causes the line numbers generated to exceed five digits, the message:

LINE NUMBER LIMIT EXCEEDED.

is returned to the terminal. In addition, the message:

RESEQ NUMERIC PARAM ERROR.

is returned if the user enters a nonnumeric value for either the nnnnn or the iiii parameters.

The RESEQ command retains at most one trailing space on a blank line. During a sort, the system deletes blank lines with one or no trailing spaces. Consequently, a RESEQ command followed by a sort causes all blank lines to be deleted from the file.

JOB SUSPENSION

A user can suspend his job at any time during program execution (RUN command issued). If an ASCII code terminal is being used, this is done by pressing the BREAK key. If the program is actively transmitting output to the terminal, the I key can also be used to suspend the job. If a correspondence code terminal is being used, only the ATTN key can be used to suspend the job. The system responds:

INTERRUPTED

A job is also suspended automatically by the system when:

- The job has exceeded its time limit. The time limit is initially set to 64 CPU seconds at login, although the user can change this value through use of the SETTL command (refer to section 4).
- The job or job step has exceeded its SRU limit. The SRU limits are initially set to 320 SRUs at login, although the user can change this value through use of the SETASL or SETJSL commands (refer to description in section 4).
- A successful recovery has been performed (refer to Recovery in this section).

If the job is suspended by the user (interrupted), the user can perform one of the following:

- ④ Continue. A portion of the output is lost if the program is transmitting output to the terminal when suspended. When output is complete, program execution continues. If the TAPE command is in effect (tape mode), carriage return is ignored when entered on any empty input line. Therefore, P ④ must be issued to resume execution.
- P ④ Proceed. If the program is transmitting output to the terminal when suspended, the system discards the data generated by the program before the user interrupts it. The amount of output discarded depends on the program being executed. Program execution continues.
- STOP ④ Stop. Terminates the job step.
- Other ④ Stops. Terminates the job step.

If the job step is suspended because it exceeded its time limit, the message:

TIME LIMIT
ENTER T TO CONTINUE OR CR KEY TO STOP:

is issued and the user can enter one of the following:

<u>Entry</u>	<u>Description</u>
T ④	Increases the central processor time limit by 64 CPU seconds. Job execution continues.
T,nnnnn ④	Increases the central processor time limit by nnnnn decimal seconds. Job execution continues. The user can enter octal seconds by specifying a B after the octal number.
T,* ④	Increases the central processor time limit to the user's maximum. Job execution continues.
④	Causes the job to go through normal abort procedures (for example, EXIT processing which can be useful when using procedure files).
Termination sequence ④	Terminates the job step. Subsequent control statements, if any, are not processed.
STOP ④	Terminates the job step. Subsequent control statements, if any, are not processed.

Any increase to the central processor time limit through either T or T,nnnn is in effect only for the current job step. When the job step terminates, the central processor time limit reverts to its original value, previously set by default or by the SETTL command.

If the user enters something other than one of the above, the system prompts the user again.

If the job is suspended because a job step or the job itself exceeded its SRU limit, the message:

SRU LIMIT
ENTER S TO CONTINUE OR CR KEY TO STOP:

is issued. The user can enter one of the following:

<u>Entry</u>	<u>Description</u>
S ^(CR)	Increases the SRU limit by 320 units. Job execution continues.
S,nnnn ^(CR)	Increases the SRU limit by nnnn decimal units. Job execution continues. The user can enter octal units by specifying a B after the octal number.
S,* ^(CR)	Increases the SRU limit to the user's maximum. Job execution continues.
^(CR)	Causes the job to go through normal abort procedures.
Termination sequence ^(CR)	Terminates the job step. Subsequent control statements, if any, are not processed.
STOP ^(CR)	Terminates the job step. Subsequent control statements, if any, are not processed.

Any increase to the SRU limit through either S or S,nnnn is in effect only for the current job step. When the job step terminates, the account block SRU limit and job step SRU limit revert to their original values, set by default or by the SETASL and SETJSL commands, respectively. Entering S,* in response to an SRU LIMIT message could cause the time-sharing session in progress to exceed the account block SRU limit set by the SETASL command or by default. The system then issues an SRU LIMIT message after a job step has begun. To remedy this situation, use the SETASL command to raise the account block SRU limit above the current number of accumulated SRUs for the time-sharing session.

If the user enters something other than one of the above, the system prompts the user again.

The following example illustrates the use of the I key to interrupt a program (the I key is effective only for ASCII code terminals).

```

lnh
00100*INTERRUPT SEQUENCE
00110 PROGRAM DO (INPUT,OUTPUT)
00120 INTEGER E
00130 DO 10 E=1,100
00140 WRITE 5,E
00150 5 FORMAT(I3)
00160 10 CONTINUE
00170 READ*,I,J
00180 WRITE 15,I,J
00190 15 FORMAT(I3,I3)
00200 STOP
00210 END
READY.

```

```

rnh
1
2
i 3 ←
*INTERRUPTED*
(CR) ←
33
34
i 35 ←
*INTERRUPTED*
p (CR) ←
65
66
i 6 *INTERRUPTED* ←
p (CR) ←
? 4,5
4 5
SRU      0.173 UNTS.
RUN COMPLETE.

```

User interrupts the program by pressing the I key.

User resumes execution by pressing (CR) (some output is lost).

User again presses the I key.

User directs the system to discard data. As execution continues, more data is printed at the terminal.

User again presses the I key.

User again directs the system to discard data. All output has been generated. Program execution continues.

The number of times the user would have to interrupt a program and enter P (CR) before all output is generated depends on the program.

In the following example, a program exceeds its allocated time limit. The program then runs to completion when a new time limit is entered.

```

RECOVER/SYSTEM: ftnts,old,nm
READY.

```

```

lnh
00100 PROGRAM T(OUTPUT)
00110 DO 6 I=1,2500
00120 DO 6 J=1,4000
00130 A=1
00140 6 CONTINUE
00150 END
READY.
setl,10
READY.
rnh
*TIME LIMIT*
ENTER T TO CONTINUE OR CR KEY TO STOP:
t
SRU      66.010 UNTS.
RUN COMPLETE.

```

RECOVERY

During job processing, recovery may be necessary when

- The terminal is accidentally disconnected from the system.
- A system malfunction occurs which requires a restart.

The terminal is placed in recovery state whenever it is disconnected from the system without being logged out (providing that it is not already in recovery state). The user has 10 minutes to initiate recovery.

If there is a system malfunction which requires a restart, the user has 10 minutes from the time the system is restarted to initiate recovery.

To recover, a user completes the login sequence to the point where the system requests RECOVER/SYSTEM (or RECOVER/CHARGE). For example:

yy/mm/dd. hh.mm.ss.

CDC MULTI-MODE OPERATING SYSTEM. NOS 1

FAMILY: AAA

USER NUMBER: USER123,ABCDEFG ← User enters number and password.

TERMINAL: 15, TTY ← Terminal number/type identifier.

RECOVER/SYSTEM:

In response, the user enters

RECOVER,nnn

nnn Terminal number being used when the failure occurred. This number was indicated when the user initially logged in (refer to preceding example). If the same terminal number is indicated when the user logs in to recover, this parameter is not required.

The RECOVER command is valid only when entered in response to the RECOVER/SYSTEM or RECOVER/CHARGE request during the login sequence. If the user's terminal number is not the same as before the failure, the previous terminal number must be entered with the RECOVER command (nnn parameter). This may occur when the failure is due to a dropped line connection. In this case, a different terminal number may be assigned at login. If the RECOVER command is not entered where indicated, the user may log out, log in again, and then enter the RECOVER command correctly.

If the system responds RECOVERY IMPOSSIBLE, one of the following conditions exists.

- The user entered nnn incorrectly. The user can reenter the RECOVER command with the correct terminal number.
- The system has no record of the specified user being logged in on the given terminal number within the past 10 minutes.
- The user's system information is incorrect due to system malfunction.

If the system responds INVALID TERMINAL NUMBER, the user entered characters other than digits or entered a number that is too large.

If the user receives either of the preceding messages, he should check the terminal number he entered. If he has not given the correct number, he may reenter RECOVER with the correct terminal number.

If recovery is successful, the system responds:

RECOVERY COMPLETE.

LAST COMMAND	= cmdname
JOB STATUS	= stat
NEXT OPERATION	= nnnn

ENTER *CR* TO CONTINUE:

cmdname Name of last command processed. If source code was being entered, the word source is output.

stat One of the following job status messages.

	<u>Message</u>	<u>Description</u>
	IDLE	No activity.
	EXECUTING	Job is in execution.
	OUTPUT AVAILABLE	Output from an executing job is available.
	OUTPUT LOST	Output from an executing job is lost.
	INPUT REQUESTED	Executing job is requesting input.
	INPUT LOST	Last line of input data is lost.
nnnn	One of the following next operation messages.	

	<u>Message</u>	<u>Description</u>
	ENTER COMMAND	System is ready to process the next command.
	RERUN OR CONTINUE	If output, binary, or auto mode input is lost, and if the output or input are significant, the user should rerun the job.
	ENTER DATA	Executing job is requesting input.
	REENTER DATA	User should reenter lost input data.
	CONTINUE	Executing job is ready to continue processing.

If the user wants to continue his job, he enters \textcircled{C} . If the job was previously transmitting output, the output data normally resumes a few lines prior to where the interruption in service occurred. If he wants to terminate the job step, he enters any character followed by \textcircled{C} .

In general, recovery in the system is designed to provide minimum inconvenience to the user with maximum security. However, under certain circumstances, recovery of user information cannot be perfect. The following are two examples of these circumstances.

- In some cases, a few lines of output may be lost when a phone line is disconnected just as a job is being restarted to generate more output. This actually occurs before current output is exhausted.
- If a phone line is disconnected when a user is entering lines, he can normally expect to lose the last few lines he has entered.

Whenever the user and terminal numbers of a user requesting a recovery state match the user and terminal numbers of a user already in a recovery state, the user already in the recovery state remains there and the other user is logged out. If a user is disconnected after logging in but before recovery from a previous disconnect, the latter time is processed as a normal logout to protect the user from an intermittent phone line failure. The user can then log in again and enter the RECOVER command to continue the job.

The following is an example of a normal recovery.

```
yy/mm/dd. hh.mm.ss.  
CDC MULTI-MODE OPERATING SYSTEM.      NOS 1  
FAMILY: aaa  
USER NUMBER: user123  
PASSWORD  
*****  
TERMINAL:    16, TTY  
RECOVER/SYSTEM: recover,12  
  
RECOVERY COMPLETE.  
LAST COMMAND   = RUN  
JOB STATUS     = IDLE  
NEXT OPERATION  = ENTER COMMAND  
ENTER *CR* TO CONTINUE:
```

The user enters **CR** to continue job processing.

The system replies:

```
READY.
```

The following example illustrates a sample recovery.

```
yy/mm/dd. hh.mm.ss.  
CDC MULTI-MODE OPERATING SYSTEM.      NOS 1  
FAMILY: aaa  
USER NUMBER: user123,abcdefg  
TERMINAL: 15, TTY  
RECOVER/SYSTEM: ftnts,old,aa  
READY.  
lnh  
  
00100 PROGRAM AA(INPUT,OUTPUT)  
00110 PRINT 4  
00120 4 FORMAT(*ENTER 2 NUMBERS IN THE FORMAT XXX.X XXX.X*)  
00130 READ 3,A,B  
00140 3 FORMAT(F5.1,1X,F5.1)  
00150 C=A*B  
00160 PRINT 5,C  
00170 5 FORMAT(*A MULTIPLIED BY B EQUALS*,1X,F15.2)  
00180 STOP  
00190 END
```

READY.

rnh

ENTER 2 NUMBERS IN THE FORMAT XXX.X XXX.X

? ← The user is disconnected before entering data and logs in again.

```
yy/mm/dd. hh.mm.ss.  
CDC MULTI-MODE OPERATING SYSTEM.      NOS 1  
FAMILY: aaa  
USER NUMBER: user123,abcdefg  
TERMINAL: 23, TTY  
RECOVER/SYSTEM: recover,15
```

RECOVERY COMPLETE.

```
LAST COMMAND = RNH  
JOB STATUS = INPUT REQUESTED  
NEXT OPERATION = ENTER DATA  
ENTER *CR* TO CONTINUE: ← The user enters CR to continue.
```

? 20.0 20.0 ← The user enters data.

A MULTIPLIED BY B EQUALS 400.00

SRU 0.670 UNTS.

RUN COMPLETE.

This example illustrates another recovery situation.

```
enquire
TERMINAL:      4, TTY
SYSTEM:        BASIC
FILE NAME:    BB
STATUS:        IDLE
MESSAGE:
```

lnh

```
10 REM THIS IS A TEST
20 REM PRINT B
30 PRINT "PRINT B"
40 LET B=1+3
50 PRINT B
60 END
```

READY.

The user is disconnected.

```
yy/mm/dd. hh.mm.ss.
CDC MULTI-MODE OPERATING SYSTEM.      NOS 1
FAMILY: aaa
USER NUMBER: user123,abcdefg
TERMINAL:      32, TTY
RECOVER/SYSTEM:
```

The user is disconnected again before recovering.
This is processed as a normal logout.

```
yy/mm/dd. hh.mm.ss.
CDC MULTI-MODE OPERATING SYSTEM.      NOS 1
FAMILY: aaa
USER NUMBER: user123,abcdefg
TERMINAL:      32, TTY
RECOVER/SYSTEM: recover,4
```

The user enters the RECOVER command.

RECOVERY COMPLETE.

```
LAST COMMAND = LNH
JOB STATUS   = IDLE
NEXT OPERATION = ENTER COMMAND
```

ENTER *CR* TO CONTINUE:

The user enters \textcircled{C} to continue.

READY.

lnh

```
10 REM THIS IS A TEST
20 REM PRINT B
30 PRINT "PRINT B"
40 LET B=1+3
50 PRINT B
60 END
```

READY.

ERROR CONTROL

When the hardware detects an error condition, the system terminates the job step and dumps the job's exchange package and a portion of the job's central memory to a local mass storage file called ZZZDUMP. The system does not rewind ZZZDUMP before or after the dump. The following message is printed.

EXCHANGE PACKAGE/MEMORY DUMP ON FILE ZZZDUMP.

To examine the exchange package and memory dump, rewind and list file ZZZDUMP. Refer to the NOS Reference Manual, volume 1, for more information on the exchange package and error conditions.

REMOTE JOB SUBMITTAL

7

GENERAL DESCRIPTION

Remote job submittal enables a validated time-sharing user to submit batch job deck images to the system for processing. The batch job image can be created interactively at the time-sharing terminal and then submitted to the local batch queue by entering the SUBMIT command. The user is no longer interactive with the job once it has been submitted. Instead, the job is treated in the same manner as a job entered from the card reader at the local computer site or from a remote batch terminal. The batch job image can consist of control statements, control language statements, program statements, and data. The structure of the batch job image must also be the same as its counterpart on cards (job deck). The user can specify how the job is to be structured through use of the reformatting directives described in this section. Refer to the NOS Reference Manual, volume 1, for information concerning job structure and a description of the control statements available.

ENTERING SUBMIT COMMAND

The format of the SUBMIT command is:

SUBMIT,lfn,q,NR.c

- lfn Specifies name of local file to be submitted for processing as a deferred batch job. If lfn is omitted, the primary file name is assumed.
- q Specifies the disposition of job OUTPUT as follows:
 - B The job is processed as a local batch job and file OUTPUT is printed at the central site.
 - N The job is processed as a local batch job and file OUTPUT (including banner page and dayfile) is not printed. The default is N.
 - E The job is processed as a remote batch job and file OUTPUT is routed for printing at a remote batch terminal.
- NR Specifies no rewind; inhibits rewind of file specified by reformatting directive cREAD (refer to Reformatting the Submit File in this section). If omitted, file specified by cREAD directive is automatically rewound.
- c Specifies escape character used to identify reformatting directives in the file to be submitted (lfn). If omitted, the system assumes c is a slash.

After the job has entered the batch queue, the system responds:

hh.mm.ss.jobname
READY.

- hh.mm.ss. Time that the job entered the batch queue (hours, minutes, seconds).
- jobname Seven-character job identification name assigned to the job by the system when it is submitted.

The last three characters of the job identification are entered with the ENQUIRE,JN=ccc command to obtain the status of a job after it has entered the system (refer to ENQUIRE command in section 4).

REFORMATTING THE SUBMIT FILE

The submit file (lfn in the SUBMIT command) is a file which contains the batch job image submitted for processing. The reformatting directives described in this section are provided to aid the user in preparing the submit file. When the SUBMIT command is entered, the submit file is reformatted according to the directives that appear in the file and is then placed in the batch input queue awaiting execution. Thus, several of the directives are provided to format the submit file to meet the structural requirements of a batch job (refer to the NOS Reference Manual, volume 1).

Each line in the submit file preceded by an escape character is recognized as a reformatting directive. The escape character to be used can be defined in the SUBMIT command (/ by default). Throughout this description, the letter c, preceding a directive, denotes the escape character. Reformatting directives can be interspersed throughout the submit file as long as transparent submit mode is not in effect. Transparent submit mode is selected by the cTRANS directive; it requires that the user observe special rules when inserting subsequent directives into the file (refer to the description of cTRANS and cNOTRANS directives).

Reformatting directives are not processed unless the first line of the submit file contains the cJOB directive. In addition, the first two statements following the cJOB directive (second and third lines of the submit file) must be a job and USER statement, respectively. All following information is determined by the user. Thus, the first lines of a submit file that is to be reformatted before processing should be:

00100 cJOB	Reformatting directive.
00110 jobname,...	Job statement.
00120 USER,...	USER statement.
00130 CHARGE,...	Charge and project numbers, if required.

NOTE

The job and USER statements are required in all batch jobs; they must be present even if the cJOB statement is omitted. If the user submits a job with an illegal USER statement, the system logs the user out without a dayfile message and decrements his security count. The user can log in again if his security count is greater than zero. For more information on use of the USER control statement, refer to section 8.

Although text mode can be used to create a submit file, it is not necessary since each line may have a leading line number. The cSEQ and cNOSEQ directives determine, during reformatting, if line numbers in the submit file will be retained. The cJOB directive is required when any line numbers are to be removed. The user can include line numbers on the entire submit file and can specify which line numbers are to be removed during reformatting. This is especially useful if the submit file contains a BASIC program where line numbers are a requirement of the language.

The reformatting directives available are described as follows:

<u>Directive</u>	<u>Description</u>
cJOB	Indicates that the submit file is to be reformatted and selects the following default reformatting directives. The default directives remain in effect until specified otherwise. cNOTRANS (disabled by cTRANS) cSEQ (disabled by cNOSEQ) cPACK (disabled by cNOPACK)
	The cJOB directive must be the first line of the submit file. If omitted, the file is not reformatted.
cEOR	Indicates that an end-of-record mark is to be placed at this point in the submit file during reformatting.
cEOF	Indicates that an end-of-file mark is to be placed at this point in the submit file during reformatting.
cSEQ	Indicates that line numbers will be removed from all subsequent lines during reformatting (default value).
cNOSEQ	Reverses the effect of the cSEQ directive. No attempt is made to remove leading line numbers from subsequent lines.
cPACK	Indicates that all succeeding internal EOR and EOF marks will be removed during reformatting (default value). This directive applies only to internal EOR and EOF marks that currently exist. The cEOR and cEOF reformatting directives are not affected.
cNOPACK	Reverses the effect of the cPACK directive. Requests the system not to discard succeeding internal EOR and EOF marks that currently exist.
cTRANS	Indicates transparent submit mode. When encountering this directive during reformatting, the system checks the next line of the submit file for an additional reformatting directive. If one exists, it is processed and the following line is checked. This continues until a line that does not contain a reformatting directive is encountered. Transparent submit mode is then selected and all directives that exist on subsequent lines are ignored until an internal EOR or EOF is encountered (not cEOR or cEOF). The cPACK and cNOPACK directives determine if the internal EOR or EOF will be retained. The line following the internal EOR or EOF mark is then checked for a reformatting directive. If one exists, it is processed and the following line is checked. All directives are processed until a line that does not contain a reformatting directive is encountered. This causes transparent submit mode to be reset unless a cNOTRANS directive was encountered. This process continues until either the end of the submit file is reached or a cNOTRANS directive following an internal EOR or EOF is encountered.
	The cTRANS directive is typically used in conjunction with the cREAD directive. This allows the user to copy the contents of an existing file into the submit file at the location of the cREAD directive. Because the file is read in transparent submit mode, no check for reformatting directives is attempted until an internal

<u>Directive</u>	<u>Description</u>
	EOR or EOF is encountered. The cREAD directive must follow the cTRANS directive and must be located before the first succeeding line that is not a reformatting directive. Otherwise, transparent submit mode is selected before the cREAD directive is encountered, and the cREAD directive is ignored.
	The cSEQ or cNOSEQ directive in effect before transparent submit mode is selected has no effect upon the submit file or the file being read (cREAD) while transparent submit mode is in effect. However, the cPACK or cNOPACK directive in effect before transparent submit mode is selected remains in effect after it is selected.
cNOTRANS	Reverses the effect of the cTRANS directive and informs the system that the submit file is to be examined on a line-by-line basis. All directives encountered in the submit file while the cNOTRANS directive is in effect are processed. This directive is initially selected by default; it remains in effect until a cTRANS directive is encountered in the submit file.
	Caution should be observed in the placement of this directive in the submit file. If transparent submit mode is selected, this directive can possibly be ignored unless it immediately follows either a cREAD directive in the submit file or an internal EOR or EOF mark.
cREAD, lfn	Requests that the system read the contents of the specified file (lfn) and insert that file in place of the cREAD directive in the submit file, during reformatting. Reading terminates when an EOF or EOI is encountered on lfn. If the file to be read is not currently a local file, the system automatically attempts a GET and then an ATTACH on the file. If the file is not specified in the directive, file TAPE1 is assumed. If the file specified cannot be found, the message NO READ FILE FOUND - lfn is issued to the job dayfile, and the job is terminated. If the read file is found to be busy (direct access files only), the message READ FILE BUSY - lfn is issued to the job dayfile, and the job is terminated. The file specified by lfn in the cREAD directive is automatically rewound before the read operation, unless the NR parameter is specified in the SUBMIT command. In this case, the CREWIND directive must precede the cREAD directive in the submit file to rewind file lfn before the read operation begins. NOS returns all files specified in cREAD directives before completion of the job. If the cPACK directive is in effect at the time of the read, all internal EOR marks are removed. If the cNOPACK directive is in effect, all internal EOR marks are read into the submit file in the proper position during reformatting. Unless transparent submit mode is in effect when file lfn is read, each line of that file is also checked for a reformatting directive. Any directives contained in the file, except another cREAD, are processed. The cREAD directive cannot be nested. In addition, any directives in effect before the cREAD directive is processed remain in effect for the file being read, unless transparent submit mode is selected. Then, only the cPACK or cNOPACK directives remain in effect for the file being read. Moreover, only those directives that immediately follow an internal EOR in the file being read are processed. If the file to be read is a binary file, it is recommended that the cTRANS directive be used. This is to ensure that binary data is not mistaken for a reformatting directive. If used, the cTRANS directive should immediately precede the cREAD directive in the submit file.

<u>Directive</u>	<u>Description</u>
cREWIND,lfn	Requests that the system rewind file lfn to BOI. If lfn is not supplied, file TAPE1 is assumed. This directive is required only if the NR parameter is included in the SUBMIT command. Otherwise, file lfn is automatically rewound. This directive is used in conjunction with the cREAD directive. Thus, to rewind a file before the read operation begins, this directive must precede the cREAD directive in the submit file.
c ₁ EC=c ₂	Indicates that the escape code character is to be changed from c ₁ (current escape code) to c ₂ (new escape code). The new escape code is used to recognize all subsequent reformatting directives until further change.

No line in a submit file should exceed 150 6-bit characters. The system processes the first 80 characters of the line as a batch control statement and discards the final 70 characters, which can be used for comments or sequence numbers. These final characters do not appear in the dayfile messages. If any input line exceeds 150 characters, the results are unpredictable.

The following procedure shows how a BASIC program and associated batch control statements are reformatted into the structure of a batch job. This batch job is then submitted from the terminal by means of the SUBMIT command.

A BASIC program for generating factorials is entered at the terminal and made into the following indirect access permanent file.

```

basic
READY.
new,basprog

READY.
00500 a=362880
00510 z=15
00520 for i=10 to z
00530 a=a*i
00540 print "factorial" i "=" a
00550 next i
00560 end
save

READY.

```

To submit this program as a batch job from a terminal, the user includes it in a submit file. Then the user sends the submit file to the local batch queue by entering the SUBMIT command. This submit file has the batch control statements that direct job execution, according to user requirements, as well as any reformatting directives that structure the batch job according to the user's particular specifications.

This submit file is created and saved as follows:

new, sbmt job

READY.

auto ← Line numbers facilitate modification while the file is being created. These line numbers are stripped off during reformatting, unless the user specifies otherwise (NOSEQ).

00100 /job ← The user has chosen the character / to identify reformatting directives. The presence of this directive at the beginning of the submit file indicates that there are other reformatting directives in the file. If the submit file is to be used without any reformatting, this directive is not necessary.

00110 jldjob(t500, cm40000) ← This is the job statement required at the beginning of every batch job. 500 is the maximum amount of time, in decimal seconds, the user allows the central processor to spend on any one job step. 40000 is the maximum amount of central memory storage, in octal units, the user allows for each job step. Refer to the job statement in the NOS Reference Manual, volume 1.

00120 user(point, into, fama) ← This is the USER statement with user number, password, and family. If the user number, password, or family is incorrect, the system logs the user out. Refer to section 8 for information concerning restrictions associated with the use of this statement.

00130 charge(6134, prj7) ← This is the CHARGE statement with charge number and project number.

00140 basic(l=results, k=results) ← This calls the BASIC compiler, initiates execution of the object code, and puts the listable compiler output and execution output on a file called RESULTS. This is done so the output can be printed out at the user's convenience.

00150 replace(results) ← This makes the output a permanent file so that it is not lost after job execution. A REPLACE is used instead of a SAVE since the user may want to execute this job several times with variations. A REPLACE keeps putting the new information in place of the old.

00160 rewind(input) ← File INPUT is rewound in preparation for the following command. Ordinarily, the pointer in batch input is at the beginning of the second record, the program. The user wants to copy the entire input file, including the first record of control statements.

```

00170 copycf(input,reformat) The entire batch job as reformatted is copied to a file called
                               REFORMAT.

00180 replace(reformat) ← The reformatted file is made permanent so that it is not lost
                           after job execution. The user can then list this file to see
                           what the reformatting looks like. A REPLACE is used
                           instead of a SAVE to allow future repetitions of this job.

00190 dayfile(myday) ← The job execution history is put on a file called MYDAY.

00200 replace(myday) ← The history file is made a permanent file for future reference.

00210 exit. ← Normal job processing ends if no errors have been en-
                  countered. However, if the system encounters any errors in
                  the preceding statements, it skips to this statement and then
                  executes those that follow.

00220 dayfile(myday) ← The job execution history is put on a file called MYDAY.

00230 replace(myday) ← The history file is made a permanent file for future
                           reference, if the job fails.

00240 /eor ← This reformatting directive causes an end-of-record mark to
                  be placed here in the batch job. This signals the end of the
                  first record, the control statements.

00250 /noseq ← This reformatting directive tells the system to leave all line
                  numbers that follow this directive when reformatting the
                  submit file. This is necessary since the second record is a
                  BASIC program which requires line numbers.

00260 /read,basprog ← This reformatting directive inserts file BASPROG at this
                           point. The system first searches local files and then the
                           user's permanent files.

00270 /eof ← This final reformatting directive puts an end-of-file mark
                  here, indicating the end of the second record and the end of
                  the file. It is included in this example only for clarity.
                  Because the system automatically writes an end-of-file, the
                  directive is not required.

00280 *DEL* ← The user presses the ESCAPE key to stop the automatic line
                  numbers.

```

To submit this job, the user enters the time-sharing command

```
submit(sbmt job)
```

The file enters the system, and the system immediately assigns it a unique name, consisting of the first four letters of the user's index hash followed by three characters which the system derives from its internal alphabetical sequencing. It then prints out the time and the name it has assigned. In this case, it is

10.23.11.AJAIAAJ

The reformatting directives are executed, and the resulting file is sent to the batch input queue.

The user is now interested in:

- What does the reformatted input look like?
- What are the results of job execution?

The answers are on the permanent files REFORMAT and RESULTS. However, these files do not exist until after job execution. The user can verify completion of job execution by entering the ENQUIRE command (section 4), using the last three characters of system-supplied job name AAJ to identify the job.

```
enquire,jn=aaJ
```

When the reply is

```
AJAIAAJ NOT FOUND.  
READY.
```

the user knows the job has finished execution. He can then obtain the following two listings.

```
get,reformat  
READY.
```

```
lnh,f=reformat
```

This produces the following listing.

```
JLDJOB(T500,CM40000)  
USER(POINT,INTO,FAMA)  
CHARGE(6134,PRJ7)  
BASIC(L=RESULTS,K=RESULTS)  
REPLACE(RESULTS)  
REWIND(INPUT)  
COPYCF(INPUT,REFORMAT)  
REPLACE(REFORMAT)  
DAYFILE(MYDAY)  
REPLACE(MYDAY)  
EXIT.  
DAYFILE(MYDAY)  
REPLACE(MYDAY)  
00500 A=362880  
00510 Z=15  
00520 FOR I=10 TO Z  
00530 A=A*I  
00540 PRINT "FACTORIAL" I "=" A  
00550 NEXT I  
00560 END  
READY.
```

```
get,results  
READY.
```

```
lnh,f=results
```

This produces the following listing.

```
00500 A=362880  
00510 Z=15  
00520 FOR I=10 TO Z  
00530 A=A*I  
00540 PRINT "FACTORIAL" I "=" A  
00550 NEXT I  
00560 END  
  
FACTORIAL 10 = 3628800  
FACTORIAL 11 = 39916800  
FACTORIAL 12 = 479001600  
FACTORIAL 13 = 6.22702E+9  
FACTORIAL 14 = 8.71783E+10  
FACTORIAL 15 = 1.30767E+12  
READY.
```

ERROR PROCESSING

If the user determines that an error occurred during processing of his job, he can reference a listing of the job dayfile as an aid in identifying the cause of the error. The job dayfile contains a record of the job processing activity and is disposed to the local batch queue or the remote batch queue for printing when the job is terminated. However, all output is dropped at job termination when a batch job is submitted from a time-sharing terminal, unless the B or E option is specified in the SUBMIT command. This includes the dayfile output as well as the job output. In this event, the user can make provisions within the job to save the contents of the dayfile if an error in processing occurs. This is done by including the following control statements at the end of the job control statement record (first record of the submit file).

```
    .  
    .  
    .  
    EXIT.  
    DAYFILE(lnf)  
    REPLACE(lnf)  
    .  
    .
```

When an error condition occurs during job processing, the system searches the job control statement record for an EXIT statement. If an EXIT statement is found, the error condition is cleared and the control statements that follow are processed. In this case, the contents of the job dayfile are copied to the temporary file specified by lfn. That file is then saved for future reference.

If an error does not occur during job processing, the EXIT statement indicates where to terminate normal control statement processing.

BATCH SUBSYSTEM

8

The batch subsystem extends the capability of the terminal user by allowing him to enter batch control statements as well as the time-sharing commands. The control statements are entered and processed one at a time, and all output is returned to the terminal unless otherwise specified. Refer to the NOS Reference Manual, volume 1, for a description of the batch control statements available and information concerning their use.

To enter the batch subsystem, the user types:

BATCH,nnnnn

nnnnn Initial running field length for subsequent job steps. Entering this value is equivalent to entering the RFL,nnnnn control statement. Refer to the NOS Reference Manual, volume 1, for further information. If nnnnn is omitted, 0 is assumed.

The system responds[†]:

\$RFL,nnnnn.
/

nnnnn Specifies field length.

The user can then enter a time-sharing command or any valid batch control statement up to 78 characters in length. Time-sharing commands cannot be abbreviated under the batch subsystem; the entire command or control statement must be entered to be valid. The user can also end each control statement with a period, but this is not required since the system automatically terminates each line of batch input with a period. However, if comments are included in the control statement, the user is required to enter a terminator (either a right parenthesis or a period) between the command and the comment. The system responds by printing the last user dayfile message. The user should check the dayfile for other messages and then refer to the appropriate manual if clarification is needed. If the command or control statement is not valid or is entered incorrectly, the system responds:

ILLEGAL CONTROL CARD.

Because the processing of certain control statements could enable unauthorized access to dumps of privileged areas of memory, thus violating system security, the following control statements are not processed if entered individually or as the first statement in a procedure file from a time-sharing terminal.

CKP	PBC
DMD	RBR
DMP	RESTART
LBC	WBR
LOC	

[†] If the batch subsystem is selected automatically by retrieving an indirect access permanent file with the OLD command, the field length is not given (and may not be the default value). Refer to the description of the SAVE command in section 5 for additional information.

If an attempt is made to enter any of these restricted control statements in the previously discussed manner, the following informative message is printed.

SECURE MEMORY, DUMP DISABLED.

The control statements listed are processed as from a batch job if they are included in a procedure file and are not the first statement (refer to -pname, BEGIN, or CALL command in section 4).

Sequences of loader statements can be entered only if they are in a procedure file. The system must read the entire sequence of loader statements (that is, from the initial loader statement to the completion statement) before acting on any one of them. Refer to the CYBER Loader Reference Manual for further information on loader statements.

All time-sharing commands for which there is a corresponding batch control statement can be included in a procedure file (refer to the NOS Reference Manual, volume 1). The remaining time-sharing commands cannot be included in procedure files.

The following command allows the user to force the system to process a valid time-sharing command (ccccccc), normally interpreted by the time-sharing executive, as a control statement.

X,eeeeeee

For example, to call the BASIC compiler, the user enters:

X,BASIC

rather than the command:

BASIC

which changes the subsystem from batch to BASIC.

To exit from the batch subsystem, the user must type the entire command (FORTRAN, FTNTS, BASIC, EXECUTE, ACCESS, TRAN, or NULL) to call one of the other subsystems. The user may also change the subsystem unintentionally by entering the OLD command. The OLD command obtains a different primary file but may also select the subsystem that was active when the specified file was originally saved (refer to the description of the SAVE command in section 5).

NOTE

Although a user can enter any batch control statement from a time-sharing terminal, the output produced by several of these statements is formatted for transmission to a line printer (137 characters per line). Through use of the LO72 control statement, the user can format the output for transmission to a time-sharing terminal (72 to 150 characters per line) but some data may be lost. Refer to the description of the LO72 control statement in the NOS Reference Manual, volume 1.

The time-sharing user should be extremely cautious in using the USER control statement to specify a different user number and password. For security reasons, some sites do not permit a time-sharing user to enter a USER statement, unconditionally logging out any user attempting to do so. At all sites, an invalid user number in a USER statement causes the system to log out the user and decrease the security count of the user number under which he was logged in. The security count is the maximum number of security violations (such as attempting to use an invalid user number) that a user is allowed. A security count is assigned to each user number by site personnel. If the user number has exhausted its security count, the associated user is denied all access to the operating system until the security count is reset by site personnel.

When the USER statement is allowed, all permanent file accesses are performed under the user number specified in the most recently entered USER statement, in accordance with the file limits and validations applicable to that user. However, all job-related restrictions (charge required, memory limits, time limits, and so on) remain those of the user specified during login.

The USER control statement and the USER time-sharing command do not perform the same function. Refer to NOS Reference Manual, volume 1, for more information on the USER control statement. The USER command is used in the access subsystem to determine another user's terminal number. Refer to section 4 for information on the USER time-sharing command.

PAPER TAPE OPERATIONS

9

GENERAL DESCRIPTION

Paper tape is used for preparing input off-line (when time is not charged) and entering it on-line (when time is charged). This ensures accuracy and speed when most needed. An input tape can include programs, data, and commands. Accordingly, it is possible for the entire terminal operation, after login, to be run from paper tape.

TELETYPEWRITER

This discussion assumes a typical teletypewriter (model 33) with a paper tape punch and a paper tape reader.

The paper tape punch has four buttons with the following labels and use.

<u>Button</u>	<u>Description</u>
ON	Turns the punch on.
OFF	Turns the punch off.
REL.	Releases the feedwheel so one can freely move the tape through the punch head.
B.SP.	Backspaces the tape one row of holes each time it is pressed. This is used to make corrections when preparing a tape off-line (refer to Corrections in this section).

The paper tape reader has one switch with four positions. Position labels and use are as follows:

<u>Position</u>	<u>Description</u>
START	Starts the reader. After switch is momentarily held in this position, it snaps back to the AUTO position and reading continues.
STOP	Immediately stops reading.
AUTO	This position is used in conjunction with the input and processing of commands and data in tape mode (refer to Tape Mode in this section). It allows the tape reader to be turned off and on so that processing of each command or line of data can be completed before additional input is entered.
FREE	Releases the feedwheel so user can freely slide the tape in and out of the read head.

NOTE

On teletypewriters lacking an AUTO position, reading must be manually restarted each time it has been stopped.

CONTROL CHARACTERS

Each line of input from paper tape must end with a particular sequence of control characters. These are punched by functional keys on the teletypewriter keyboard. The control characters used with paper tape are as follows:

<u>Character</u>	<u>Description</u>
④	Message terminator. This is the RETURN key on the model 33. It moves the print head to the beginning of the line and informs the system that this line of input is completed.
LF	Line feed. This is the LINE FEED key on the keyboard. It advances the paper roller one line. In normal mode, the system sends a signal that initiates this feed; in tape mode, this signal is not sent and is therefore required on tape.
CTRL/X-OFF	X-off. The appearance of this character on the tape during reading turns the tape reader off. This character is punched by holding down the CTRL key and pressing the X-OFF key.
RO(n)	Rubout. This is the RUB OUT key on the keyboard. It punches a full row of holes. This row is interpreted as null by the system, and hence this character is used for spacing and overpunching errors. The n parameter specifies the minimum number of times this character should be punched in sequence.
NUL	Null. This is the ASCII character represented by a feed hole only (blank tape). It may be used as a fill character.

INPUT LINES

The following line formats are used to enter programs, data, and commands. The left half of the page shows an example of the input; the right half of the page shows the control characters that immediately follow the last input character. Each line is terminated with three rubouts to provide separation from the next line. While it is possible that adequate separation is provided with less than three rubouts, this is the recommended number. (The system punches three rubouts at the end of each line when a tape is produced on-line.)†

PROGRAM LINE

100 PROGRAM TEST (INPUT,OUTPUT) ④ LF RO(3)

DATA LINE

? 12.44,18.31,29.08 ④ LF CTRL/X-OFF RO(3)

(The ? is supplied by the system. The remainder of the line comes from the tape. X-OFF turns the reader off to allow this data to be processed before the next line is read.)

COMMAND LINE

CATLIST,LO=F ④ LF CTRL/X-OFF RO(3)

(X-OFF turns the reader off to allow processing of the command to be completed before the next line is read.)

†Lines are blank-padded to an even number of characters.

PUNCHING A TAPE OFF-LINE

The following procedure is used to punch a tape when the teletypewriter terminal is not logged in.

1. Place the teletypewriter in local mode.
2. Turn the paper tape punch on by pressing the ON button located on the punch.
3. Prepare a tape leader of about 30 rubouts (3 inches) either by simultaneously pressing the RUB OUT and REPT keys or, if the terminal has the capability, by punching blank tape.
4. With a pencil, trace the arrow above the punch output onto the tape. This identifies the start of the tape for reading. (It is possible to put the tape in the reader backwards.)
5. Type in the input lines with their appropriate control characters.
6. Add a 3-inch trailer of rubouts or blank tape and tear the tape off.
7. Turn the teletypewriter off.

TAPE MODE

To read and process data and commands from tape, it is necessary that the tape reader be turned off, to allow time for processing, then be restarted. By entering tape mode, the user enables the system to synchronize tape input with its processing. This mode also inhibits the output of header and READY messages when the LIST command (or LNH) is used to punch a tape on-line (refer to Punching a Tape On-Line in this section). CR entered on an empty input line is ignored by the system when in tape mode.

The user enters tape mode by typing the TAPE command and pressing CR . The system acknowledges entry into this mode by advancing the printout paper one line; it does not print a READY message. If the reader switch is at AUTO, reading begins automatically; if the reader switch is in the STOP position, reading does not begin until manually initiated by momentarily moving the switch to START. All subsequent commands and data entries are acknowledged by the one-line advancement without the printout of a READY message.

The user exits from tape mode by typing the NORMAL command and pressing CR .

PUNCHING A TAPE ON-LINE

To punch a tape with information already contained in a system file, the following procedure is used.

1. If not logged in, log in.
2. If not in tape mode, type TAPE and press CR .
3. Type in LIST if the primary file is to be punched; type in LIST,F=lfn if a local file other than the primary file is to be punched. Do not press CR . (The LNH command is permissible.)
4. Turn the punch on.
5. Run a leader of rubouts or blank tape.
6. Press CR . The file is listed and punched simultaneously. The system adds the appropriate control characters at the end of each line.
7. Run a trailer of rubouts or blank tape.

CORRECTIONS

When punching a tape off-line, corrections can be made by backspacing over the incorrect punch (use the B.SP. button on the punch) and punching a rubout over the error (use the RUB OUT key on the keyboard). Then punch in the correct character.

It is also possible to make corrections by using the backspace key as in normal, on-line operation. However, the error and/or any overtyping appear in the printout. When line numbers are being used, an erroneous line can be retyped (and repunched) with the same line number. Although the erroneous and correct versions appear in printout, the system retains only the last occurrence of the specific numbered line during processing.

Example:

In this example, the input tape contains only program lines. Commands and input are typed in by the user before and after the tape is read in.

The following is the tape as punched. The left half of the page shows the printed copy produced as the tape is punched; the right half of the page shows the control characters entered at the end of each line but not printed. In this example, any character shown as RO (rubout) can be replaced by NUL (blank tape).

```
100 LET FACTOR=1
110 INPUT N
120 FOR I=1 TO N
130 LET FACTOR=FACTOR*I
140 PRINT FACTOR
150 NEXT I
160 END
```

```
RO(30) or NUL(30)
(CR) LF RO(3)
(CR) LF RO(30)
```

To demonstrate the use of this tape, it is assumed the user is logged in under the null subsystem. He positions this tape in the reader and sets the reader switch to AUTO. The following is the printout that results from the interaction of user and tape with the terminal. The right half of the page contains explanations of the action.

basic ←
READY.

By typing this command, the user enters the BASIC subsystem.

new,fact1 ←
READY.

The user types in this line to establish a new primary file with the name FACT1.

tape ←

```
100 LET FACTOR=1
110 INPUT N
120 FOR I=1 TO N
130 LET FACTOR=FACTOR*I
140 PRINT FACTOR
150 NEXT I
160 END
```

The user types TAPE. As soon as the carriage return takes effect, the tape reader goes on and the entire tape (seven lines) is read in. The reader runs through the trailer of rubouts and then turns itself off.

normal ←
READY.

rnh

```
ILLEGAL STATEMENT AT 100
ILLEGAL STATEMENT AT 130
ILLEGAL STATEMENT AT 140
```

The user types NORMAL to return the terminal to normal mode. He then types RNH and receives the diagnostics shown. BASIC variables cannot be more than two characters.

```

SRU      0.015 UNTS
RUN COMPLETE.
100 let f=1
130 let f=f*i } ← The user types in the corrective coding.
140 print f
rnh ← The user initiates another run with the
? 5
1
2
6
24
120
SRU      0.017 SECS.
RUN COMPLETE.
tape
lnh ←
100 LET F=1
110 INPUT N
120 FOR I=1 TO N
130 LET F=F*I
140 PRINT F
150 NEXT I
160 END

```

The user types in the corrective coding.

The user initiates another run with the RNH command. The system replies with ? He types in 5 and receives the desired printout.

lnh ← The user wants to punch a new tape containing this corrected program. He enters tape mode and then types LNH but does not press \textcircled{CR} . He turns on the punch, runs off some leader, and then enters \textcircled{CR} . As the system prints the listing, it also punches a tape with the same information and the control characters \textcircled{CR} , LF, and RO(3) at the end of each line. After this, the user punches a trailer of rubouts, tears off the new tape, and discards the original tape.

Example:

In this example, the input tape contains not only a program but also the commands to execute, modify, list, and save that program and the input data used by that program.

The following is the complete tape as punched. The left half of the page gives the printed copy produced as the tape is punched; the right half of the page shows the control characters entered at the end of each line but not printed. Any character shown as RO (rubout) may be replaced by NUL (blank tape).

```

FTNTS
NEW,DEMO
100 PROGRAM DEMO(INPUT,OUTPUT)
110 DIMENSION II(5)
120 READ 10, (II(J),J=1,5)
130 10 FORMAT(5I5)
140 PRINT 20, (II(J),J=1,5)
150 20 FORMAT(5I10)
160 END
RNH
111112222233334444455555
SAVE,DEMO=TAPE1
135 DO 1 I=1,5
137 II(I)=II(I)+4444
139 1 CONTINUE
RESEQ,100,5
LNH
RNH
111112222233334444455555
REPLACE,DEMO=TAPE1

```

RO(30) or NUL(30)			
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	RO(3)	
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	RO(3)	
(CR)	LF	RO(3)	
(CR)	LF	RO(3)	
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(3)
(CR)	LF	CTRL/X-OFF	RO(30)

To execute this tape, the user logs in, positions the tape in the reader, sets the reader switch to AUTO, types TAPE and, upon pressing **CR**, starts the reading of the tape. He then observes the remainder of the action without intervention.

The following is the printout of the execution of the tape. The right half of the page contains explanations of the action.

FTNTS ← After FTNTS is read, the reader is turned off. As soon as FTNTS is established as the current subsystem, the system turns the reader on.

NEW,DEMO ← 100 PROGRAM DEMO(INPUT,OUTPUT)
110 DIMENSION II(5)
120 READ 10, (II(J),J=1,5)
130 10 FORMAT(5I5)
140 PRINT 20, (II(J),J=1,5)
150 20 FORMAT(5I10)
160 END
RNH ← After the command to run the program is read, the reader is turned off.

? 1111122222333334444455555 ← After processing reaches the READ statement (line 120) and the system prints ?, the system turns the reader on, and the line of data is read. Then the reader is turned off to allow data processing.

Resulting printout:

11111	22222	33333	44444	55555
END.				
SRU	0.130 UNTS.			

RUN COMPLETE.

SAVE,DEMO=TAPE1 ← After the SAVE command is read, the reader is turned off. A copy of the program DEMO is made an indirect access permanent file with the name TAPE1. Then the reader is turned back on.

135 DO 1 I=1,5
 137 II(I)=II(I)+4444
 139 1 CONTINUE } ← These three lines of modification are read in from the tape without interruption.

RESEQ,100,5 ← After the RESEQ command is read, the reader is turned off. When resequencing is accomplished according to specifications, the reader is turned back on.

SRU 0.576 UNTS.

LNH ← After the LNH command is read from tape, the tape reader is turned off. Then the system lists the primary file which contains the above modifications in resequenced format. When the listing is completed, the reader is turned on again.

```

00100 PROGRAM DEMO(INPUT,OUTPUT)
00105 DIMENSION II(5)
00110 READ 10, (II(J),J=1,5)
00115 10 FORMAT(5I5)
00120 DO 1 I=1,5
00125 II(I)=II(I)+4444
00130 1 CONTINUE
00135 PRINT 20, (II(J),J=1,5)
00140 20 FORMAT(5I10)
00145 END
  
```

RNH ← After the RNH command is read in, the reader is turned off, and the modified program is executed. The system prints the ? and then turns the reader on. The line of data is read in from the tape and the reader turned off. The data is processed and results printed.

```

? 111112222233334444455555
      15555     26666     37777     48888     59999
END.
  
```

SRU 1.185 UNTS.
 RUN COMPLETE

REPLACE,DEMO=TAPE1 ← This last command is read in and the reader turned off. A copy of this revised version of DEMO replaces the old one that was made an indirect access permanent file under the name TAPE1. It is now the indirect access permanent file that is referenced by the name TAPE1.

The reader is turned back on. It runs through the trailer of rubouts and then turns itself off.

If, at this point, the user has no more tapes to run, he should exit tape mode by entering the NORMAL command.

BLOCK EDIT MODE TERMINAL USE

10

Block edit mode is designed for use with ASCII code terminals with hardware buffering and editing capabilities. Typically, these terminals allow the user to compose a message which is stored in the terminal buffer until the operator presses the SEND key, at which time it is transmitted from the buffer. Before the message is sent from the buffer, it can be edited or corrected by the operator. The data in the buffer can consist of one or more lines, separated by carriage return codes and terminated by an ASCII end-of-transmission (EOT) code. The EOT code is generated automatically by most terminals.

Because of the editing capabilities of the terminal, data is assumed to be correct as received. Therefore, in block edit mode the escape (ESC) and backspace (BS) characters are regarded as input data rather than as control characters.

The operator can send multiple line messages in text mode or as line numbered source data. Multiple lines of input to a running program or multiple command lines are not permitted; however, the last line of line-numbered source data can be a command.

The following are the special characters that are recognized and processed by the system when the terminal is in block edit mode.

<u>Character</u>	<u>Description</u>
EOT (CNTL D)	The EOT character signals the end of the input transmission. If this code is used to terminate an input line, the system generates a carriage return and a line feed at the terminal.
(CR)	The carriage return denotes the end of a line of data. Because block edit mode terminals automatically provide a line feed, no line feed is returned by the system.
ETX (CNTL C)	The ETX character is used to exit from text mode. It is sent as the first and only character after an EOT.

CHARACTER SETS

A

The standard character set used by NOS is the 64-character set shown in table A-1. As an option, a slightly different 63-character set can be selected by an installation as the default character set.

CONVERT COMMAND

The CONVERT command converts one character set to another. The command format is

CONVERT,*p₁,p₂,...,p_n*

p_i can be any of the following in any order.

<u><i>p_i</i></u>	<u>Description</u>
P=lfn ₁	Input on file lfn ₁ ; if omitted, the file named OLD is assumed.
N=lfn ₂	Output on file lfn ₂ ; if omitted, the file named NEW is assumed.
RS=n	Maximum record size in characters, where n is 1 to 500. If omitted, assumed maximum record size is 300 characters.
64	Converts 63-character set to 64-character set. If omitted, no 63- to 64-character set conversion takes place (TS must be specified if 64 is not).
TS=t	Converts old time-sharing record (61-character set) to new time-sharing record (63-character set).† t may be one of the following terminal types.

<u>t</u>	<u>Terminal Type</u>
TTY	ASCII code terminal with standard print
COR	Correspondence code terminal with standard print
CORAPL	Correspondence code terminal with APL print
MEMAPL	Memorex 1240 (ASCII code) terminal with APL print
BLKEDT	Block transmission (ASCII code) terminal with standard print

If t is omitted, it is assumed to be TTY. If TS is omitted, no time-sharing conversion takes place (64 must be specified if TS is not).

R Rewinds input and output files prior to processing. If omitted, no rewinding occurs.

† Old time-sharing character set refers to the 61-character set used by preceding systems (for example, early versions of KRONOS 2.1).

<u>P_i</u>	<u>Description</u>
RC=n	Converts n decimal records. If n is omitted, converts to EOF. If RC is omitted, converts one record.
NM	Used in conjunction with TS parameter and specifies that conversion is to normal mode; if omitted, conversion is to ASCII mode. Conversion has the indicated effect on the following characters.
<u>Character</u>	<u>Effect</u>
^ (circumflex)	If TS is specified, display code 70 (circumflex character) is converted to 76. If NM is omitted, conversion is to 7402 (ASCII mode).
: (colon)	If TS and 64 are specified, display code 63 (colon character) is converted to 00. If NM is omitted, conversion is to 7404 (ASCII mode).

The following conversions are possible as indicated by the legal choice of parameters.

<u>Type of Record</u>	<u>Legal Conversion</u>
63-character set, nontime-sharing record	64
Old time-sharing record	TS or 64 and TS
New normal time-sharing record (equivalent to batch character set)	64
New ASCII time-sharing record	None

CONVERT command error messages are listed in appendix B.

CENTRAL MEMORY CHARACTER EQUIVALENTS FOR DATA INPUT

The manner in which characters entered from the terminal are interpreted by the system depends on whether the user has specified that the characters belong to the full ASCII set (refer to the ASCII command in section 4). For example, if the user enters the following characters:

aAbBcCdDeE

to be mapped into the full ASCII set (128 characters recognized), the central memory equivalent (in display code) is

59	53	47	41	35	29	23	17	11	5	0	← Bit position (60-bit central memory word)
76	01	01	76	02	02	76	03	03	76		
04	04	76	05	05	55	00	00	00	00		

However, if the characters are mapped into the subset of the ASCII character set used during normal operation (64 characters recognized), the central memory equivalent (in display code) is

59	53	47	41	35	29	23	17	11	5	0
01	01	02	02	03	03	04	04	05	05	

Characters entered under control of the BINARY command are not translated into display code. It makes no difference if the ASCII command is in effect (128 characters recognized) or if the characters are entered during normal operation (64 characters recognized). However, the system weights each 8-bit character by 4000_8 and places a 0007_8 control byte at the beginning of the data. For example, if the BINARY command is in effect, and the user enters the following characters:

aAbBcCdDeE

the central memory equivalent is

59	47	35	23	11	0
0007	4341	4101	4342	4102	
4143	4303	4344	4104	4145	
4305	0000	0000	0000	0000	

The characters illustrated are entered from an ASCII code terminal, although a correspondence code terminal can also be used.

CENTRAL MEMORY CHARACTER EQUIVALENTS FOR DATA OUTPUT

The preceding discussion applies but with the following note.

NOTE

Data output is in either a 64/63- or 128-character set, depending on whether the terminal is in normal or ASCII mode, respectively. When the terminal is in normal mode, display codes 74 and 76 represent data rather than escape codes. When in ASCII mode, display code 74 or 76 is treated as the beginning of an escape sequence. Under the 64-character set in ASCII mode, display code 63 maps to %, while display code 7404 maps to :. Under the 63-character set in ASCII mode, display code 63 maps to :, while display code 7404 maps to %. Under the 63-character set in ASCII mode, display code 00 does not exist.

TABLE A-1. TIME-SHARING 64-CHARACTER SET

ASCII CODE TERMINAL†			CORRESPONDENCE CODE TERMINAL††			DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT	STANDARD PRINT		APL PRINT	
CHAR.	CODE (8-BIT OCTAL)	CHAR. (8-BIT OCTAL)	CHAR. (7-BIT OCTAL)	CODE (7-BIT OCTAL)	CHAR. (7-BIT OCTAL)	
:	072	:	276	:	153	00†††
A	101	A	341	A	171	01
B	102	B	342	B	166	02
C	303	C	143	C	172	03
D	104	D	344	D	052	04
E	305	E	145	E	112	05
F	306	F	146	F	163	06
G	107	G	347	G	043	07
H	110	H	350	H	046	10
I	311	I	151	I	031	11
J	312	J	152	J	103	12
K	113	K	353	K	032	13
L	314	L	154	L	106	14
M	115	M	355	M	141	15
N	116	N	356	N	122	16
O	317	O	157	O	105	17
P	120	P	360	P	013	20
Q	321	Q	161	Q	133	21
R	322	R	162	R	051	22
S	123	S	363	S	045	23
T	324	T	164	T	002	24
U	125	U	365	U	062	25
V	126	V	366	V	061	26
W	327	W	167	W	165	27
X	330	X	170	X	142	30
Y	131	Y	371	Y	147	31
Z	132	Z	372	Z	124	32
0	060	0	060	0	144	33
1	261	I	261	I	040	34
2	262	2	262	2	020	35
3	063	3	063	3	160	36
4	264	4	264	4	004	37
5	065	5	065	5	010	40
6	066	6	066	6	130	41
7	267	7	267	7	150	42
8	270	8	270	8	070	43
9	071	9	071	9	064	44
+	053	+	055	+	023	45
-	055	-	275	-	067	46
*	252	*	120	*	070	47
/	257	/	257	/	007	50
(050	(053	(064	51
)	251)	252)	144	52
\$	044	\$	374	\$	004	53
=	275	=	245	=	023	54

† THE OCTAL CODES LISTED FOR ASCII CODE TERMINALS ARE SHOWN WITH EVEN PARITY (NORMAL).

†† THE OCTAL CODES LISTED FOR CORRESPONDENCE CODE TERMINALS ARE SHOWN WITH ODD PARITY (NORMAL).

††† USE OF THE COLON IN PROGRAM AND DATA FILES CAN CAUSE PROBLEMS. THIS IS PARTICULARLY TRUE WHEN IT IS USED IN PRINT AND FORMAT STATEMENTS. FOR 63-CHARACTER SET, THE DISPLAY CODE 00 LINE IS REPLACED BY NULL CHARACTERS.

TABLE A-1. TIME-SHARING 64-CHARACTER SET (Contd)

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				DISPLAY CODE (6/12-BIT OCTAL)	
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT			
CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)		
(SPACE)	240	(SPACE)	240	(SPACE)	100	(SPACE)	100	55	
,	254	,	254	,	073	,	073	56	
.	056	.	056	.	121	.	121	57	
#	243	"	041	#	160	"	040	60	
[333	[273	/4	001	[153	61	
]	335]	072	/2	001]	111	62	
%	245	%	176	%	010	%	023	63††	
"	042	#	050	"	111	#	070	64	
-	137†	-	306	-	067	-	163	65	
!	041	v	251	\$	130	v	064	66	
8	246	^	137	8	150	^	144	67	
·	047	·	113	·	111	·	032	70	
?	077	?	321	?	007	?	133	71	
<	074	<	243	NULL	---	<	160	72	
>	276	>	047	NULL	---	>	150	73	
@	300	≤	044	@	020	≤	004	74	
\	134	\	077	NULL	---	\	007	75	
^	336	-	042	NULL	---	-	020	76	
;	273	;	074	;	153	;	073	77	
NULL	---	NULL	---	NULL	---	NULL	---	7600	
a	341	¤	101	¤	171	¤	171	7601	
b	342	↳	102	b	166	↳	166	7602	
c	143	□	303	c	172	□	172	7603	
d	344	l	104	d	052	l	052	7604	
e	145	€	305	e	112	€	112	7605	
f	146	X	336	f	163	X	023	7606	
g	347	▽	107	g	043	▽	043	7607	
h	350	△	110	h	046	△	046	7610	
i	151	↶	311	i	031	↶	031	7611	
j	152	◦	312	j	103	◦	103	7612	
k	353	⊣	173	k	032	NULL	---	7613	
l	154	□	314	l	106	□	106	7614	
m	355	-	115	m	141	-	141	7615	
n	356	T	116	n	122	T	122	7616	
o	157	0	317	o	105	0	105	7617	
p	360	-	300	p	013	-	001	7620	
q	161	-	140	q	133	-	001	7621	
r	162	P	322	r	051	P	051	7622	
s	363	Γ	123	s	045	Γ	045	7623	
t	164	~	324	t	002	~	002	7624	
u	365	↓	125	u	062	↓	062	7625	
v	366	U	126	v	061	U	061	7626	
w	167	ω	327	w	165	ω	165	7627	
x	170	□	330	x	142	□	142	7630	
y	371	↑	131	y	147	↑	147	7631	

† ON TTY MODELS HAVING NO UNDERLINE, THE BACKARROW (←) TAKES ITS PLACE.

†† FOR 63-CHARACTER SET, THE DISPLAY CODE 63 LINE (EXCEPT THE 63 DISPLAY CODE) IS REPLACED BY THE DISPLAY CODE 00 LINE OF THE 64-CHARACTER SET.

TABLE A-1. TIME-SHARING 64-CHARACTER SET (Contd)

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				DISPLAY CODE (6/12-BIT OCTAL)	
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT			
CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)		
z	372	c	132	z	124	c	124	7632	
{	173	{	335	NULL	—	NULL	—	7633	
:	374	≥	246	±	040	≥	130	7634	
}	175	}	175	NULL	—	NULL	—	7635	
~	176	†	333	NULL	—	NULL	—	7636	
DEL	377	DEL	377	NULL	—	NULL	—	7637	
NUL	000	NUL	000	NUL	075	NUL	075	7640	
SOH	201	SOH	201	SOA	166	SOA	166	7641	
STX	202	STX	202	EOA	064	EOA	064	7642	
ETX	003	ETX	003	NULL	—	NULL	—	7643	
EOT	204	EOT	204	EOT	174	EOT	174	7644	
ENQ	005	ENQ	005	NULL	—	NULL	—	7645	
ACK	006	ACK	006	ACK	067	NULL	—	7646	
BELL	207	BELL	207	NULL	—	NULL	—	7647	
BS	210	BS	210	BS	135	BS	135	7650	
HT	011	HT	011	HT	057	HT	057	7651	
LF	012	LF	012	LF	156	LF	156	7652	
VT	213	VT	213	NULL	—	NULL	—	7653	
FF	014	FF	014	NULL	—	NULL	—	7654	
CR	215	CR	215	CR	155	CR	155	7655	
SO	216	SO	216	UCS	034	UCS	034	7656	
SI	017	SI	017	LCS	037	LCS	037	7657	
DLE	220	DLE	220	NULL	—	NULL	—	7660	
DC1	021	DC1	021	NULL	—	NULL	—	7661	
DC2	022	DC2	022	NULL	—	NULL	—	7662	
DC3	223	DC3	223	NULL	—	NULL	—	7663	
DC4	024	DC4	024	STO	054	STO	054	7664	
NAK	225	NAK	225	NAK	001	NAK	001	7665	
SYN	226	SYN	226	IL	075	IL	075	7666	
ETB	027	ETB	027	EOB	136	EOB	136	7667	
CAN	030	CAN	030	DEL	177	DEL	177	7670	
EM	231	EM	231	NULL	—	NULL	—	7671	
SUB	232	SUB	232	NULL	—	NULL	—	7672	
ESC	033	ESC	033	PF	076	PF	076	7673	
FS	234	FS	234	NULL	—	NULL	—	7674	
GS	035	GS	035	NULL	—	NULL	—	7675	
RS	036	RS	036	NULL	—	NULL	—	7676	
US	237	US	237	NULL	—	NULL	—	7677	
NULL	—	NULL	—	NULL	—	NULL	—	7400	
@	300	≤	044	@	020	≤	004	7401	
^	336	—	042	NULL	—	—	020	7402	
NULL	—	NULL	—	CNL	001	CNL	001	7403	
:	072	:	276	:	153	:	121	7404†	
NULL	—	NULL	—	NULL	—	NULL	—	7405	
NULL	—	NULL	—	NULL	—	NULL	—	7406	
NULL	—	NULL	—	NULL	—	NULL	—	7407	

† FOR 63-CHARACTER SET, THE DISPLAY CODE 7404 LINE (EXCEPT THE 7404 DISPLAY CODE) IS REPLACED BY THE DISPLAY CODE 63 LINE OF THE 64-CHARACTER SET.

3AE5B

TERMINAL MESSAGES

B

This appendix contains an alphabetical listing of messages of importance to the time-sharing user. (A list of dayfile messages can be found in appendix B of the NOS Reference Manual, volume 1.) Each message is followed by an explanation of the message and/or the circumstances causing it to be issued, the recommended action, and the routine which issued the message.

Lowercase letters are used within a message to identify variable fields. All messages beginning with lowercase (variable) fields follow those beginning with A through Z. These messages are then alphabetized according to the first nonvariable field.

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
ARGUMENT ERROR.	A control statement is syntactically incorrect. Refer to the appropriate control statement or command for further information.	Recheck parameters.	CONVERT, COPY, COPYBR, COPYBF, COPYEI, COPYX, CPMEM, L072, RESEX, TCOPY, UPMOD
BUFFER ARG. ERROR.	When the system processes tape management statements, it issues this message if both ring enforcement options (PO=R and PO=W) or more than one EOT option (PO=I, PO=P, and PO=S) is specified. Also, specification of duplicate parameters (more than one occurrence of a keyword) or multiple equivalent parameters (such as MT/NT, CB/CK, FI/L, R/W, and so forth) is not allowed on tape assignment control statements.		
CATALOG OVERFLOW - FILES, AT addr.	The address parameter on DMPECS, DMDECS, LBC, LOC, PBC, or WBR must be numeric.	Verify that operation does not reference address beyond end of buffer or job's field length.	TCS
CATALOG OVERFLOW - SIZE AT addr.	CM address in call is not less than the field length minus the word count; buffer extends past the job's field length.		
CCL100-SEPARATOR FOLLOWING VERB MUST BE COMMA OR LEFT PARENTHESIS	The number of files in the user's catalog exceeds his limit.	One or more permanent files must be purged in order to save or define additional files.	PPM
CCL101-LAST NON-BLANK CHARACTER MUST BE SEPARATOR	The cumulative size of the indirect access files in the user's catalog exceeds his limit.	One or more indirect access files must be purged or shortened to allow additional permanent file space.	PPM
	Fatal user error. Separator following verb in a CCL statement must be a comma or a left parenthesis.	Change separator following verb to a comma or a left parenthesis.	CCL
	Fatal user error. Last character string of card or line was not followed by a separator or a terminator.	To terminate statement, make last nonblank character a period or right parenthesis. To continue statement on next card or line, make last nonblank character a valid separator.	CCL

MESSAGE -----	SIGNIFICANCE -----	ACTION -----	ROUTINE -----
CCL103-STATEMENT INCOMPLETE	Fatal user error. A terminator was detected immediately following a verb.	Check statement format (refer to section 4 of NOS Reference Manual, volume 1) and rewrite statement, using a comma or left parenthesis after verb.	CCL
CCL200-PROCEDURE NESTING LEVEL xx EXCEEDED	Fatal user error. Current procedure call forced procedure nesting to exceed limit of xx, which is defined by installation.	Reposition procedure call statements so limit xx is not exceeded.	CCL
CCL201-PROCEDURE NAME MORE THAN 7 CHARACTERS	Fatal user error. Procedure name cannot be greater than 7 characters.	Rename procedure.	CCL
CCL203-PROCEDURE FILE NAME NOT SPECIFIED OR INVALID	Fatal user error with following causes. - File name specified on BEGIN statement was greater than 40 characters. - pfile parameter on BEGIN statement was null, indicating default file name. CCL was installed with default file flag turned off; no default file name is allowed.	- Specify a file name with seven or fewer characters. - Specify a file name.	CCL
CCL204-MULTIPLE EQUIVALENCE SPECIFICATIONS FOR xx	Informative message. A keyword has been specified more than once on a procedure call statement. The last specification is used.	If a preceding specification is desired, remove succeeding specifications. Otherwise, no action is required.	CCL
CCL205-FORMAL PARAMETER LIST DOES NOT INCLUDE -x	Fatal user error. While in equivalence mode, CCL discovered a keyword x on the procedure call statement that was not specified on header statement.	Remove keyword x from procedure call statement.	CCL
CCL206-SYMBOLIC SPECIFICATION INVALID xxxx+	Fatal user error. A parameter, xxxx, on procedure call statement is followed by a plus sign. Plus sign indicates that CCL is to convert value to display code, and is valid only when preceded by a symbolic name and followed by a D, B, or a null field. (Example: R1+B is a legal value, but 37+R1 is not.)	If plus sign is part of a character string, \$-delimit character string. If plus sign should convert a symbolic name to display code, replace xxxx with a valid symbolic name. Refer to section 4 of NOS Reference Manual, volume 1 for a list of valid symbolic names.	CCL

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
CCL207-PROCEDURE NAMED BEGIN IS INVALID	Fatal user error. A procedure must not be named BEGIN.	Select another procedure name.	CCL
CCL211-SPECIFICATION EXCEEDS xx CHARACTERS	Fatal user error. Value on a procedure call statement or default value on a procedure header statement is greater than xx characters. xx is defined by installation.	Specify a value with xx or fewer characters.	CCL
CCL212-SEPARATOR INVALID strng s	Fatal user error. s is illegal separator and strng is character string that precedes s. Any of following conditions produces this error. <ul style="list-style-type: none">- On a procedure call statement, the separator preceding a formal keyword or a positionally specified value is not a comma.- Invalid separator is part of a character string. If a valid separator immediately precedes illegal separator s, the string is null.- #DATA or #FILE has been specified on the procedure call statement.	<ul style="list-style-type: none">- Change separator s to a comma.- \$-delimit character string.- Remove #DATA or #FILE from procedure call statement and specify on procedure header statement.	CCL
CCL230-PROCEDURE FILE NOT FOUND	Fatal user error. The file named on the BEGIN statement was not assigned to the job and the installation had chosen to inhibit automatic retrieval of a permanent file with that name.	Check file name for errors and correct. If name is correct, retrieve file prior to BEGIN statement.	CCL
CCL231-PROCEDURE NOT FOUND	Fatal user error. Local or permanent file indicated on BEGIN statement was found, but CCL could not find procedure on the file.	Check procedure name for errors and correct. If name is correct, check file for procedure.	CCL
CCL235-FORMAL PARAMETER GT xx CHARACTERS	Fatal user error. Number of characters in a keyword on header statement exceeds xx, as defined by installation.	Define a keyword with xx or fewer characters.	CCL
CCL237-SEPARATOR FOLLOWING SECOND DEFAULT IS */*	Fatal user error. In form keyword=default1/ default2/ on procedure header statement, second / is illegal.	If / is part of default2, \$-delimit character string. If not, either remove it or replace it with a comma or period.	CCL
CCL238-FORMAL PARAMETER LIMIT xx EXCEEDED	Fatal user error. Number of parameters on procedure header statement has exceeded xx, as defined by installation.	Remove parameters from procedure header statement until xx or fewer parameters remain.	CCL

MESSAGE -----	SIGNIFICANCE -----	ACTION -----	ROUTINE -----
CCL239-PROCEDURE HEADER NOT TERMINATED	Fatal user error. Procedure header statement was not terminated by a period, and no control statements were found after header statement.	Terminate header statement with a period, and add at least one control statement to procedure.	CCL
CCL252-PROCEDURE CONTAINS NO CONTROL STATEMENTS	Informative message. A procedure should contain at least one control statement.	None.	CCL
CCL263 EXTERNAL ABORT DURING BEGIN.	An external abort during BEGIN processing occurred. Check user's dayfile to find where error occurred.	Correct BEGIN or PROC statement as indicated by dayfile.	CCL
CCL263 EXTERNAL ABORT DURING ENDW.	An external abort occurred during CCL search for the WHILE control statement. The abort is due to a DROP or RERUN command typed by the operator, or an exceeded time limit, mass storage limit, or ECS limit.	Check validation limits. Modify job to fit within limits or ask site to increase the limits.	CCL
CHARGE ABORTED.	Dayfile message indicating that a central site operator action caused the CHARGE operation to abnormally terminate.	Reenter CHARGE statement.	CHARGE
CHARGE FILE BUSY.	Dayfile message indicating that the file which the system uses to validate charge numbers and project numbers is busy.	Reenter CHARGE statement.	CHARGE
CHARGE ILLEGAL AT THIS HOUR.	Dayfile message indicating that the specified project number cannot be used at this time of day.	Retry during the time the project number is valid.	CHARGE
CHARGE NUMBER EXPIRED.	Dayfile and output file message indicating the charge number expiration date has occurred.	None.	CHARGE
CHARGE NUMBER REQUIRED.	Charge number and project number are required to complete login sequence.	Enter CHARGE command with appropriate charge and project numbers.	CHARGE
COMMAND NOT UNIQUE.	The characters entered do not uniquely specify a command.	Ensure accuracy of entry. Enter a sufficient number of characters to specify a unique command.	IAFEX, TELEX
COMMAND TOO LONG.	A command longer than 80 characters was entered.	Reenter using a shorter version of the command.	IAFEX, TELEX
CONFLICTING PARAMETERS.	Input queue type entered more than once.	Reenter with correct parameters.	SUBMIT

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
CONTROL CARD ERROR.	An illegal or invalid parameter was specified on the control statement.	Correct control statement and retry.	EDIT, LISTLB
CONTROL STATEMENT LIMIT.	The user has entered too many control statements.	Reduce the number of control statements to a value within the specified limit. Refer to LIMITS command.	TCS
CONVERSION NOT FOUND.	The specified TS conversion table was not found.	Refer to description of CONVERT command for valid TS options.	CONVERT
CONVERSION NOT SPECIFIED.	Neither the TS nor 64 parameter was specified on the CONVERT control statement.	Correct error and retry.	CONVERT
CUMULATIVE LIMIT EXCEEDED.	Dayfile and output file message indicating that one of the installation-defined resource usage accumulators for this project exceeded the maximum allowed. The system does not update these accumulators in PROFILA. Each installation must provide this capability if desired.	None.	CHARGE
CUMULATIVE SRU LIMIT EXCEEDED.	Dayfile and output file message indicating that accumulated SRUs have exceeded the maximum allowed.	None.	CHARGE
DATA BASE ERROR.	Dayfile message indicating that the system has detected an error in its validation file.	Contact installation personnel.	CHARGE, MODVAL
DAYFILE - ARGUMENT ERROR.	Keyword specified is not recognizable or control statement is not properly formatted.	Check keyword and control statement formats.	DAYFILE
DAYFILE - BUFFER TOO SMALL.	The buffer DAYFILE uses to hold the central memory dayfile is not large enough.	Inform the site analyst that either the internal DAYFILE buffer must be made larger or the site must use smaller dayfile buffers for user dayfile message processing.	DAYFILE
DAYFILE - DATA LOST.	A data read error occurred while processing an active file. Processing continues with the next readable message. Lost data is not recoverable.	Inform site analyst.	DAYFILE

MESSAGE -----	SIGNIFICANCE -----	ACTION -----	ROUTINE -----
DAYFILE - FILE/CM BUFFER BOUNDARY ERROR.	The disk resident portion of the dayfile cannot be joined correctly with the CM buffer portion.	Inform site analyst.	DAYFILE
DAYFILE - FILE TOO LONG.	The active dayfile being processed is longer than the system reservation status indicates.	Inform site analyst.	DAYFILE
DAYFILE - FILE TOO SHORT.	The active dayfile being processed is shorter than the system reservation status indicates.	Inform site analyst.	DAYFILE
DAYFILE - FR INVALID FOR THIS OPTION.	The FR=string parameter is not allowed with the OP=I parameter.	Omit the FR=string parameter and retry the command.	DAYFILE
DAYFILE - ILLEGAL PAGE SIZE FORMAT.	The page size option is non-numeric.	Retry with a valid option.	DAYFILE
DAYFILE - ILLEGAL PRINT DENSITY.	The print density option is not 3,4,6, or 8.	Retry with a valid option.	DAYFILE
DAYFILE - ILLEGAL PRINT DENSITY FORMAT.	The print density option is non-numeric. Print density must be 3,4,6, or 8.	Retry with a valid option.	DAYFILE
DAYFILE - JOB FIELD DISALLOWED ON USER DAYFILE.	Options specified are not allowed for user dayfile processing.	Retry with a correct OP=op parameter.	DAYFILE
DAYFILE - READ ERROR ON SEARCH FILE.	A read error occurred during the incremental dump and search option processing.	Retry the command.	DAYFILE
DAYFILE - RESERVED FILE NAME.	The file name specified for the L=lfn or I=infile parameter of the DAYFILE command is a reserved name.	Retry using a nonreserved name.	DAYFILE
DAYFILE - UNKNOWN *OP* FIELD.	The option specified is not valid.	Retry using a valid option.	DAYFILE
-DEBUG-GARBAGE IN ZZZZZDS FILE.	File ZZZZZDS contains information other than the suspended job of the current session. Retrieval of debug information from session is impossible.	Return file ZZZZZDS and restart debug session.	DEBUG
DEBUG-ILLEGAL PARAMETER-p.	User specified an illegal parameter on DEBUG command. The three parameters allowed are ON, OFF, or RESUME.	Reenter DEBUG command with correct parameter.	DEBUG

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
DEBUG—ONLY ONE PARAMETER ALLOWED.	User entered more than one parameter on the DEBUG command.	Reenter DEBUG command with correct parameter.	DEBUG
-DEBUG-ZZZZZDS FILE NOT FOUND.	User did not suspend the debug session during the current terminal session. Refer to the SUSPEND command (CYBER Interactive Debug Version 1 Reference Manual).	Restart debug session.	DEBUG
DEVICE UNAVAILABLE, AT addr.	Access to the permanent file device requested is not possible. User may have attempted to access files on a device not present in the alternate system.	Ask the operator to make the device available.	PFM
DIRECT ACCESS DEVICE ERROR, AT addr.	The file specified already exists on a device other than the device requested or an illegal device type was specified. The device on which the file resides may not contain direct access files because of one of the following reasons. - The device is not specified as a direct access device in the catalog descriptor table. - The device is not specified as ON and initialized in the catalog descriptor table. - The device is a dedicated indirect access permanent file device. If on an alternate system, the user's master device may not have been transferred to that system.	Specify correct device type.	PFM
EMPTY CATALOG.	No entries are present in the catalog.	None.	CATLIST, NDA
EMPTY SORT INPUT FILE.	Dayfile message indicating that the file specified on the SORT control statement contains no data.	Correct and rerun.	SORT
EOF ENCOUNTERED BEFORE TERMINATION.	An end-of-file was encountered on a CONVERT input file before the specified record count was reached.	Ensure accuracy of input file.	CONVERT
EOI ENCOUNTERED BEFORE TERMINATION.	An end-of-information was encountered on a CONVERT input file before the specified record count was reached.	Ensure accuracy of input file.	CONVERT
ERROR FLAG TERMINATION, FILE lfn AT addr.	The job was aborted with an input/output request in progress. The operation/request is not complete.	None.	CIO

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
ERROR IN ARGUMENT.	<p>One of the following.</p> <ul style="list-style-type: none"> - The pfn is blank or fn = lfn. - The user specified no arguments or a blank argument. - The user specified too many files. - The user entered an illegal parameter. 	Reenter the command or control statement with correct parameters.	PFILES
ERROR IN BACKUP REQUIREMENT.	The option specified on the BR parameter on the DEFINE, CHANGE, or SAVE statement is incorrect.	Check the BR parameter description and correct the statement.	PFILES
ERROR IN FILE ARGUMENTS.	The system did not recognize one or more parameters.	Compare the parameters specified with the command or control statement description.	MFILES
ERROR IN FILE CATEGORY.	The user specified an illegal file category.	Refer to description of file categories for valid entry.	PFILES, PURGALL
ERROR IN LIMITS ARGUMENT.	Dayfile message indicating that parameters were included on the LIMITS statement.	Enter LIMITS. without additional parameters.	MODVAL
ERROR IN MODE.	Permanent file mode specified was not recognized.	Refer to description of permanent file modes for valid entry.	PFILES
ERROR IN PASSWOR ARGUMENTS.	PASSWOR control statement or command parameters are incorrect.	Correct control statement or command and reenter.	PASSWOR, MODVAL
ERROR IN PREFERRED RESIDENCE.	The option specified on the PR parameter on the DEFINE or CHANGE statement is incorrect.	Check the PR parameter description and correct the statement.	PFILES
ERROR IN SUBSYSTEM.	The option specified on the SS parameter on the CHANGE or SAVE statement is incorrect.	Check the SS parameter description and correct the statement.	PFILES
EXCHANGE PACKAGE/MEMORY DUMP ON FILE ZZZDUMP.	The exchange package and memory dump is written on local file ZZZDUMP because the job is of time-sharing origin and file OUTPUT is assigned to a terminal.	To examine the exchange package and dump, list file ZZZDUMP.	CPMEM
EXECUTE ONLY FILE.	The permanent file accessed can only be executed and cannot be read, written, or listed.	Ensure accuracy of entry.	IAFEX, TELEX

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
FILE NAME CONFLICT.	Either the user tried to process two files having the same name or he specified a reserved file name.	Correct the control statement so all files have a unique name.	CONVERT, FCOPY, TCOPY, COPY, VERIFY, LIBGEN, UPMOD
FILE NAME ERROR.	File name contains illegal characters or contains more than seven characters.	Ensure that legal file name is specified.	LFM, LIBGEN, LISTLB
FILE NAME ERROR, AT addr.	File name contains illegal characters.	Verify that file name contains only valid characters.	PFM
FILE NAME MISSING.	A file name must be specified before a listing containing the permit information can be obtained with a CATLIST command.	Specify file name with FN parameter.	CATLIST
FILE NOT SORTED.	Primary file has one or more lines without line numbers.	Enter PACK or NOSORT command or use SORT command to obtain more detailed diagnostics.	IAFEX, TELEX
FILE TOO LONG, AT addr.	The local file specified for a SAVE, REPLACE, or APPEND command exceeds the length allowed or the direct access file specified for an ATTACH operation in WRITE, MODIFY, or APPEND mode exceeds the direct access file length limit for which the user is validated.	Reduce length of file or save as a direct access file.	PFM
FILE TOO LONG, FILE lfn AT addr.	The length of the direct access permanent file currently being written exceeds the direct access file length limit for which the user is validated.	Reduce the length of the file or ask site personnel for a larger direct access file length limit.	CIO
FILE TOO LONG TO SORT.	Primary file is too long to be sorted automatically.	Enter SORT command to obtain more detailed diagnostics.	IAFEX, TELEX
FL TOO SHORT FOR PROGRAM.	The user's field length is too short for the program.	Rerun the job with larger field length specification.	TCS, IAJ
FR INVALID FOR THIS OPTION.	The user entered a FR=string parameter with the OP=I parameter (incremental dump).	Check the DAYFILE statement description and correct the statement.	DAYFILE

MESSAGE -----	SIGNIFICANCE -----	ACTION -----	ROUTINE -----
FR NOT FOUND.	The specified string was not in the dayfile. A time-sharing user will get a full dayfile dump. A dayfile dump is not produced for a batch user.	Retry with a corrected FR=string.	DAYFILE
ILLEGAL CHARGE.	Dayfile and output file message indicating one of the following. - The charge or project number does not exist. - The project number is not available to a user with this user number. - The charge or project number exists but is inactive.	Check to see that charge and project numbers are correct and reenter.	CHARGE
ILLEGAL COMMAND.	The user entered an illegal command or a command he was not validated to use.	Correct command and reenter.	IAFEX, TELEX
ILLEGAL CONTROL CARD.	One of the following has occurred. - The control statement could not be identified. - The USER control statement does not have a user number specified. - An invalid parameter was specified or no terminator was detected. - The user included too many parameters on the program call statement or the program was not present. - The user submitted a control statement that he was not validated to use (for example, the use of PASSWOR by user not validated to change password). - The user submitted a control statement that is illegal for a particular job type or file type (for example, the use of a FAMILY statement in a nonsystem origin job).	Ensure accuracy and/or suitability of control statement.	TCS, CHARGE, CONFIG, MODVAL, RESEX, EXU, PASSWOR, 026
ILLEGAL DEVICE REQUEST.	The auxiliary device specified by the PN (pack name) parameter does not match the device type specified by the R (residency) parameter, or the number of units specified by the R parameter does not agree with the actual number of units available for that device type.	Ensure that devices specified by PN and R parameters are of same type, or number of units specified matches number of units available.	PFM
ILLEGAL LIST OPTION.	The list option specified in a CATLIST command is illegal.	Refer to description of CATLIST command for valid list options.	CATLIST

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
ILLEGAL LOGIN.	The validation file is not present for the specified family.	Ensure accuracy of response to FAMILY: request. Inform site analyst if problem persists.	TELEX
ILLEGAL PARAMETER.	One of the following. <ul style="list-style-type: none"> - Parameter value is outside legal bounds. - The user specified a parameter that cannot be included on the command or control statement. - Command/control statement is invalid. 	Ensure accuracy of command/control statement.	IAFEX, LISTLB, CATLIST, TELEX
ILLEGAL PASSWORD.	One of the following. <ul style="list-style-type: none"> - The password entered is greater than seven characters or contains an invalid character. - In the PASSWOR command either an incorrect old password was specified or the new password was unacceptable. - In the MODVAL control statement (for a create or update run) the password for a new user contained fewer characters than the minimum length required by the site. If entered from a K display, the line of input is ignored; otherwise, that particular user number is disregarded. 	Correct error and retry.	PASSWOR, MODVAL, PFILES
ILLEGAL QUEUE SPECIFIED.	Queue type specified is illegal.	Retry with valid queue type.	SUBMIT
ILLEGAL RUBOUT COUNT.	The count specified in the ROUT command was illegal.	Specify rubout count as described in section 4.	TELEX
ILLEGAL TERMINAL.	Four unsuccessful attempts at login were made after which the terminal was disconnected from the system.	Obtain accurate login information before attempting to log in. Inform site analyst if problem persists.	TELEX
ILLEGAL USER ACCESS.	The user tried to perform an operation for which he is not validated. Possible causes include attempts to <ul style="list-style-type: none"> - run a system origin job from nonsystem origin - access a restricted subsystem without proper validation - enter an invalid SRU value - use the V carriage control character without validation 	Ensure accuracy of control statement or determine proper validation requirements via LIMITS statement.	LFM, MSI, NETVAL, QFSP, RESEX, IMA

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
ILLEGAL USER ACCESS, AT addr.	The user is not validated to create direct access or indirect access files or to access auxiliary devices.	Contact site personnel concerning validations.	PFM
ILLEGAL USER ACCESS - CONTACT SITE OPR.	The security count for the user number specified has been decremented to zero. Therefore, the user is denied all access to the operating system until the operator resets the security count.	Contact site personnel to reestablish access.	CPM, NETVAL, IAJ, ILS
IMPROPER LOGIN, TRY AGAIN.	The family name, user number, or password is not acceptable as entered.	Ensure accuracy of entries and/or accuracy of information.	NETVAL, TELEX
INITIAL MESSAGE LIMIT.	The number of messages the job entered in the dayfile exceeded user's limit. Eight additional messages are allowed for error processing.	Restructure job to reduce dayfile messages.	IMA
INVALID ENTRY, TRY AGAIN	The user entered incorrect characters.	Enter correct characters.	IAFEX, TELEX
INVALID TERMINAL NUMBER.	The terminal number entered by the user during an attempted recovery contained a format error or size error.	Retry using corrected terminal number.	IAFEX, TELEX
JOB ACTIVE.	The system is unable to process the command because the previous command has not completed.	Retry when current operation is complete.	IAFEX, TELEX
JOB STEP EXCEEDS ACCOUNT BLOCK.	The user tried to set his job step limit to a value greater than his account block limit or tried to set his account block limit to a value less than his job step limit.	Check values on SETJSL and SETASL statements.	CPM
LINE NUMBER LIMIT EXCEEDED.	The line number encountered or required during a resequencing (RESEQ) operation exceeded 99999.	Examine program and correct line number in error.	RESEQ
LOCAL FILE LIMIT.	The user has too many local files.	Return one or more local files and retry.	LFM, QFM, IRO
MESSAGE LIMIT.	The number of messages the job issued has exceeded the limit for which the user is validated. Message functions issued by compilers or applications programs that run at the user's job control point are also counted as user dayfile messages and thus are subject to the user's validated	Split job into two or more jobs and retry.	IAJ, IMA

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
MODVAL ABORTED.	dayfile message limit. The terminal user was prevented from changing his password because the installation was updating the validation file or another user was modifying his password.	Retry operation at a later time.	MODVAL
NEXT VSN est,vsn	Tape volume vsn is mounted on tape unit est and accepted as the next volume in the file being processed.	None.	IMT
NO CONNECT TIME AVAILABLE.	Dayfile message indicating that the user has accumulated the maximum connect time allowed for the specified project number.	Contact installation personnel in order to increase maximum connect time allowed.	CHARGE
NO CPU TIME AVAILABLE.	Dayfile message indicating that the user has accumulated the maximum CPU time allowed for the specified project number.	Contact installation personnel in order to increase maximum CPU time allowed.	CHARGE
NO FILE FOUND - filenam.	The file specified in the READ directive of a submit file could not be found.	Ensure that the specified file exists.	SUBMIT
NO LINE NUMBER ON SORT FILE.	A line on the input file to a SORT request is missing a line number or a line exceeded the 150-character limit.	Check the format of the input file.	SORT
NO PRIMARY FILE.	The command entered requires a primary file.	Establish a primary file with NEW, OLD, or PRIMARY command and reenter original command.	IAFEX, TELEX
NO READ FILE FOUND - filenam.	The file specified on the /READ directive cannot be found.	Ensure that file name specified is correct and that the file is a local file or a permanent file.	SUBMIT
NO SOURCE FILE SPECIFIED.	No file name was specified on the control statement call.	Specify a file name on the SUBMIT control statement.	SUBMIT
NOTICE*** DATA READ ERROR - n WORDS LOST.	Read error caused loss of n words in the dayfile.	None.	DAYFILE
NOTICE*** FILE TOO LONG - n WORDS.	The dayfile is n words longer than the system file manager reported to DAYFILE.	Inform site analyst.	DAYFILE
NOTICE*** FILE TOO SHORT - n WORDS LOST.	The dayfile is n words shorter than the system file manager reported to DAYFILE.	Inform site analyst.	DAYFILE

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
PASSWORD TOO SHORT.	In the PASSWOR command the new password specified contains fewer characters than the minimum required.	Use a longer password.	PASSWOR, MODVAL
PF STAGING DISABLED, AT addr.	The specified direct access file resides on the Mass Storage Facility (MSF) and the site has temporarily disabled all MSF file staging.	Ask site operator when MSF file staging will resume and retry the job at that time.	PFM
PF UTILITY ACTIVE, AT addr.	Because a permanent file utility is currently active, the operation was not attempted; the user should retry the operation.	Wait until PF utility is not active and retry.	PFM
PFM ILLEGAL REQUEST, AT addr.	One of the following. - Illegal command code passed to PFM - Illegal permit mode or catalog type specified - CATLIST request has permit specified without a file name - PERMIT command or macro attempted on a public file	Verify that PFM request is valid.	PFM
PROCEDURE FILE EMPTY.	Procedure file specified contains no data.	Verify that procedure file contains data and retry.	CONTROL
PROJECT NUMBER EXPIRED.	Dayfile and output file message indicating that the project number expiration date has occurred.	None.	CHARGE
PRU LIMIT, AT addr.	The job's mass storage PRU limit was exceeded during preparation of a local copy of an indirect access file.	Return one or more local files and retry.	PFM
PRUS REQUESTED UNAVAILABLE.	No device currently has available the amount of space requested by the S parameter.	If possible, reduce the number of PRUs specified by the S parameter, or retry at another time when space might be available.	PFM
READ FILE BUSY - filenam.	The read file is found to be busy (direct access file only).	Retry after file is not busy.	SUBMIT
READY.	The system is ready to process the next command.	Enter your next command.	IAFEX, TELEX

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
RECORD SIZE EXCEEDS 500.	The maximum line length for a record to be converted (500 characters) was exceeded.	Split lines that are too long into two or more lines.	CONVERT
RECOVERY IMPOSSIBLE.	During an attempted recovery, one of the following conditions occurred. - User entered incorrect terminal number. - The system could find no record of the specified user being logged in on the given terminal number in the past 10 minutes. - Information on the user necessary for recovery was incorrect because of a system malfunction.	- Retry with correct terminal number. - Restart terminal session. - Restart terminal session.	IAFEX, TELEX
RESEQ NUMERIC PARAM ERROR.	A parameter which is supposed to be numeric contains a nonnumeric character.	Correct error and rerun.	RESEQ
RESERVED FILE NAME.	A reserved file name was incorrectly used.	Choose a nonreserved file name.	OPLEDIT, EDIT, DATADEF, IAFEX, TELEX
SECURE MEMORY, DUMP DISABLED.	An attempt was made to dump memory protected by the system. This message is also issued when a user enters a DMD or DMP statement at a terminal if the statements are not part of a procedure.	Refer to Security Control in section 3 and the DMD and DMP statement descriptions in section 9 of the NOS Reference Manual, volume 1, or to Security Considerations in section 2 of the NOS Reference Manual, volume 2. To obtain memory dumps from a time-sharing job, include the DMD or DMP statement in a procedure or in an ENTER statement. A DMD statement can be included only in a batch job.	!AJ
SRU LIMIT ENTER S TO CONTINUE OR CR KEY TO STOP:	The user has exceeded his system resource unit (SRU) limit during a terminal session.	To continue, allocate nnnnn additional SRUs with S,nnnnn command or enter S alone to allocate an installation-defined number of SRUs.	IAFEX, TELEX

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
SUBMIT COMPLETE. JOBNAME IS jobname	The SUBMIT statement successfully entered the file in the input queue. The system's job name for the entered job is jobname.	None.	SUBMIT
SUBMIT FILE EMPTY.	An EOR or EOF was encountered on the submit file before any data was found.	Rewind submit file, verify format of data and that the file is a local file, and retry operation.	SUBMIT
SYSTEM BUSY, PLEASE TRY LATER.	The maximum number of terminals allowed for the site are currently active on the system.	Try again at a later time.	IAFEX, TELEX
SYSTEM CLOSED.	The system is not available for use by terminals.	None.	NETVAL, IAFEX, TELEX
SYSTEM ERROR.	A software or hardware system error occurred.	Inform site analyst immediately.	MODVAL, PFM
TERMINAL nnnnn TIME OUT.	The terminal has been logged off after no input was received for several minutes. (Limit is set by installation.)	Attempt recovery or log in again.	IAFEX, TELEX
TIME LIMIT ENTER T TO CONTINUE OR CR KEY TO STOP:	The system detected that the time limit for the job step has expired.	To continue, reset time limit with T,nnnnn command or enter T alone to set time limit to an installation-defined default.	IAFEX, TELEX
TL NOT VALIDATED.	The time limit requested exceeds that for which the user is validated.	Request smaller time limit.	CPM, ACCFAM
TOO MANY FILES - NOT ALL PROCESSED.	The job had more files than could be processed.	Issue a series of REWIND statements or commands specifying file names. Inform the site analyst if this error occurs frequently.	MFILES
TOO MANY PARAMETERS.	More parameters were entered (including null parameters) than are allowed for command.	Ensure accuracy of entry.	GTR, IAFEX, TELEX,
TRANSACTION SYSTEM CLOSED.	The transaction facility (TAF/TS) is not operative.	Contact system operator.	TELEX

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
USER DAYFILE DUMPED.	The user dayfile dump is complete.	None.	DAYFILE
WRITE ON READ-ONLY FILE lfn AT addr.	Either the user attempted to write on a file with write interlock or the direct access file was not attached in WRITE mode.	Reattach file in write mode or clear write interlock.	CIO, IAFEX, TELEX
filename ALREADY PERMANENT, AT addr.	The user has already saved or defined a file with the name specified.	Save or define file using different file name or purge existing file.	PFM
pfn BUSY, AT addr.	The specified direct access file is attached in an incompatible mode, or it is currently being accessed by one of the following. - More than 77B users in READ mode - More than 77B users in READAP mode - More than 7777B users in READMD mode	Reissue ATTACH until file becomes available, or issue ATTACH specifying NA option.	PFM
lfn EMPTY, AT addr.	The file specified on a SAVE request contains no data.	Verify that file contains data and retry.	PFM
nnn FILE(S) PROCESSED.	The operation was performed on nnn files.	None.	MFILES
pfn FOUND. or un FOUND.	The file name specified on the CATLIST (LO=0) statement was found in the user's catalog or the file name or user number specified on the CATLIST (LO=P) statement was granted permission.	None.	PFM
proc IS A CCL PROCEDURE.	Procedure specified in CALL statement is a CCL procedure.	Use BEGIN command.	CONTROL
pfn IS DIRECT ACCESS, AT addr.	An indirect access file operation was attempted on a direct access file.	Use the appropriate direct access file request.	PFM
pfn IS INDIRECT ACCESS, AT addr.	A direct access file operation was attempted on an indirect access file.	Use the appropriate indirect access file request.	PFM
filename NOT FOUND.	RESTART was unable to retrieve a file named, but not included, on filename.	Verify that filename is valid.	RESTART
proc NOT FOUND.	Procedure specified in CALL statement cannot be found.	Verify that procedure name is correct and retry.	CONTROL
filename NOT FOUND, AT addr. or un NOT FOUND, AT addr.	One of the following. - The specified permanent file could not be found. - The specified user number could not	Verify that file name/ user number is correct, that access permission has been granted, and	PFM

MESSAGE -----	SIGNIFICANCE -----	ACTION -----	ROUTINE -----
	<p>be found.</p> <ul style="list-style-type: none"> - The user is not allowed to access the specified file. - The specified local file could not be found or was an execute-only file. 	that correct access is being attempted.	
filenam NOT ON MASS STORAGE, AT addr.	The file to be saved is not on mass storage; the first track of the file is not recognizable.	Verify that file is on mass storage.	PFM
pfn PERMANENT ERROR, AT addr.	The specified direct access file resides on the Mass Storage Facility (MSF) and has data errors that cannot be corrected.	Ask site analyst about file recovery from a backup copy.	PFM
nnnnn RECORDS CONVERTED.	Informative message indicating the number of records (nnnnn) converted from one character set to another.	None.	CONVERT

GLOSSARY

C

Account Block	That portion of a session from login to logoff or entry of a CHARGE statement which signals a new account block.
Alphanumeric	Consisting of alphabetic and/or numeric characters only.
Answerback Code	A character code that uniquely identifies a teletypewriter unit. If this code is present in the user's validation file, the user's activities are limited to this particular unit. The code is set within the terminal. The system checks only the first 10 characters.
APL	A programming language. An interactive interpreter available to the user.
Append	Add information to the end of a file.
ASCII	American National Standard Code for Information Interchange; the 128-character set utilized by the terminal user.
Auxiliary Device	Mass storage permanent file device used to supplement storage provided by the normal permanent file devices associated with the system (refer to family device).
BASIC	<ol style="list-style-type: none">1. Beginner's all-purpose symbolic instruction code is an elementary programming language available to the user.2. Subsystem that uses the BASIC compiler.
Bit	An abbreviation of binary digit. It is a single digit, 0 or 1, in a binary number. Bit is also used to represent the smallest unit of information. A central memory word (one storage location) contains 60 bits.
BOI	Beginning-of-information.
Catalog	The list of names of permanent files belonging to a particular user number; this list also contains information about the permanent files.
Compile	To translate a program from a higher level programming language (for example, FORTRAN or BASIC) into machine instructions called object code.
Control Language Statements	Statements that the user can insert in the control statement record of a job to initiate tests, transfers, and loops within that record.
Control Statement	A set of characters used to signify a specific set of instructions. Control statement is used when referring to nontime-sharing and batch instructions.

Control Statement Record	The first, and possibly only, record on an INPUT file or a SUBMIT file consisting of statement images that start with a job statement and end with the first EOR, EOF, or EOI.
CYBER Record Manager	A software product supported under NOS that allows a variety of record types, blocking types, and file organizations to be created and read. The execution time input/output of COBOL 4, COBOL 5, FORTRAN Extended 4, FORTRAN 5, Sort/Merge 4, ALGOL 4, BASIC, and DMS-170 products is implemented through CYBER Record Manager. The system input/output of NOS is not implemented through CYBER Record Manager. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines. COMPASS programs can use either CYBER Record Manager or NOS input/output (CIO).
Default	A system-supplied option used when the option is not supplied by the user.
Device	Peripheral equipment.
Direct Access File	A permanent file that can be attached to the user's job. All changes to this file are made on the file itself rather than a temporary copy of the file (refer to indirect access file).
Directive	Control information that is in a separate file or record of the job.
Display Code	A character code set used to represent alphanumeric and special characters (refer to appendix A).
ECS	Extended core storage. An additional memory available as an option. This memory can only be used for program and data storage, not for program execution. Special hardware instructions exist for transferring data between central memory and ECS.
EOF	End-of-file is a boundary within a sequential file but not necessarily the end of a file that can be referenced by name. The actual end of a named file is defined by an EOI. CYBER Record Manager divides files into partitions; therefore, a NOS multifile file is a multipartition file.
EOI	End-of-information.
EOR	End-of-record is the terminator of a logical record.
Empty Record	A PRU that contains no user data (same as zero length PRU).
Error Flag	A character or bit that signals the occurrence or presence of an error.
Exchange Package	A table that contains information used during job execution. It is printed as part of the output when a job aborts.

Executable Object Code	Machine language instructions that can be executed directly by the machine. A compiled program is composed of executable object code.
Family Device	Mass storage permanent file device associated with a specific system. A family may consist of 1 to 63 logical devices. Normally, a system runs with one family of permanent file devices available. However, additional families may be introduced during normal operation. This enables the user associated with the additional families to access their permanent files via an alternate system.
Family Name	The name that is associated with a group of permanent file devices.
Field Length	Number of central memory words required to process a job.
File	Set of information that begins at BOI and ends at EOI and that is referred to by a logical file name. This is the only definition of a file in CYBER Record Manager and the languages that use CYBER Record Manager. In NOS, a file is also defined as that portion of a file terminated by EOF; thus, a multifile file can exist. Generally, when an NOS command has a parameter that is a file name, that parameter refers to the BOI and EOI definition. Files within a multifile file cannot be individually referenced by name. When an NOS command has a parameter that specifies the number of files, that parameter uses the EOF definition.
Flag	A character or bit that signals the occurrence or presence of a particular condition.
FORTRAN	<ol style="list-style-type: none"> 1. Formula Translation, a high-level language consisting of symbols and statements that can be used to create a program closely resembling mathematical notation. 2. The subsystem that uses the FORTRAN Version 5 compiler.
FTNTS	The subsystem that uses the FORTRAN Extended Version 4 compiler.
Indirect Access File	A permanent file that is accessed only by making a temporary copy of the file (GET, OLD, or LIB commands). It is created or altered by saving or substituting the contents of an existing temporary file (REPLACE or SAVE commands).
Interactive	Job processing in which the user and the system communicate with each other, rather than the user submitting his job and receiving output, having no control over the job while processing occurs.
Library File	A permanent file residing under the special user number LIBRARY. Several other definitions exist (refer to the NOS Reference Manual, volume 1.)
Line	Refer to zero byte terminator.

Local File	Any file that is currently associated with a job.
Locked File	A file on which a user cannot write.
Logical Record	A logical record on mass storage is terminated by an EOR; on tape, it is terminated by the conditions described in the NOS Reference Manual, volume 1, for individual tape formats. Often, a logical record contains more than one CYBER Record Manager record. Since CYBER Record Manager defines a line as a logical record, a NOS logical record may contain several record manager logical records.
Login	A procedure that is performed by the terminal user to establish communication with the system.
Mass Storage	The equipment used to hold temporary and permanent files within the system.
Master Device	Mass storage device that contains the user's permanent file catalog entries, all his indirect access files, and all, part, or none of his direct access files.
Nonallocatable Device	A device, such as a magnetic tape, which is used by only one job at a given time.
Normal Mode	The standard 63- or 64- character set where all lowercase letters are converted to uppercase.
Null Character	Character (NUL) produced in paper tape by punching only feed holes. These are normally used for punching tape leaders and trailers.
Object Code	Executable machine language instructions. An object code program need not be recompiled each time it is executed.
On-Line	Equipment under direct control of the computer.
Order Dependent	Items which must appear in a specific order.
Order Independent	Items which need not be given in any specific order. Parameters may be order independent. For example, the UN=usernum parameter in the GET command is order independent.
Parameter	A variable that is given a specific value for a particular purpose or process.
Password	<ol style="list-style-type: none"> 1. A system access word that must be used in addition to the user number at login. 2. A file access word that defines access to a particular file by alternate users.
Permanent File	A file which is created by a user, stored under his user number in the permanent file system, and is removed from permanent file storage only when specified by a user who has write permission.
Permanent File Device	Mass storage device defined by the user's site to hold permanent files.

Permanent File Family	Permanent files which reside on the family devices of a specific system.
Port	The point at which a communication line is attached to the computer.
Post Radix	A letter following a numeral that indicates the base numbering system.
Prefix Character	A character that has a special significance to a program or the operating system and is used in front of a string of characters.
Primary File	Any temporary file created with the OLD, LIB, PRIMARY, or NEW command. The primary file is assumed to be the file on which most time-sharing operations are performed unless another file is specified.
Private Auxiliary Device	Auxiliary device associated with a specific user. Only that user may create files on the device, although he may permit other users to access files which reside on the device.
Procedure File	Under NOS, a file containing control language and control statements, but not commands defined only for time-sharing.
PRU	Physical record unit. The amount of information transmitted by a single physical operation of a specified device. A PRU for mass storage devices is 64 central memory words in length. Refer to the NOS Reference Manual, volume 1 or 2, for information concerning tape PRU sizes.
Public Auxiliary Device	Auxiliary device available for access by all validated users knowing the correct pack name. Additional validation is required to create or replace files on an auxiliary device (refer to LIMITS Command in section 4).
Read-Only Permission	If a user has read-only permission on a file, he can read and execute the file. He cannot write, modify, append, or purge the file.
Record	A unit of information which is interchangeable with logical records in NOS.
	In CYBER Record Manager and its language processors, a unit of information produced by a single write request.
	In FORTRAN, a formatted write produces zero byte terminated records, and an unformatted write produces W type records.
Record Separator	In NOS, another name for an EOR.
Record Type	May have one of several meanings, depending upon its context. In CYBER Record Manager, there are seven record types defined by the RT field in the FIT.

Recovery	The process by which a terminal user reestablishes communication with the system after inadvertent disconnection and is able to continue processing at the point of disconnection.
Rollout File	A file containing a job (and system information) that has been temporarily removed from the main processing area of the system.
Rubout Characters	Characters created by pressing the RUB OUT key on the terminal. These are considered null input by the system and these (or NUL) characters are required in paper tape output.
Security Count	The number of security violations the user has left before he is denied access to the system.
Sequence Numbers	The AUTO command adds line numbers at the beginning of each line of a file. If a file uses sequence numbers, zero byte terminated lines or logical records are implied.
Sequential File	A file in which records are accessed in the logical order in which they occur. Any file can be accessed sequentially. Sequential files must be accessed sequentially because no key or address is associated with each record in the files. All CYBER Record Manager files are considered sequential files by NOS.
Source Code	Code input to the computer for later translation into executable machine language instructions (object code).
SRU	Refer to System Resource Unit.
String	A sequence of characters.
Subsystem Flag	A character or bit that tells the system that a particular subsystem is associated with a file.
System	Refers to the Network Operating System.
System File	A file that can be accessed only by a system program.
System Resource Unit	A unit of measurement of system usage. The number of SRUs includes the CPU time, memory usage, and I/O activity.
System Utilities	System programs used to perform system functions.
Temporary File	This file is either a file created by the user that is not a permanent file, or it is a copy of a file that already exists in the system. All temporary files no longer exist once they are returned to the system (either specifically or at user logout).
Time Slice (CPU)	The amount of CPU time a job is allowed to use before the system lowers its priority to allow other jobs to execute.
TTY	Teletypewriter.

Validated User	A validated user is a person who has access to certain system resources. He can be allowed to access one or more resource.
Validation File	File containing validation information for all users (user numbers, passwords, resources allowed, and so on).
Write Interlock	Write interlock ensures that only one person at a time can attach a direct access file in write mode. A direct access file that is attached in write mode is in write interlock.
W Type Record	CYBER Record Manager record type in which user data is preceded by a system-supplied control word. FORTRAN unformatted writes and Sort/Merge use W type records as default record types. EOF and partition boundaries are not equivalent on files with this type of record.
XEDIT	The extended text editor available for use on NOS.
Zero Byte Terminator	The 12 bits of zero in the low order position of a central memory word. Either the zero byte terminator is used to terminate a line of coded information listed at the terminal or line printer, or it is used to represent cards entered through a card reader. Files with names INPUT and OUTPUT have such terminators while in storage. Any file to be displayed at a terminal must also have such terminators for each line to be displayed correctly. A record with such a terminator in CYBER Record Manager is a zero-byte record (Z type record). The SUBMIT command requires files whose lines are terminated with zero bytes. A record (marked by EOR) in NOS may contain one or several zero-byte records. In display code, two colons create 12 bits of zeros. If two consecutive colons occur in a file that contains zero-byte records, they may be stored in the lower order portion of a word and create a zero-byte terminator.
	Files created at a terminal under AUTO and TEXT commands or by using Text Editor or XEDIT contain zero-byte terminated records.
Zero Length PRU/Record	A PRU that contains no user data.

MASS STORAGE DEVICE STATISTICS

D

The mass storage devices supported by the system and the size of blocks allocated for direct access files are as follows:

Device Code	Device	Block Size		
		PRUs	CM Words	Characters
DE	Extended core storage (ECS)	16	1024	10240
DIn	844-21 Disk Storage Subsystem (half track) (1≤n≤8)	n*107	n*6848	n*68480
DJn	844-4x Disk Storage Subsystem (half track) (1≤n≤8)	n*227	n*14528	n*145280
DKn	844-21 Disk Storage Subsystem (full track) (1≤n≤8)	n*112	n*7168	n*71680
DLn	844-4x Disk Storage Subsystem (full track) (1≤n≤8)	n*227	n*14528	n*145280
DMn	885 Disk Storage Subsystem (half track) (1≤n≤3)	n*640	n*40960	n*409600
DP	Distributive data path to ECS	16	1024	10240
DQn	885 Disk Storage Subsystem (full track) (1≤n≤3)	n*640	n*40960	n*409600

In this table, n indicates the unit count for multiunit devices and x is 1 or 4.

If the user's validation permits (refer to LIMITS Command, section 4), the maximum size of an indirect access file is the device limit less the space allocated for catalog information and other files.

The size of a direct access file is limited by the device; it may be limited by validation if the file is to be accessed in write mode. There are no other system limitations.

TEXT EDITOR COMMAND INDEX

E

This is a list of the commands available under Text Editor. For additional information, refer to the Text Editor Reference Manual.

ADD(S)	ES:/string/;n
ADD(S);n	ES:/string1/,/string2/
ADD(S):/string/	ES:/string1/,/string2/;n
ADD(S):/string/;n	
ALIGN	EXTRACT
ALIGN;n	EXTRACT;n
ALIGN:/string/	EXTRACT:/string/
ALIGN:/string/;n	EXTRACT:/string/;n
ALIGN:/string1/,/string2/	EXTRACT:/string1/,/string2/
ALIGN:/string1/,/string2/;n	EXTRACT:/string1/,/string2/;n
BLANK(S)	FIND(S)
BLANK(S);n	FIND(S);n
BLANK(S):/string/	FIND(S):/string/
BLANK(S):/string/;n	FIND(S):/string/;n
BLANK(S):/string1/,/string2/	FIND(S):/string1/,/string2/
BLANK(S):/string1/,/string2/;n	FIND(S):/string1/,/string2/;n
CHANGE(S)	INSERTS:/string1/,/string2/;n
CHANGE(S);n	
CHANGE(S):/string/	LENGTH;n
CHANGE(S):/string/;n	LENGTH;*
CHANGE(S):/string1/,/string2/	
CHANGE(S):/string1/,/string2/;n	
CLEAR	LINE
DEFTAB	LIST(S)
DEFTAB:/tabchar/	LIST(S);n
DELETE(S)	LIST(S):/string/
DELETE(S);n	LIST(S):/string/;n
DELETE(S):/string/	LIST(S):/string1/,/string2/
DELETE(S):/string/;n	LIST(S):/string1/,/string2/;n
DELETE(S):/string1/,/string2/	
DELETE(S):/string1/,/string2/;n	
END	LISTAB
ES	MERGE:/lfn/
ES;n	MERGE:/lfn/;n
ES:/string/	MERGE:/pfn/
	MERGE:/pfn/;n
	MERGE:/lfn/,/string/
	MERGE:/lfn/,/string/;n
	MERGE:/pfn/,/string/
	MERGE:/pfn/,/string/;n

NUMBER(S)
NUMBER(S):/string/
NUMBER(S):/string1/,/string2/

RESET

RS:/string/
RS:/string/;n
RS:/string1/,/string2/
RS:/string1/,/string2/;n

SET
SET;n

SET;-n
SET:/string/
SET:/string/;n

STOP

TAB
TAB:/t₁,...,t_n/

WIDTH;n (6 ≤ n ≤ 150)

XEDIT COMMAND INDEX

F

This is a list of the commands available under XEDIT. For additional information, refer to the XEDIT Reference Manual.

ADD n	FBADL n
ADDLN(S)	FILE, fname, mode
BOTTOM	FINDLL n
BRIEF	HELP, command
BRIEF+	INPUT e
BRIEF-	INSERT(B) n
CHANGE/string1/string2/n	line number
CHANGE/string1a...string1b/string2/n	LISTAB
CHANGES/string1/string2/m	LOCATE/string/n
CHANGES/string1a...string1b/string2/m	LOCATE/string1...string2/n
COPY(D)fname n	LOCATE/string1--string2/n
COPY(D)fname/string/n	LOCATE/- -string2/n
COPY(D)fname/string1...string2/n	MODIFY
COPY(D)fname/string1--string2/n	NEXT n
COPY(D)fname/- -string2/n	NEXT -n
(CR)	NOBELLS
DBADL n	OCTCHANGE oct1 oct2 n
DEFTAB char	PRINT n
DELETE n	QMOD n
DELETE/string/n	QUIT, fname, mode
DELETE/string1...string2/n	READ(P) fname ₁ fname ₂ ... fname _n
DELETE/string1--string2/n	REPLACE n
DELETELN	REPLACELN ln n
DELIMIT char	RESTORE
DEOF m	RMARGIN m
DEOR m	STOP
DLBLANKS n	
eEDIT	
END, fname, mode	
EXPLAIN	

TAB(S)t ₁ t ₂ ... t _n (n≤8)	
TEOF	VERIFY
TEOF+	VERIFY+
TEOF-	VERIFY-
TEOR	WEOF
TEOR+	WEOR
TEOR-	WHERE
TOP	WMARGIN lm rm
TOPNULL	YQMOD n
TRIM	Y\$ cmd ₁ \$ cmd ₂ \$...\$cmd _n
TRIM+	Z\$ cmd ₁ \$ cmd ₂ \$...\$cmd _n
TRIM-	
TRUNCATE n	.n
	-n

INDEX

- Abbreviating command 4-1; 8-1
Access command 4-8
Access mode 5-1
Access subsystem 1-2; 4-8
Accidental disconnect 6-8
Accumulated CPU time 4-14,22
Acoustic coupler 2-3
Add line numbers 6-2
Alter line number 4-3; 6-2
Alternate catalog requests 5-14,23
Alternate user access 5-23
Answerback capability 4-39
APL character set A-1,4,5,6
APL command 4-12
APL print selection (refer to TERM command)
APL print terminal 2-4,5
APPEND command 5-6
Append mode 5-2
ASCII character set 4-2,3; A-3,4,5,6
ASCII code terminals 1-1; 2-1,2,4,6;
4-2,3,4,19,24; 6-4,6; A-1,2
ASCII command 4-3
ATTACH command 5-6
AUTO command 4-3
Automatic generation of line numbers 4-3
Automatic logout 4-7
Automatic read-only permission 3-3
Automatic subsystem selection 3-5; 4-8;
5-11
Auxiliary device 4-39; 5-3,16
- Backspace key 1-3; 2-6; 4-2
Backup requirement parameter 5-1
BASIC command 4-9
BASIC subsystem 1-2; 4-9
BATCH command 4-9; 8-1
Batch job submittal 4-34; 7-1
Batch output 8-2
Batch subsystem 1-2; 4-9; 8-1
Baud rate switch 2-2,3
BEGIN command 4-12
Beginning-of-information 3-1
BINARY command 4-13
Binary file 4-9,13
Binary input mode 4-13
Blank lines 3-8; 6-4
- Block edit terminal use 10-1
Block size
 Direct access file 3-4; D-1
 Indirect access file 3-3
Block transmission terminal 2-4
BOI (beginning-of-information) 3-1
BREAK key vi
BRIEF command 4-6
BYE command 2-11; 3-10; 4-14
- CALL command 4-14
Calling procedure files 4-12,14,35
Cancel line 4-2
Carriage return (refer to RETURN key)
Catalog entries 5-13
CATLIST command 5-13
CCL 4-12
CE parameter 5-1
Central memory character equivalents A-2
Central processor time limit 4-33
cEOF directive 7-3
cEOR directive 7-3
CHANGE command 5-9
Changing character sets 4-3,5; A-1
Character count delay 4-6
Character set A-1
Character set mode 4-3,5
CHARGE command 2-10; 4-16
Charge/project number 2-10; 4-16
cJOB directive 7-3
CLEAR command 3-10; 4-16; 5-8
Clear error flag (CE) parameter 5-1
cNOPACK directive 7-3
cNOSEQ directive 7-3
cNOTRANS directive 7-4
CNTL (control) key vi
Command
 Abbreviation 4-1
 Entry of 4-1,2
 Error correction 4-2
 Format, permanent files 5-1
 Information 4-24
 Maximum number of characters 4-1
 Minimum number of characters 4-1
 Processing 4-1
 Communication between users 4-8,19
 Compile/execute file 4-31

Compiled program 4-9
 Compress a file 4-29
 Continue job after suspension 6-5
 Control bytes 1-5
 Control characters vi, 9-2
 Control keys 2-6
 Control language 4-12,13
 Control record 7-6,7
 Control statements 7-1; 8-1
 Conventions,
 Manual v
 Numeric 4-2
 Operating 1-3
CONVERT command A-1
COR (refer to **TERM** command)
CORAPL (refer to **TERM** command)
 Correcting errors 4-2
 Correcting input line 1-3; 4-2; 6-1
 Correcting program statements 6-1
 Corrections of paper tape 9-1
 Correspondence code terminals 1-1;
 2-1,2,4,6; 4-2,3,5,25,32; 6-4; A-1
cPACK directive 7-3
 CPU time, accumulated 4-14,23
(CR) v; 4-16
cREAD directive 7-4
 Creating a file 4-20,28,34,37; 5-9
cREWIND directive 7-5
cSEQ directive 7-3
CSET command 4-6
cTRANS directive 7-3
CTRL key vi; 9-2
CYBER Control Language 4-12
CYBER Interactive Debug 4-19

Data entry 1-3
 Data set 2-3
 Dayfile 7-7
DAYFILE command 4-16
DAYFILE control statement 4-16; 7-8,9
DEBUG command 4-19
DEFINE command 5-9
 Delay carriage return (refer to **ROUT** command)
 Delete input lines 1-4; 4-2; 6-1
 Delete program statements 6-1
 Device 3-1; 5-4
 Device type parameter 5-4; D-1
DIAL command 4-19
 Dial-in procedure 2-3
 Direct access files 3-4; 5-6,7,8,9
 Direct entry of data 4-20,37
 Disconnect terminal 2-11; 4-14
 Display code A-2,3,4,5,6
 Dropping local files 5-3
 Duplex 2-3; 4-5

ECS and LCM differences vi
ECS limit 4-41
EDIT command 4-20
 Editing a program 4-20,37; 6-1
 End-of-file 3-1; 7-3
 End-of-information 3-1
 End-of-record 3-1; 7-3
ENQUIRE command 4-21,22
Entry
 Batch control statements 8-1
 Charge/project number 2-10; 4-16
 Commands 4-1; 8-2
 Direct 4-20,37
 Errors 4-1,2
 Family name 2-8
 File status 4-22,25; 7-1
 Lowercase letters 4-3
 Numbers 4-2
 Password 2-7; 4-29
 Subsystem 2-9; 4-8
 System commands 4-1
 Terminal identification 2-4
 Terminal type 2-8,9
 User number 2-7
EOF (end-of-file) 3-1; 7-3,7
EOI (end-of-information) 3-1
EOR (end-of-record) 3-1; 7-3,7
EOT code 10-1
 Error control 6-13
 Error correction 4-2
 Error flag 4-22
 Error messages B-1
 Error processing 7-9
ESC (ESCAPE) key vi; 4-2
 Escape character 7-1
 Escape code convention vi; 4-2
ETX character 10-1
 Even/odd parity switch 2-3; 4-6
 Even parity 4-6
 Exchange package 6-13
EXECUTE command 4-9
 Execute mode 5-2
 Execute subsystem 1-2; 4-9,31
EXIT control statement 7-7,9
 Exit from batch subsystem 8-2
 Extended memory differences vi

Family name 2-7
 Field length required for control cards
 8-1
 Field length specified on **RUN** command
 4-31
File
 Additions 5-6
 Catalog 3-3; 5-13
 Category 5-1,2

Control 3-9
 Copy 4-28; 5-10,11
 Direct access 3-4; 5-9
 Entering information 4-20,34,37
 Indirect access 3-3
 Inhibit sorting 3-6
 Local 3-4
 Name entry 2-11; 3-2,7; 4-8,9,10,11;
 5-6; 7-1
 Organization 3-1
 Permanent 3-2
 Permission information 5-2
 Primary 3-4; 4-26,28,30
 Processing 3-2; 5-6
 Reformatting 7-2
 Reserved names 3-2
 Residency 5-4,16
 Sorting 3-6
 Status 4-21,22,25
 Structure 3-1
 Subsystem association 3-5; 5-5,9,13
 Temporary 3-4
 Types 3-3
 Usage 3-1
 FORTRAN command 2-9; 4-10
 FORTRAN Extended (refer to FTNTS)
 FORTRAN subsystem 1-2; 4-10
 FTNTS 1-2; 2-9; 4-11
 FULL command 4-5
 Full duplex/half duplex (FDX/HDX) switch
 2-3; 4-5

 Generate binary (object code) file 4-31
 GET command 3-5,9; 5-10
 Glossary C-1
 GOODBYE command 2-11; 3-10; 4-24

 HALF command 4-5
 Hardwired terminal 2-1,3
 Header 4-26,31
 Header suppression 4-5
 HELLO command 2-12; 3-10; 4-24
 HELP command 4-24

 I key 4-24
 Improper login 2-8
 Increment job field length 4-31
 Indirect access files 3-3; 5-10,11,13
 Input line
 Termination 1-3
 Correction 1-3
 Deletion 1-4

 Input mode control 1-5
 Input/output conventions 1-3
 Input to executing program 1-4
 Inserting statements in a program 6-1
 Interrupt a job 6-4
 Interrupt program output 1-4; 4-24; 6-4

 Job
 Name 7-1
 Status 4-16,21,22,34
 Suspension 6-4
 Termination 4-32,33
 JOB statement 7-2
 Job step limit 4-33

 KCL 4-14
 Keep local files 5-3
 Keyboard configuration 2-5
 Keyword 5-1,14

 LENGTH command 4-25
 Length of input/output lines 1-3
 LIB command 3-3; 4-26; 5-8,11,26
 LIMITS command 4-26,39
 Line correction 4-2
 Line editing 6-1
 Line feed (LF) character 9-2
 Line numbers 3-6; 4-3; 6-2
 Line overflow 1-3
 LIST command 3-7; 4-26
 Listing local files 4-26,27
 Listing primary file 4-26,27
 LNH (list-no-header) command 4-27
 Local file 3-4,9
 Local file status 4-22
 Locked file 4-25
 Logical file 3-1
 Logical record 3-1
 LOGIN command 2-12; 3-9; 4-27; 5-8
 Login procedure 2-1
 LOGOUT command 2-11; 3-9; 4-27; 5-8
 Logout procedure 2-11
 Lowercase letters 4-3

 Magnetic tape limit 4-39
 Mass storage devices 5-4,16; D-1
 Mass Storage Facility (MSF) 5-4,5,7
 Maximum length
 Batch control statement 8-1
 Command 4-1

Input line 1-3
Output line 1-3
Submit file line 7-5
MEMAPL (refer to TERM command)
Message
 Current job status 4-21
 Dayfile 4-16
 Input 1-3
Messages B-1
 Minimum number of characters in command
 4-1
 Modify mode 5-2
 MONITOR command 4-27
 MSF 5-4,5,7
 Multifile file 3-1
 Multiple file requests 3-1

NA parameter 5-3,17
ND parameter 5-3
NEW command 4-28; 5-8
No abort parameter 5-3,17
NORMAL command 4-6
 Normal mode 4-5
NOS character set A-1,4,5,6
NOSORT command 3-8; 4-28
NULL command 4-11
 Null permission 5-2
 Null subsystem 1-2; 3-5; 4-11; 5-6,12

Object code 4-32
 Odd parity 4-6
OLD command 3-3,4,5,9; 4-8,28; 5-8,11
 Output file name 5-17

Pack a file 3-8; 4-29
PACK command 3-8; 4-29
PACKNAM command 5-11
 Packname specifcations 5-3,16
Paper tape
 Control characters 9-2
 Corrections 9-4
 Input lines 9-2
 Off-line punching 9-3
 On-line punching 9-3
 Operations 9-1
 Output 9-5
 Reading 4-6
PARITY command 4-6
Parity switch 2-3; 4-6
PASSWOR command 4-29
 Password 2-7; 4-29; 5-4,9
 Password parameter 5-4,9

Permanent file
 Access 3-3,4
 Append file to 5-6
 Attach 5-6
 Catalog 3-3; 5-13
 Category 5-1,2
 Change 5-9
 Command format 5-1
 Commands 3-3,4,5
 Creation 5-9,13
 Direct 3-4
 Family (refer to family name)
 Granting access 5-12
 Indirect 3-3
 Limit 4-39
 Mode of access 5-2
 Name 5-6,15
 Permission mode 5-2
 Release 5-8
 Remove 5-12
 Replace 5-13
 Residency 5-4
 Retrieve 5-10,11
 Size 3-3,4
 Types 3-3,4
 Permission 5-2,12,15,16,23
 Permission information 5-14,18
 Permission mode 5-2
PERMIT command 5-12
 pfn 5-1
 Physical record units 3-1; 4-25
 Preferred residence parameter 5-4
PRIMARY command 4-30
 Primary file 3-4; 4-28
 Private file 5-2
 Private file access 5-23
 Procedure file usage 4-12,35; 8-2
 Program editing 6-1
 Program execution 3-9; 4-9,31
 Program input 1-4
 Program interruption 1-4; 6-4
 Program termination 1-4; 6-4
 PRU (physical record unit) 3-1; 4-25
 PRU limit 4-39
 Public file access 5-25
 Public files 5-2,25
 Punching a paper tape 9-3
PURGE command 5-12

Question mark 1-4

 Read mode 5-2
 Read-only permission 3-3
 Read with append mode 5-2

Read with modify mode 5-2
 Real time parameter 5-5,7
 Record 3-1
 Record compressing 4-29
 RECOVER command 4-30; 6-8
 Recovery 4-30; 6-8
 Reformatting a file 7-2
 Reformating directives 7-3
 Releasing local files 3-9; 4-14,16,28,30
 Releasing permanent files (refer to PURGE command)
 Remote job entry 4-34; 7-1
 RENAME command 4-30
 REPLACE command 5-13; 7-6
 Requesting system information 4-21,22
 RESEQ command 3-8; 4-29; 6-2
 Resequencing line numbers 4-30; 6-2
 Reserved file names 3-2
 RESOURC control statement 5-16
 RETURN command 4-30; 5-8
 RETURN key v
 Return local files 4-14,16,28,30
 REWIND command 4-29
 RFL system response 8-1
 RNH (run-no-header) command 4-30
 ROUT command 4-6
 Rubout 9-2
 RUN command 4-10,31,32
 Run compiled programs 4-9

S key 4-32
 SAVE command 3-5,6; 4-10; 5-11,13
 Security count 2-8; 8-3
 Semiprivate file 5-2,24
 Sending messages 4-19
 Set time limit 4-33
 SETASL command 4-33
 SETJSL command 4-33
 SETTL command 4-33
 S,nnnn 6-6
 SORT command 3-6; 4-33,34
 Sort flag 3-6; 4-33
 Source program 4-8; 6-1
 SRU limit 4-33; 6-5
 Standard character set A-1,4,5,6
 STATUS command 4-34
 Status of files 4-21,22,25
 Status of job 4-16,21,22
 STOP command 4-34; 6-5
 Stopping a running job 6-4
 SUBMIT command 4-34; 7-1
 Submit error processing 7-9
 Subsystem association 4-8,11; 5-5
 Subsystem descriptions 1-1
 Subsystem parameter 3-5; 4-8,11; 5-5,9,13
 Subsystem selection, automatic 3-5; 4-8; 5-13

Subsystem selection commands 2-9; 4-8
 SUMMARY command 4-34
 Suspend job 4-24; 6-4
 Switch checking (login) 2-3
 System commands 4-1
 System information 4-21,22

TAPE command 4-7
 Tape mode 9-3
 TAPE1 7-5
 Temporary files 3-4; 5-10
 TERM command 4-7
 Terminal
 Characteristics 4-5,6,7
 Control commands 4-3
 Control keys 2-6
 Disconnection 6-8
 Identification 2-4
 Messages B-1
 Number 2-8
 Parity 4-6
 Types 2-4,8,9
 Terminate
 Executing program 1-4; 2-6; 6-5
 Input line 1-3
 Job 4-32
 Output 1-4; 2-6; 4-32
 Program 1-4; 2-6; 4-32,33,34
 TEXT command 4-34
 Text Editor 4-20; 6-1; E-1
 Text Editor commands E-1
 Text mode 2-6; 4-34
 Time limit specification 4-33; 6-5
 TIMEOUT command 4-8
 Time-sharing activity 4-16
 Time-sharing job commands 4-12
 Time-sharing messages B-1
 T,nnnn 6-5
 TRAN command 4-11
 Transaction subsystem 4-11
 Transparent submit mode 7-3
 TTY (refer to TERM command)
 Types of permanent files 3-3,4

Unsuccessful login 2-1,8
 Unsuccessful recovery 6-8
 USER command 4-37; 8-3
 User number, entry of 2-7; 8-3
 User number parameter 5-6,15

Validation limits 4-39

Working (temporary) copy of file 3-4;
5-10

Working (temporary) files 3-4

Write interlock 3-4

Write mode 5-2,6,9

XEDIT commands F-1
X-off character 9-2

ZZZDUMP 6-13
ZZZZZxx 3-2

X,ccccccc 8-2

XEDIT command 4-37; 6-1

-pname 4-12,38

COMMENT SHEET

MANUAL TITLE: CDC NOS Version 1 Time-Sharing User's Reference Manual

PUBLICATION NO.: 60435500

REVISION: J

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ **STATE:** _____ **ZIP CODE:** _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

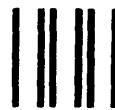
FOLD ON DOTTED LINES AND STAPLE

TAPLE

STAPLE

OLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS

PERMIT NO. 8241

MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

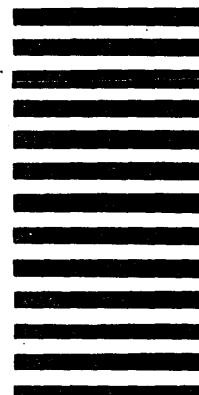
CONTROL DATA CORPORATION

Publications and Graphics Division

ARH219

4201 North Lexington Avenue

Saint Paul, Minnesota 55112



OLD

FOLD

CUT ALONG LINE

TITLE: CDC NOS Version 1 Time-Sharing User's Reference Manual

PUBLICATION NO.: 60435500

REVISION: J

External A/D

CONTROL DATA CORPORATION

Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

REASON FOR CHANGE:

DATE 05-23-80

Updated to reflect NOS 1.4 at PSR level 518 and to make numerous miscellaneous technical corrections. Included are the LIB command and the automatic packing of text files. This edition obsoletes all previous editions.

56657

COMMAND INDEX (Contd)

<u>Command</u>	<u>Description</u>	<u>Page</u>
MONITOR	Connects validated user in access subsystem with specified terminal	4-27
NEW	Creates new primary file	4-28
NORMAL	Clears ASCII, AUTO, BRIEF, CSET, NOSORT, PARITY, and TAPE commands	4-6
NOSORT	Prevents automatic sorting of primary file	4-28
NULL	Selects null subsystem	4-11
OLD	Retrieves copy of indirect access permanent file for use as primary file	4-28; 5-11
P	Continues program listing or execution	6-5
PACK	Compresses logical records of file into one logical record	4-29
PACKNAM	Directs permanent file requests to specified auxiliary devices	5-11
PARITY	Sets terminal parity (odd or even)	4-6
PASSWOR	Changes user's password	4-29
PERMIT	Grants explicit private or semiprivate file access permission	5-12
PRIMARY	Makes a temporary file the current primary file	4-30
PURGE	Removes specified permanent file(s) from the system	5-12
RECOVER	Resumes processing after system disconnection	4-30
RENAME	Changes name of temporary file(s)	4-30
REPLACE	Replaces contents of indirect access permanent file(s) with contents of temporary file(s)	5-13
RESEQ	Resequences or adds line numbers	4-30; 6-2
RETURN	Releases local file(s)	4-30
REWIND	Repositions file(s) to its beginning	4-30
RNH	Runs a program (does not print header)	4-30
ROUT	Sets amount of time necessary for terminal return function	4-6
RUN	Runs a program (prints header)	4-31
S	Increases SRU job step limit	6-6
S key	Terminates job transmitting output to terminal	4-32
SAVE	Creates indirect access permanent files	5-13
SETASL	Sets SRU account block limit	4-33
SETJSL	Sets SRU job step limit	4-33
SETTL	Sets time limit	4-33
SORT	Sorts a file with line numbers	4-33
STATUS	Same as ENQUIRE command	
STOP	Terminates program in execution	4-34
SUBMIT	Allows remote batch job entry	4-34; 7-1
SUMMARY	Lists resources used during session	4-34
T	Increases CPU time limit	6-5
TAPE	Selects paper-tape input mode	4-7
TERM	Redefines terminal characteristics	4-7
TEXT	Selects text mode	4-34
TIMEOUT	Changes no-timeout terminal to standard timeout status	4-8
TRAN	Selects transaction subsystem	4-11
USER	Supplies terminal number of specified user	4-37
XEDIT	Selects XEDIT	4-37
-pname	Same as BEGIN command	4-38

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

