



**NOS VERSION 2
REFERENCE SET**

Volume 3

SYSTEM COMMANDS

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

ALPHABETICAL LIST OF COMMANDS†

ACCESS ††	8-8	DMDECS	11-4	LOCK	9-24	ROUTE	9-37
ACCOUNT	6-2	DMP	11-5	LOGIN ††	8-18	RTIME	7-34
ALGOL	60496600	DMPECS	11-6	LOGOUT ††	8-18	RUN ††	8-23
ALGOL5	60481600	DOCUMENT	9-19	LO72	9-25	SATISFY	60429800
ALTER ††	8-29	DROP	7-6	MACHINE	7-23	SAVE	10-25
APEX	76070000	DUP ††	8-32	MAP	60429800	SCOPY	9-43
APL	60454000	EDIT	15-1;	MFL	7-23	SET	6-22;
APPEND	10-6		60450100	MODE	6-19		H-7
ASCII ††	8-2	ELSE	6-15	MODIFY	15-3;	SETASL	7-35
ASSIGN	9-1; 12-5	ENDIF	6-16		60450100	SETCORE	7-35
ATTACH	10-6	ENDW	6-16	MOVE ††	8-35	SETFS	9-46
AUTO ††	8-2	ENQUIRE	7-7	NEW	9-29	SETJOB	7-36
BASIC	19983900	ENTER	7-13	NOEXIT	6-21	SETJSL	7-37
BASIC ††	8-8	EVICT	9-21	NORERUN	7-24	SETPR	7-38
BATCH ††	8-9	EXECUTE	60429800	NORMAL ††	8-4	SETTL	7-38
BEGIN	4-19	EXECUTE ††	8-9	NOSORT ††	8-18	SKIP	6-27
BKSP	9-3	EXIT	6-17	NOTE	7-24	SKIPEI	9-46
BLANK	12-7	FCOPY	9-22	NULL ††	8-11	SKIPF	9-47
BRIEF ††	8-4	FILE	60495700	OFFSW	7-25	SKIPFB	9-47
BYE ††	8-13	FORTRAN ††	8-10	OLD	10-19	SKIPR	9-48
CALL	H-5	FTN	60497800	ONEXIT	6-21	SLOAD	60429800
CATALOG	16-4	FTNTS ††	8-11	ONSW	7-25	SORT	9-48
CATLIST	10-9	FTN5	60481300	OPLEDIT	15-6;	SORTMRG	60497500
CFO	7-2	F45	60483000		60450100	STIME	7-39
CHANGE	10-14	GET	10-18	OUT	9-29	SUBMIT	7-40
CHARGE	7-2	GO	7-13	PACK	9-30	SUMMARY	7-44
CKP	13-1	GOODBYE ††	8-15	PACKNAM	10-20	SWITCH	7-45
CLEAR	9-4	GOTO	H-4	PASSWOR	7-26	Status (cE) ††	8-13
COBOL	60496800	GPSS	84003900	PAUSE	7-26	Status (cS) ††	8-13
COBOL5	60497100	GTR	16-11	PBC	11-8	TCOPY	9-49
COMMENT	7-3	HELLO ††	8-15	PERMIT	10-21	TDUMP	9-52
COMMON	10-15	HELP ††	8-16	PLI	60388100	TEXT ††	8-24
COMPASS	60492600	HTIME	7-14	PRIMARY	9-30	TIMEOUT ††	8-5
CONVERT	9-4	IF	H-6	PROFILE	15-8	TRMDEF ††	8-5
COPY	9-6	IFE	6-17	PROTECT	7-27	UNLOAD	9-54
COPYBF	9-11	ITEMIZE	16-14	PURGALL	10-22	UNLOCK	9-55
COPYBR	9-12	Job	7-15	PURGE	10-23	UPDATE	15-10
COPYCF	9-13	KRONFEF	15-2	QGET	7-27	UPROC	7-45
COPYCR	9-15	LABEL	12-11	RBR	11-8	USECPU	7-46
COPYEI	9-16	LBC	1-7	READ ††	8-36	USER	7-46
COPYL	16-7	LDI	7-17	RECOVER ††	8-19	VERIFY	9-55
COPYLM	16-7	LDSET	60429800	REDUCE	60429800	VFYLIB	16-34
COPYSBF	9-17	LENGTH	7-18	RENAME	9-31	VSN	12-24
COPYX	9-18	LIB ††	8-16	REPLACE	10-24	WBR	11-9
CSET ††	8-4	LIBEDIT	16-17	REQUEST	9-32;	WHATJSN ††	8-24
CTIME	7-3	LIBGEN	16-32		12-21	WHILE	6-27
DAYFILE	7-4	LIBLOAD	60429800	RERUN	7-28	WRITE ††	8-38
DEBUG	60481400	LIBRARY	60429800	RESEQ	9-34	WRITEF	9-58
DEFINE	10-15	LIBTASK	60459500	RESOURC	7-28	WRITEN ††	8-39
DELETE ††	8-31	LIMITS	7-19	RESTART	13-2	WRITER	9-58
Detach ††	8-12	LIST ††	8-18, 33	RETURN	9-35	X ††	8-25
DIAL ††	8-15	LIST80	9-24	REVERT	4-25	XEDIT	15-14;
DISPLAY	6-14; H-6	LISTLB	12-19	REWIND	9-36		60455730
DMB	11-1	LOAD	60429800	RFL	7-33	*	6-3
DMD	11-4	LOC	11-7	ROLLOUT	7-34		

† Reference to a page number indicates the command is described in this manual; a manual publication number means the command is described in that manual. Manual titles are listed in the preface. Refer to the NOS System Maintenance Reference Manual for a list of system analyst-oriented commands.

†† You can only enter this command from an interactive job.



**NOS VERSION 2
REFERENCE SET**

Volume 3

SYSTEM COMMANDS

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

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	-	4-1	A	6-24	A	8-7	A	9-26	A
Inside Front Cover	A	4-2	A	6-25	A	8-8	A	9-27	A
Title Page	-	4-3	A	6-26	A	8-9	A	9-28	A
2	A	4-4	A	6-27	A	8-10	A	9-29	A
3	A	4-5	A	6-28	A	8-11	A	9-30	A
4	A	4-6	A	7-1	A	8-12	A	9-31	A
5	A	4-7	A	7-2	A	8-13	A	9-32	A
6	A	4-8	A	7-3	A	8-14	A	9-33	A
7	A	4-9	A	7-4	A	8-15	A	9-34	A
8	A	4-10	A	7-5	A	8-16	A	9-35	A
9	A	4-11	A	7-6	A	8-17	A	9-36	A
10	A	4-12	A	7-7	A	8-18	A	9-37	A
11/12	A	4-13	A	7-8	A	8-19	A	9-38	A
13	A	4-14	A	7-9	A	8-20	A	9-39	A
14	A	4-15	A	7-10	A	8-21	A	9-40	A
15	A	4-16	A	7-11	A	8-22	A	9-41	A
16	A	4-17	A	7-12	A	8-23	A	9-42	A
17	A	4-18	A	7-13	A	8-24	A	9-43	A
18	A	4-19	A	7-14	A	8-25	A	9-44	A
19	A	4-20	A	7-15	A	8-26	A	9-45	A
1-1	A	4-21	A	7-16	A	8-27	A	9-46	A
1-2	A	4-22	A	7-17	A	8-28	A	9-47	A
1-3	A	4-23	A	7-18	A	8-29	A	9-48	A
1-4	A	4-24	A	7-19	A	8-30	A	9-49	A
1-5	A	4-25	A	7-20	A	8-31	A	9-50	A
1-6	A	4-26	A	7-21	A	8-32	A	9-51	A
2-1	A	4-27	A	7-22	A	8-33	A	9-52	A
2-2	A	4-28	A	7-23	A	8-34	A	9-53	A
2-3	A	5-1	A	7-24	A	8-35	A	9-54	A
2-4	A	5-2	A	7-25	A	8-36	A	9-55	A
2-5	A	5-3	A	7-26	A	8-37	A	9-56	A
2-6	A	5-4	A	7-27	A	8-38	A	9-57	A
2-7	A	5-5	A	7-28	A	8-39	A	9-58	A
2-8	A	5-6	A	7-29	A	9-1	A	10-1	A
2-9	A	5-7	A	7-30	A	9-2	A	10-2	A
2-10	A	6-1	A	7-31	A	9-3	A	10-3	A
2-11	A	6-2	A	7-32	A	9-4	A	10-4	A
2-12	A	6-3	A	7-33	A	9-5	A	10-5	A
2-13	A	6-4	A	7-34	A	9-6	A	10-6	A
2-14	A	6-5	A	7-35	A	9-7	A	10-7	A
2-15	A	6-6	A	7-36	A	9-8	A	10-8	A
2-16	A	6-7	A	7-37	A	9-9	A	10-9	A
2-17	A	6-8	A	7-38	A	9-10	A	10-10	A
3-1	A	6-9	A	7-39	A	9-11	A	10-11	A
3-2	A	6-10	A	7-40	A	9-12	A	10-12	A
3-3	A	6-11	A	7-41	A	9-13	A	10-13	A
3-4	A	6-12	A	7-42	A	9-14	A	10-14	A
3-5	A	6-13	A	7-43	A	9-15	A	10-15	A
3-6	A	6-14	A	7-44	A	9-16	A	10-16	A
3-7	A	6-15	A	7-45	A	9-17	A	10-17	A
3-8	A	6-16	A	7-46	A	9-18	A	10-18	A
3-9	A	6-17	A	7-47	A	9-19	A	10-19	A
3-10	A	6-18	A	8-1	A	9-20	A	10-20	A
3-11	A	6-19	A	8-2	A	9-21	A	10-21	A
3-12	A	6-20	A	8-3	A	9-22	A	10-22	A
3-13	A	6-21	A	8-4	A	9-23	A	10-23	A
3-14	A	6-22	A	8-5	A	9-24	A	10-24	A
		6-23	A	8-6	A	9-25	A	10-25	A

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
10-26	A	16-8	A	B-13	A	G-5	A	Index-7	A
11-1	A	16-9	A	B-14	A	G-6	A	Index-8	A
11-2	A	16-10	A	B-15	A	G-7	A	Index-9	A
11-3	A	16-11	A	B-16	A	G-8	A	Index-10	A
11-4	A	16-12	A	B-17	A	G-9	A	Index-11	A
11-5	A	16-13	A	B-18	A	G-10	A	Index-12	A
11-6	A	16-14	A	B-19	A	G-11	A	Index-13	A
11-7	A	16-15	A	B-20	A	H-1	A	Index-14	A
11-8	A	16-16	A	B-21	A	H-2	A	Comment	
11-9	A	16-17	A	B-22	A	H-3	A	Sheet	A
12-1	A	16-18	A	B-23	A	H-4	A	Back Cover	-
12-2	A	16-19	A	B-24	A	H-5	A		
12-3	A	16-20	A	B-25	A	H-6	A		
12-4	A	16-21	A	B-26	A	H-7	A		
12-5	A	16-22	A	B-27	A	H-8	A		
12-6	A	16-23	A	B-28	A	H-9	A		
12-7	A	16-24	A	B-29	A	H-10	A		
12-8	A	16-25	A	B-30	A	H-11	A		
12-9	A	16-26	A	B-31	A	H-12	A		
12-10	A	16-27	A	B-32	A	H-13	A		
12-11	A	16-28	A	B-33	A	I-1	A		
12-12	A	16-29	A	B-34	A	I-2	A		
12-13	A	16-30	A	B-35	A	I-3	A		
12-14	A	16-31	A	B-36	A	I-4	A		
12-15	A	16-32	A	B-37	A	I-5	A		
12-16	A	16-33	A	B-38	A	I-6	A		
12-17	A	16-34	A	B-39	A	I-7	A		
12-18	A	16-35	A	B-40	A	J-1	A		
12-19	A	16-36	A	B-41	A	J-2	A		
12-20	A	16-37	A	B-42	A	J-3	A		
12-21	A	16-38	A	B-43	A	J-4	A		
12-22	A	16-39	A	B-44	A	K-1	A		
12-23	A	16-40	A	B-45	A	K-2	A		
12-24	A	16-41	A	B-46	A	K-3	A		
12-25	A	17-1	A	B-47	A	K-4	A		
13-1	A	17-2	A	B-48	A	K-5	A		
13-2	A	17-3	A	B-49	A	K-6	A		
13-3	A	17-4	A	B-50	A	K-7	A		
13-4	A	17-5	A	B-51	A	K-8	A		
14-1	A	17-6	A	C-1	A	K-9	A		
14-2	A	17-7	A	C-2	A	K-10	A		
14-3	A	17-8	A	C-3	A	K-11	A		
14-4	A	17-9	A	C-4	A	K-12	A		
14-5	A	17-10	A	C-5	A	L-1	A		
14-6	A	A-1	A	C-6	A	L-2	A		
14-7	A	A-2	A	C-7	A	M-1	A		
14-8	A	A-3	A	C-8	A	M-2	A		
14-9	A	A-4	A	C-9	A	M-3	A		
15-1	A	A-5	A	C-10	A	M-4	A		
15-2	A	A-6	A	C-11	A	M-5	A		
15-3	A	A-7	A	C-12	A	M-6	A		
15-4	A	A-8	A	C-13	A	M-7	A		
15-5	A	A-9	A	D-1	A	M-8	A		
15-6	A	A-10	A	D-2	A	M-9	A		
15-7	A	A-11	A	D-3	A	N-1	A		
15-8	A	A-12	A	D-4	A	N-2	A		
15-9	A	A-13	A	D-5	A	N-3	A		
15-10	A	A-14	A	E-1	A	N-4	A		
15-11	A	B-1	A	E-2	A	N-5	A		
15-12	A	B-2	A	F-1	A	N-6	A		
15-13	A	B-3	A	F-2	A	N-7	A		
15-14	A	B-4	A	F-3	A	N-8	A		
15-15	A	B-5	A	F-4	A	N-9	A		
16-1	A	B-6	A	F-5	A	O-1	A		
16-2	A	B-7	A	F-6	A	Index-1	A		
16-3	A	B-8	A	F-7	A	Index-2	A		
16-4	A	B-9	A	G-1	A	Index-3	A		
16-5	A	B-10	A	G-2	A	Index-4	A		
16-6	A	B-11	A	G-3	A	Index-5	A		
16-7	A	B-12	A	G-4	A	Index-6	A		

PREFACE

This manual describes the Network Operating System (NOS) Version 2. NOS controls the operation of CDC® CYBER 170 Computer Systems Models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 825, 835, and 855; CDC CYBER 70 Computer Systems Models 71, 72, 73, and 74; and CDC 6000 Computer Systems.

AUDIENCE

Volume 1, Introduction to Interactive Usage (60459660), is written for the novice.

Volume 2, Guide to System Usage (60459670), is written for the applications or systems programmer who is unfamiliar with NOS.

Volume 3, System Commands (60459680), is written for all NOS users.

Volume 4, Program Interface (60459680), is written for the experienced COMPASS applications programmer or systems programmer.

The reader of each volume should have a knowledge of the material contained in the previous volumes.

ORGANIZATION

The NOS reference set describes the external features of NOS 2.0. This reference set comprises four separate volumes.

Volume 1, Introduction to Interactive Usage, shows a user at an interactive terminal how to enter, run, and correct programs, and how to create, retrieve, and maintain permanent files. Other topics covered include physical terminal connection, and login/logout procedures.

Volume 2, Guide to System Usage, describes the general concepts of NOS and some of the utilities used with NOS. Topics included are job processing, file concepts, procedures, magnetic tape processing, modify, and loading files. This volume is a learning tool and does not contain comprehensive descriptions of all NOS commands.

Volume 3, System Commands, describes the system commands that form the user interface to NOS.

Volume 4, Program Interface, describes the COMPASS program interface to NOS. Detailed descriptions of function processors and macros available to COMPASS user programs are included.

CONVENTIONS

References

Throughout this manual, references to Volumes 1, 2, and 4 of the NOS Reference Set are in the form: refer to section n, (volume name). If a volume number is not specified, the reference is to Volume 3.

Extended Memory

Extended memory for the CYBER 170 Model 176 is large central memory (LCM) or large central memory extended (LCME). Extended memory for the CYBER 170 Models 825, 835, and 855 is unified extended memory (UEM). Extended memory for all other NOS computer systems is extended core storage (ECS) or extended semiconductor memory (ESM).

In this manual, the term extended memory refers to all forms of extended memory unless otherwise noted. However, in the context of a multimainframe environment or distributive data path (DDP) access, models 176, 825, 835, and 855 are excluded.

Programming information for the various forms of extended memory can be found in the COMPASS Version 3 Reference Manual and in the appropriate computer system hardware reference manual.

CYBER 70 Computer Systems References

References to the CYBER 70 Computer Systems pertain only to models 71, 72, 73, and 74.

Control Statement

The manuals of many NOS products use the term control statement instead of the term command. This manual uses the term command almost exclusively. You can consider the two synonymous.

Examples

The following conventions apply to examples that appear in this manual:

- Examples of actual terminal sessions that appear in this manual were produced on a terminal class 1 terminal in NORMAL character mode unless otherwise specified. Uppercase characters represent terminal output; lowercase characters represent user input unless otherwise noted. (However, user input that is displayed within the text of this manual is shown in uppercase characters). The vertical spacing in examples does not necessarily coincide with the spacing that appears on your terminal.
- Program examples are written either in FORTRAN 5 or BASIC 3 (hereafter referred to as just BASIC).
- This manual uses commas to delimit command parameters and periods to terminate commands. For clarity, however, commands that appear in text use a left parenthesis for their initial delimiter and a right parenthesis for their terminator.
- All batch jobs in this manual have the following form:

```
ujn.  
USER,username,password,familyname.  
CHARGE,*.
```

```
.  
.  
.
```

At all sites, you must specify a user job name (UJN) on a Job command and a user name and password on a USER command. The CHARGE command is not required at all sites. However, the system processes your user prologue only after it has processed a CHARGE command. Thus, you may want to insert a CHARGE(*) command in your batch jobs to ensure that your user prologue is executed even if your site does not otherwise require it.

Command Format

Interpret uppercase characters within command formats literally. Lowercase characters are variables and are described immediately following the line that shows the command format.

Special Keyboard Entries

This manual uses special notation to represent certain keyboard entries:

- $\text{\textcircled{CR}}$ denotes the message transmission key on the keyboard. Depending on the terminal class, this key may be the RETURN, CR, CARRIAGE RETURN, NEW LINE, SEND, or ETX key. For some terminal classes, the RETURN or NEW LINE key also denotes a message terminator, but the message is kept in the terminal buffer until the corresponding SEND or ETX key is pressed, sending all messages in the buffer to IAF. Appendix G supplies more information. IAF and the network respond to the $\text{\textcircled{CR}}$ by positioning the carriage to the first character position on the next line.
- ← designates the backspace character.
- The notation CTRL/x directs you to press the control key (which is labeled CTRL, CNTL, CNTRL, or similar characters) on the terminal and, while holding this key down, press the key specified by x. For example, CTRL/H means press and hold the control key while you press the H key.

RELATED PUBLICATIONS

The following is a list of NOS operating system manuals and NOS product set reference manuals.

The NOS Manual Abstracts is a pocket-sized manual containing brief descriptions of the contents and intended audience of all NOS and NOS products manuals. The abstracts can be useful in determining which manuals are appropriate for your use.

Control Data publishes a Software Publications Release History of all software manuals and revision packets it has issued. This history lists the revision level of a particular manual that corresponds to the level of software installed at the site.

These manuals are available through Control Data sales offices or Control Data Literature Distribution Services (308 North Dale, St. Paul, Minnesota 55103).

If you need a brief description of the product commands and their parameters, you should refer to the NOS Applications Programmer's Instant.

<u>Control Data Publication</u>	<u>Publication Number</u>
ALGOL-60 Version 5 Reference Manual	60481600
APEX III Reference Manual	76070000
AFL Version 2 Reference Manual	60454000
APT IV Version 2 Reference Manual	17326900
BASIC Version 3 Reference Manual	19983900
COBOL Version 5 Reference Manual	60497100
Communication Control Program (CCP) Version 3 Reference Manual	60471400
Common Memory Manager Version 1 Reference Manual	60499200
COMPASS Version 3 Reference Manual	60492600
Conversion Aids System Version 2 Reference Manual	19265358
CYBER Common Utilities Reference Manual	60495600
CYBER Database Control System 2 Database Administrator Reference Manual	60485200
CYBER Database Control System Version 2 Application Programming Reference Manual	60485300
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
CYBER 70 Model 71 Computer System Hardware Reference Manual	60453300
CYBER 70 Model 72 Computer System Hardware Reference Manual	60347000
CYBER 70 Model 73 Computer System Hardware Reference Manual	60347200
CYBER 70 Model 74 Computer System Hardware Reference Manual	60347400
CYBER 170 Computer Systems Models 171 through 175 (Levels A, B, C) Model 176 (Level A) Hardware Reference Manual	60420000

<u>Control Data Publication</u>	<u>Publication Number</u>
CYBER 170 Computer Systems Models 720, 730, 740, 750, and 760, Model 176 (Level B/C) Hardware Reference Manual	60456100
CYBER 170 Computer System Model 825 Hardware Reference Manual	60469350
CYBER 170 Computer Systems Models 835 and 855 Hardware Reference Manual	60469290
Data Base Utilities Version 1 Reference Manual	60498800
Data Catalogue 2 Reference Manual	60483200
DDL Version 3 Reference Manual, Volume 1	60481900
DDL Version 3 Reference Manual, Volume 2	60482000
DDL Version 3 Reference Manual, Volume 3	60482100
FORM Version 1 Reference Manual	60496200
FORTRAN Common Library Mathematical Routines Reference Manual	60498200
FORTRAN Data Base Facility Version 1 Reference Manual	60482200
FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN Extended Version 4 to FORTRAN Version 5 Conversion Aids Program Version 1 Reference Manual	60483000
FORTRAN Version 5 Common Library Mathematical Routines Reference Manual	60483100
FORTRAN Version 5 Reference Manual	60481300
General Purpose Simulation System V (GPSS) General Information Manual	84003900
Information Management Facility Version 1 Application Programming Reference Manual	60484500
Message Control System Version 1 Reference Manual	60480300
Modify Version 1 Instant Manual	60450200
Modify Version 1 Reference Manual	60450100

<u>Control Data Publication</u>	<u>Publication Number</u>
Network Access Method Version 1 Network Definition Language Reference Manual	60480000
Network Access Method Version 1 Reference Manual	60499500
Network Terminal User's Instant	60459380
NOS Version 2 Application Installation Handbook	84002760
NOS Version 2 Applications Programmer's Instant	60459360
NOS Version 2 Diagnostic Index	60459390
NOS Version 2 Installation Handbook	60459320
NOS Version 2 Manual Abstracts	60485500
NOS Version 2 Operator/Analyst Handbook	60459310
NOS Version 2 Reference Set, Volume 1, Introduction to Interactive Usage	60459660
NOS Version 2 Reference Set, Volume 2, Guide to System Usage	60459670
NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NOS Version 2 System Maintenance Reference Manual	60459300
NOS Version 2 System Overview	60459270
NOS Version 2 Systems Programmer's Instant	60459370
On-Line Maintenance Software Reference Manual	60454200
PERT/Time Version 2 Reference Manual	60456030
PL/I Version 1 Reference Manual	60388100
Query Update Version 3 Reference Manual	60498300
Remote Batch Facility Version 1 Reference Manual	60499600
SIMSCRIPT Version 3 Reference Manual	60358500
Software Publications Release History	60481010
Sort/Merge Versions 4 and 1 Reference Manual	60497500
Stimulator Version 1 Reference Manual	60480500
SYMPL Version 1 Reference Manual	60496400
TAF/CRM Data Manager Version 1 Reference Manual	60459510
TAF Version 1 Reference Manual	60459500
TAF Version 1 User's Guide	60459520

<u>Control Data Publication</u>	<u>Publication Number</u>
Text Editor Version 1 Reference Manual	60436100
TOTAL-CDC Version 2 Reference Manual	76070300
Update Version 1 Reference Manual	60449900
XEDIT Version 3 Reference Manual	60455730
6000 Series Computer Systems Hardware Reference Manual	60100000
8-Bit Subroutines Reference Manual	60495500

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 undefined parameters.

5. COMMAND PROCESSING	5-1	LIMITS Command	7-19
Command Format	5-2	Resource Limits	7-20
Command Processing Flow	5-5	Other Characteristics	7-20
Exit Processing	5-7	User Permissions	7-21
		Network Applications	7-21
		Privileged Network Users	7-21
		Local Applications	7-22
6. EXECUTION CONTROL COMMANDS	6-1	MACHINE Command	7-23
Command Syntax	6-2	MFL Command	7-23
Operators	6-2	NORERUN Command	7-24
Arithmetic Operators	6-2	NOTE Command	7-24
Relational Operators	6-3	OFFSW Command	7-25
Logical Operators	6-4	ONSW Command	7-25
Order of Evaluation	6-4	PASSWOR Command	7-26
Operands	6-4	PAUSE Command	7-26
Constants	6-5	PROTECT Command	7-27
Numeric Strings	6-5	QGET Command	7-27
Literals	6-5	RERUN Command	7-28
Symbolic Names	6-6	RESOURC Command	7-28
Functions	6-10	Deadlock Prevention	7-29
FILE Function	6-10	Single Resource Use	7-30
DT Function	6-12	Tape Units	7-30
NUM Function	6-13	Resource Overcommitment	7-31
Command Descriptions	6-14	Altering Resource Requirements	7-32
BEGIN Command	6-14	Unit Assignment	7-33
DISPLAY Command	6-14	RFL Command	7-33
ELSE Command	6-15	ROLLOUT Command	7-34
ENDIF Command	6-16	RTIME Command	7-34
ENDW Command	6-16	SETASL Command	7-35
EXIT Command	6-17	SETCORE Command	7-35
IFE Command	6-17	SETJOB Command	7-36
MODE Command	6-19	SETJSL Command	7-37
NOEXIT Command	6-21	SETPR Command	7-38
ONEXIT Command	6-21	SETTL Command	7-38
REVERT Command	6-21	STIME Command	7-39
SET Command	6-22	SUBMIT Command	7-40
SKIP Command	6-27	SUMMARY Command	7-44
WHILE Command	6-27	SWITCH Command	7-45
		UPROC Command	7-45
		USECPU Command	7-46
		USER Command	7-46
7. JOB CONTROL COMMANDS	7-1	8. COMMANDS FOR INTERACTIVE JOBS	8-1
CFO Command	7-2	Terminal Control Commands	8-2
CHARGE Command	7-2	ASCII Command	8-2
COMMENT Command	7-3	AUTO Command	8-2
CTIME Command	7-3	BRIEF Command	8-4
DAYFILE Command	7-4	CSET Command	8-4
DROP Command	7-6	NORMAL Command	8-4
ENQUIRE Command	7-7	TIMEOUT Command	8-5
ENTER Command	7-13	TRMDEF Command	8-5
GO Command	7-13	Subsystem Selection Commands	8-7
HTIME Command	7-14	ACCESS Command	8-8
Job Command	7-15	BASIC Command	8-8
LDI Command	7-17	BATCH Command	8-9
LENGTH Command	7-18	EXECUTE Command	8-9

FORTRAN Command	8-10	DOCUMENT Command	9-19
FTNNTS Command	8-11	EVICT Command	9-21
NULL Command	8-11	FCOPY Command	9-22
Interactive Status Commands	8-12	LIST80 Command	9-24
Detach Command (cD)	8-12	LOCK Command	9-24
Immediate Job Status Command (cE)	8-13	LO72 Command	9-25
Abbreviated Job Status Command (cS)	8-13	NEW Command	9-29
Job Processing Commands	8-13	OUT Command	9-29
BYE Command	8-13	PACK Command	9-30
DIAL Command	8-15	PRIMARY Command	9-30
GOODBYE Command	8-15	RENAME Command	9-31
HELLO Command	8-15	REQUEST Command	9-32
HELP Command	8-16	RESEQ Command	9-34
LIB Command	8-16	RETURN Command	9-35
LIST Command	8-18	REWIND Command	9-36
LOGIN Command	8-18	ROUTE Command	9-37
LOGOUT Command	8-18	Parameter Descriptions	9-37
NOSORT Command	8-18	Implicit Routing	9-42
RECOVER Command	8-19	SCOPY Command	9-43
RUN Command	8-23	SETFS Command	9-46
TEXT Command	8-24	SKIPEI Command	9-46
WHATJSN Command	8-24	SKIPF Command	9-47
X Command	8-25	SKIPFB Command	9-47
Primary File Editing Commands	8-26	SKIPR Command	9-48
Parameter Format	8-26	SORT Command	9-48
Suppressing Editing Responses	8-28	TCOPY Command	9-49
Line Number Overlap	8-28	TDUMP Command	9-52
Inserting Lines	8-29	UNLOAD Command	9-54
ALTER Command	8-29	UNLOCK Command	9-55
DELETE Command	8-31	VERIFY Command	9-55
DUP Command	8-32	WRITEF Command	9-58
LIST Command	8-33	WRITER Command	9-58
MOVE Command	8-35		
READ Command	8-36		
RESEQ Command	8-38		
WRITE Command	8-38		
WRITEN Command	8-39		
9. FILE MANAGEMENT COMMANDS	9-1	10. PERMANENT FILE COMMANDS	10-1
ASSIGN Command	9-1	Common Parameters	10-2
BKSP Command	9-3	APPEND Command	10-6
CLEAR Command	9-4	ATTACH Command	10-6
CONVERT Command	9-4	CATLIST Command	10-9
COPY Command	9-6	CHANGE Command	10-14
Copy Termination	9-10	COMMON Command	10-15
Block Sizes	9-10	DEFINE Command	10-15
Processing Options	9-10	GET Command	10-18
COPYBF Command	9-11	OLD Command	10-19
COPYBR Command	9-12	PACKNAM Command	10-20
COPYCF Command	9-13	PERMIT Command	10-21
COPYCR Command	9-15	PURGALL Command	10-22
COPYEI Command	9-16	PURGE Command	10-23
COPYSBF Command	9-17	REPLACE Command	10-24
COPYX Command	9-18	SAVE Command	10-25
		11. LOAD/DUMP MEMORY COMMANDS	11-1
		DMB Command	11-1
		DMD Command	11-4

DMDECS Command	11-4	CATALOG Command	16-4
DMP Command	11-5	COPYL and COPYLM Commands	16-7
DMPECS Command	11-6	GTR Command	16-11
LBC Command	11-7	ITEMIZE Command	16-14
LOC Command	11-7	LIBEDIT Command	16-17
PBC Command	11-8	Command Format	16-18
RBR Command	11-8	LIBEDIT Directives	16-20
WBR Command	11-9	ADD Directive	16-23
		BEFORE Directive	16-24
		BUILD Directive	16-24
		COMMENT Directive	16-25
		COPY Directive	16-25
		DATE Directive	16-25
		DELETE Directive	16-26
		FILE Directive	16-26
		IGNORE Directive	16-27
		INSERT or AFTER Directive	16-27
		LIBGEN Directive	16-28
		LIST Directive	16-28
		NEW Directive	16-28
		NOINS Directive	16-28
		NOREP Directive	16-29
		NOREW Directive	16-29
		OLD Directive	16-29
		RENAME Directive	16-29
		REPLACE Directive	16-30
		REWIND Directive	16-30
		TYPE or NAME Directive	16-31
		VFYLIB Directive	16-31
		LIBEDIT Output	16-31
		LIBGEN Command	16-32
		VFYLIB Command	16-34
		Library Processing Examples	16-36
12. TAPE MANAGEMENT	12-1		
Tape Assignment	12-1		
Command Rules	12-2		
Processing Options	12-3		
ASSIGN Command	12-5		
BLANK Command	12-8		
LABEL Command	12-11		
LISTLB Command	12-19		
REQUEST Command	12-21		
VSN Command	12-24		
13. CHECKPOINT/RESTART	13-1		
CKP Command	13-1		
RESTART Command	13-2		
14. DEBUGGING AIDS	14-1		
Exchange Package Dumps	14-1		
Using Dumps	14-5		
15. SYSTEM UTILITY COMMANDS	15-1		
EDIT Command	15-1		
KRONREF Command	15-2		
MODIFY Command	15-3		
OPLEDIT Command	15-6		
PROFILE Command	15-8		
UPDATE Command	15-10		
XEDIT Command	15-14		
16. LIBRARY MAINTENANCE	16-1		
Library File Access Methods	16-1		
Library Record Types	16-2		
		17. TERMINAL INPUT/OUTPUT	17-1
		Terminals	17-1
		Input/Output Conventions	17-2
		Input From Terminal	17-2
		Input Line Delimiters	17-3
		Length of Input/Output Lines	17-3
		Network Messages During Input	17-3
		Terminating Input Line	17-4
		Correcting Input Line	17-4
		Deleting Input Line	17-5
		Entering Input Before Prompts	17-5
		Page Wait	17-6
		Suspending Output	17-6
		Aborting Output Line	17-6
		Interrupting Job Step	17-7
		Terminating Job Step	17-7
		Job Suspension	17-8
		Using ATTN on IBM 2741 Terminals	17-9
		Control Bytes	17-10

APPENDIXES

A. CHARACTER SETS	A-1	H. CONTROL LANGUAGE (KCL)	H-1
Character Set Anomalies	A-2	Expressions	H-1
Character Set Tables	A-2	Operators	H-1
Interactive Jobs	A-2	Functions	H-1
Batch Jobs	A-3	Symbolic Names	H-2
Jobs Using Line Printers	A-3	Evaluation of Expressions	H-3
Jobs Using Magnetic Tapes	A-11	Control Language Commands	H-4
		GOTO Command	H-4
		CALL Command	H-5
		DISPLAY Command	H-6
		SET Command	H-7
		IF Command	H-8
B. DIAGNOSTIC MESSAGES	B-1	Control Language Functions	H-9
		FILE Function	H-9
		NUM Function	H-11
C. GLOSSARY	C-1	Procedure Files	H-12
D. SAMPLE JOB OUTPUT	D-1		
		I. LINE PRINTER CARRIAGE CONTROL	I-1
E. TERMINAL CHARACTER CONVERSION	E-1	Printed Data	I-1
Data Input	E-1	Carriage Control	I-1
Data Output	E-2	Carriage Control Using Format	
		Channel Selection	I-3
		Carriage Control Arrays for 580	
F. CARD FILE DATA CONVERSION	F-1	PFC Printers	I-5
Input Card File Formats	F-2	PFC Array Syntax	I-5
Coded Cards	F-3		
Conversion Modes	F-3	J. OBSOLETE TAPE FORMATS	J-1
Literal Input	F-3	B (Blocked) Format	J-1
Binary Cards	F-5	E (Line Image) Format	J-2
Summary	F-5	X (External) Format	J-3
Punch File Formats	F-5	End-Of-Tape/End-Of-Reel Conditions	J-3
Coded Cards (Punch)	F-6		
Binary Cards (Punch B)	F-6		
Absolute Binary Cards (P8)	F-7		
		K. TERMINAL DEFINITION COMMANDS	K-1
		Introduction	K-1
G. ANSI TAPE LABEL FORMATS	G-1	Terminal Definition Command Format	K-1
Required Labels	G-1	Terminal Definition Parameters	K-2
VOL 1 - Volume Header Label	G-2	Terminal Class (TC)	K-2
HDR1 - First File Header Label	G-4	Control Characters (CT)	K-3
EOF1 - First End-Of-File Label	G-8	Backspace Characters (BS)	K-3
EOV1 - First End-Of-Volume Label	G-9	Cancel Characters (CN)	K-4
Optional Labels	G-10	Abort Output Line Character (AL)	K-4
HDR2 through HDR9 - Additional		Interruption Character (BI)	K-4
File Header Labels	G-10	Termination Character (B2)	K-4
EOF2 through EOF9 - Additional		Carriage Return Idle Count (CI)	K-4
End-Of-File Labels	G-10	Line Feed Idle Count (LI)	K-5
EOV2 through EOV9 - Additional		Page Width (PW)	K-5
End-Of-Volume Labels	G-10	Page Length (PL)	K-5
User Labels	G-11	Page Wait (PG)	K-5
		Parity Selection (PA)	K-6

Special Editing (SE)	K-6	Order-Dependent Parameter	
Transparent Input Mode		Matching Mode	M-2
Delimiter (DL)	K-6	Order-Independent Parameter	
Input Device (IN)	K-7	Matching Mode	M-6
Output Device (OP)	K-7		
Echoplex Mode (EP)	K-7		
Message Command (MS)	K-7	N. PAPER TAPE OPERATIONS	N-1
Character Code Set Command	K-8	General Description	N-1
		Teletypewriter	N-1
		Control Characters	N-2
L. LINE FEED AND TRANSMISSION KEYS	L-1	Input Lines	N-2
		Punching Tape Off-Line	N-3
		Tape Mode	N-3
M. PASSIVE PROCEDURES	M-1	Punching a Tape On-Line	N-4
		Corrections	N-4
.PROC Directive	M-1		
Procedure and Parameter Descriptions	M-2	O. MASS STORAGE DEVICE STATISTICS	O-1
Parameter Matching	M-2		

INDEX

FIGURES

1-1 Central Memory Allocation	1-3	14-4 Example 1: Partial Load Map	14-7
2-1 Logical Structure of Files	2-2	14-5 Example 1: Dayfile from a	
2-2 NOS and CRM File Terminology	2-3	Job Run	14-7
2-3 Sample Card File Structure	2-6	14-6 Example 1: Exchange Package	
2-4 Use of ANSI Labels	2-8	Dump	14-9
3-1 FORTRAN Compile and Execute		14-7 Example 2: Central Memory Dump	14-9
Deck	3-2	16-1 Random Access File Structure	16-2
4-1 Calling a Procedure	4-19	16-2 LIBEDIT Input and Output	16-18
5-1 Command Processing Flow	5-6	16-3 User Library Structure	16-33
7-1 Resource Commitment Processing		F-1 Examples of Coded Card	
(Simplified)	7-29	Conversion	F-4
14-1 Exchange Package Dump	14-2	I-1 Carriage Control Tape Format	I-4
14-2 Exchange Package Dump for		M-1 Keyword Substitution in Two	
Model 176	14-2	Procedures	M-5
14-3 Example 1: Program Listing		M-2 Keyword Substitution in Nested	
and Symbolic Reference Map	14-6	Procedures	M-9

TABLES

2-1 Physical File Structure on Different Devices	2-5	A-4 Nine-Track ASCII Coded Tape Conversion	A-12
9-1 Range of Permissible Formats for the COPY Command	9-6	A-5 Nine-Track EBCDIC Coded Tape Conversion	A-13
9-2 Compatible File Structures for the VERIFY Command	9-57	A-6 Seven-Track Coded Tape Conversions	A-14
10-1 Access Mode Granted When Attaching a Currently Attached Direct Access File	10-9	I-1 Carriage Control Characters	I-2
A-1 Character Sets for Interactive Jobs	A-4	K-1 Terminal Definition Parameters	K-9
A-2 Character Sets for Batch Jobs	A-6	L-1 Terminal Key Equivalences	L-1
A-3 ASCII to 6/12 Display Code Conversion	A-9	M-1 Parameter Substitution in Order-Dependent Mode	M-4
		M-2 Parameter Substitution in Order-Independent Mode	M-8

SYSTEM DESCRIPTION

1

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

- Local batch.
- Remote batch.
- Transaction.
- Interactive.

The network processing modes (remote batch, transaction, and interactive) operate through the Network Access Method (NAM) communications software. These processing modes are implemented, respectively, by the following NAM applications: Remote Batch Facility (RBF), Transaction Facility (TAF), and Interactive Facility (IAF).

The primary emphasis of this manual is interactive and local batch processing. For the other processing modes, consult the appropriate manuals listed in the preface.

NOS, like all operating systems, is the interface between user software and the capabilities of system hardware components. The remainder of this section describes the hardware and software that make up a NOS-controlled computer system. In most cases, you need not understand the operation of system hardware or the internal operation of system software. This manual describes these topics only as general background for understanding NOS commands.

SYSTEM HARDWARE

NOS can operate within the CYBER 170, CYBER 70, and 6000 Computer Systems (refer to the preface for an exhaustive list of model numbers). The primary hardware components of each system are the following.

- Central processor unit(s).
- Central memory.
- Extended memory (optional).
- Peripheral processors.
- Peripheral equipment.

CENTRAL PROCESSOR UNIT

The central processor unit (CPU) executes instructions and manipulates and stores data retrieved from central memory. The number and type of CPUs within a mainframe vary with the machine model. As a result, some models can execute additional instructions. These model differences do not affect applications written in higher level languages.

CENTRAL MEMORY

The primary functions of central memory (CM) are:

- To buffer data to and from the peripheral processors.
- To transfer instructions and data to and from the CPU.

Job Field Length

The job field length is the portion of central memory that is assigned to your job. Several jobs can reside in CM simultaneously. The field length separates each job while it resides in CM. The job is assigned a starting CM address (its reference address or RA) and is allocated an initial field length (the CM words in which the job resides and executes). The field length is adjusted during job execution as described in section 3. Figure 1-1 shows a job field length within CM.

A reference to an address outside the job's field length range causes a hardware error condition and job termination.

The maximum field length depends on the CM size and installation parameters used to control memory usage. The system assigns the CPU to jobs requiring CPU activity. Rapid switching of CPU control between jobs enables them to execute concurrently. The exact amount of time allowed for each job depends on system activity and system parameter settings. Thus, the time required to complete a job may vary from run to run, although the actual CPU execution time is similar.

When a job completes, aborts, or rolls out (that is, its execution is suspended), the field length is released, cleared, and made available to another job.

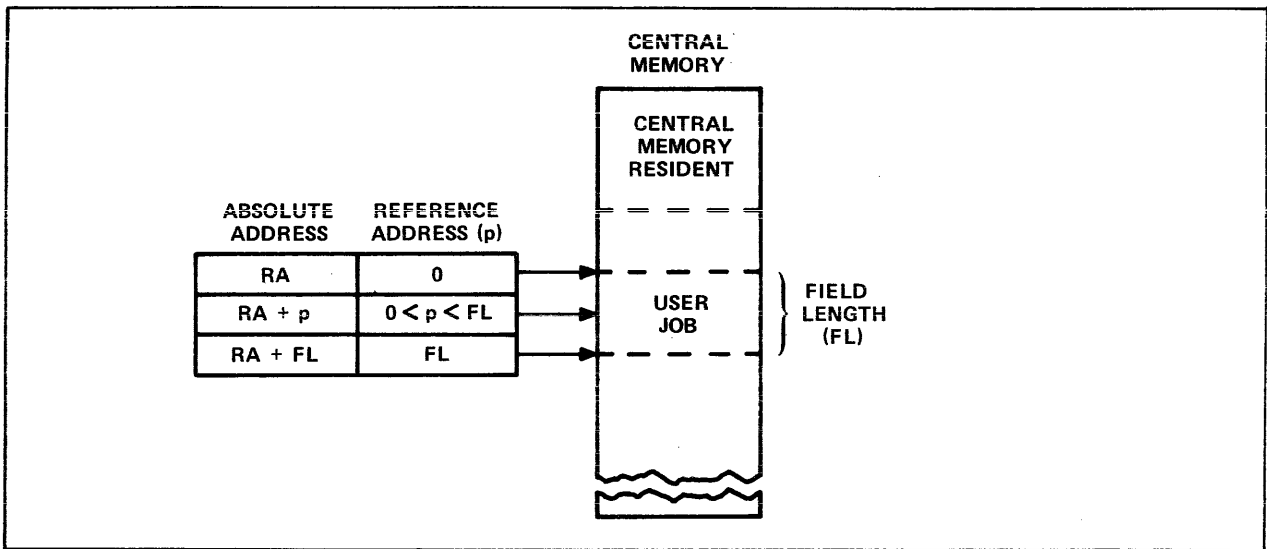


Figure 1-1. Central Memory Allocation

Central Memory Resident

The portion of CM reserved for system use is called central memory resident (CMR). It contains system tables, directories, and the CM portion of the system monitor (CPUMTR).

EXTENDED MEMORY

Extended memory for NOS computer systems (refer to preface), can be used for the following purposes.

- As a directly accessible memory device via FORTRAN or COMPASS statements for extended memory data storage and retrieval (refer to the FORTRAN 5 Reference Manual or appendix D of Volume 4, Program Interface).
- As storage for frequently accessed small files (refer to ASSIGN command in section 7 and Permanent File Commands in section 9).
- As an alternate system device for storing copies of frequently used routines.
- As a link between mainframes in a multimainframe configuration.†

You must be authorized to use extended memory (refer to LIMITS command in section 7).

† Models 176, 825, 835, and 855 cannot use extended memory to link mainframes.

PERIPHERAL PROCESSORS

The peripheral processors (PPs) process communications between CM and individual peripheral devices. They also perform system control functions. A peripheral processor can:

- Read and write CM.
- Read and write extended memory indirectly via CM or directly via the distributive data path (DDP).†
- Transfer data to and from peripheral devices through the data channels.

NOS supports a variety of PP configurations. Each computer system supports a different range of configurations. To determine the configurations supported by your computer system consult the appropriate system hardware reference manual listed in the preface.

PERIPHERAL EQUIPMENT

Peripheral equipment varies among installations but usually includes card readers and punches, line printers, mass storage devices, and magnetic tape units. NOS supports the following equipment models.

405 Card Reader

415 Card Punch

580-12, 580-16, and 580-20 Line Printers

844-21 Disk Storage Subsystem

844-41 and 844-44 Disk Storage Subsystems

885 Disk Storage Subsystem

819 Disk Storage Subsystem ††

Mass Storage Facility (MSF)

667, 669, 677, and 679 Magnetic Tape Units

255x Network Processing Units

†DDP does not apply to model 176, 825, 835, or 855 peripheral processors.

††Model 176 only.

SYSTEM SOFTWARE

Software executed within a computer system can be divided between software that is built into the system during system initialization and software that executes as jobs within the running system. Software present when the system begins running includes the operating system and products such as compilers, CYBER Loader, and CYBER Record Manager. Jobs run within the system are categorized according to their origin as described in section 3.

USER PROGRAMS

A user program is a group of CPU instructions that perform a certain task or calculate a result. A user program can be written in a language at any of three levels.

- Compiler languages provide you with a language suited to your particular needs. The program statements are translated by the appropriate compiler (FORTRAN, COBOL, ALGOL, and so on), which generates assembler language or machine language instructions. Programs written in compiler languages may be machine-independent.
- Assembler languages provide a one-to-one relationship between instructions and machine operation. Mnemonics are provided for each instruction. These languages are machine-dependent. Much of the NOS system is written in COMPASS, the assembler language of the CYBER 170, CYBER 70, and 6000 Computer Systems.
- Hardware instructions are interpreted directly by the computer. Each hardware instruction is a binary number. You are rarely concerned with instructions written at this level except when program debugging requires that you interpret memory dumps.

OPERATING SYSTEM

NOS is a group of programs that supervise and coordinate the operation of system hardware and the execution of products and user programs. The following lists some of the functions of NOS.

- Loading and scheduling of user programs.
- Job validation and accounting.
- Command translation.
- File retrieval, manipulation, routing, and storage.
- Job input and output.
- Normal and abnormal job termination.
- Memory dumps.

CYBER Loader

CYBER Loader prepares programs for execution. Following your directions, it allocates memory for a program, loads the program modules into their appropriate locations, generates a load map, and initiates program execution. It can load subdivided programs for more efficient use of memory. Refer to the CYBER Loader Reference Manual for more information.

CYBER Record Manager

CYBER Record Manager (CRM) is the interface between program input/output (I/O) functions and NOS physical I/O functions. NOS commands do not use CRM. Some of the products that use CRM are COBOL 5, FORTRAN Extended 4, FORTRAN 5, Sort/Merge 5, PL/I (Programming Language I), and DMS-170.

The functions of CRM are divided between the Basic Access Methods (BAM) and Advanced Access Methods (AAM). BAM handles sequential and word-addressable file organizations; AAM handles indexed sequential, direct access, and actual key file organizations. Refer to the appropriate CYBER Record Manager manual listed in the preface.

A file is a collection of information addressable by name. All NOS data processing involves operations performed on files. Files can be differentiated by their name, structure, or file type or by whether they are assigned to a job (NOS jobs are described in section 3).

FILE NAMES

A file name consists of unique string of seven or less alphanumeric characters. Although NOS allows file names that begin with a digit, many of its products (FORTRAN 5 and COBOL 5, for example) do not support such file names. Also, COBOL 5 supports only six-character file names. Refer to the product reference manual listed in the preface for details.

Examples:

A A1 TAPE DUMP12 COMPILE

NOS uses many internal scratch files. Most of these scratch files have names beginning with the letters ZZ or SCR. Avoid using a name starting with these letters for any of your files.

The following file names are significant because they are associated with system-managed files that are assigned to your job.

INPUT OUTPUT PUNCH PUNCHB P8

Refer to the description of File Types for more information.

FILE STRUCTURE

A file consists of a well-defined set of data. Its physical representation varies markedly with the device upon which it resides. For instance, the physical structure of a file on magnetic tape is only remotely similar to its physical structure on disk. To facilitate the manipulation of file data by user and system programs, the system imposes a logical structure on a file that does not vary with device residence.

A logical file consists of a contiguous series of data (physically, the data is not always contiguous). A file always has a beginning of information and an end of information. It can be subdivided into a maximum of three levels. Figure 2-1 illustrates the structure of a file with one, two, and three levels of subdivisions. NOS and its products (refer to section 1) all use this basic structure for a logical file.

NOS and some of its products do differ in both the levels of file subdivision they recognize and the names of the subdivisions. One of the most prominent of these products is CYBER Record Manager (CRM). It handles input and output for some widely used products like FORTRAN 5 and COBOL 5. The next two subsections contrast the treatment that NOS and CRM give a logical file.

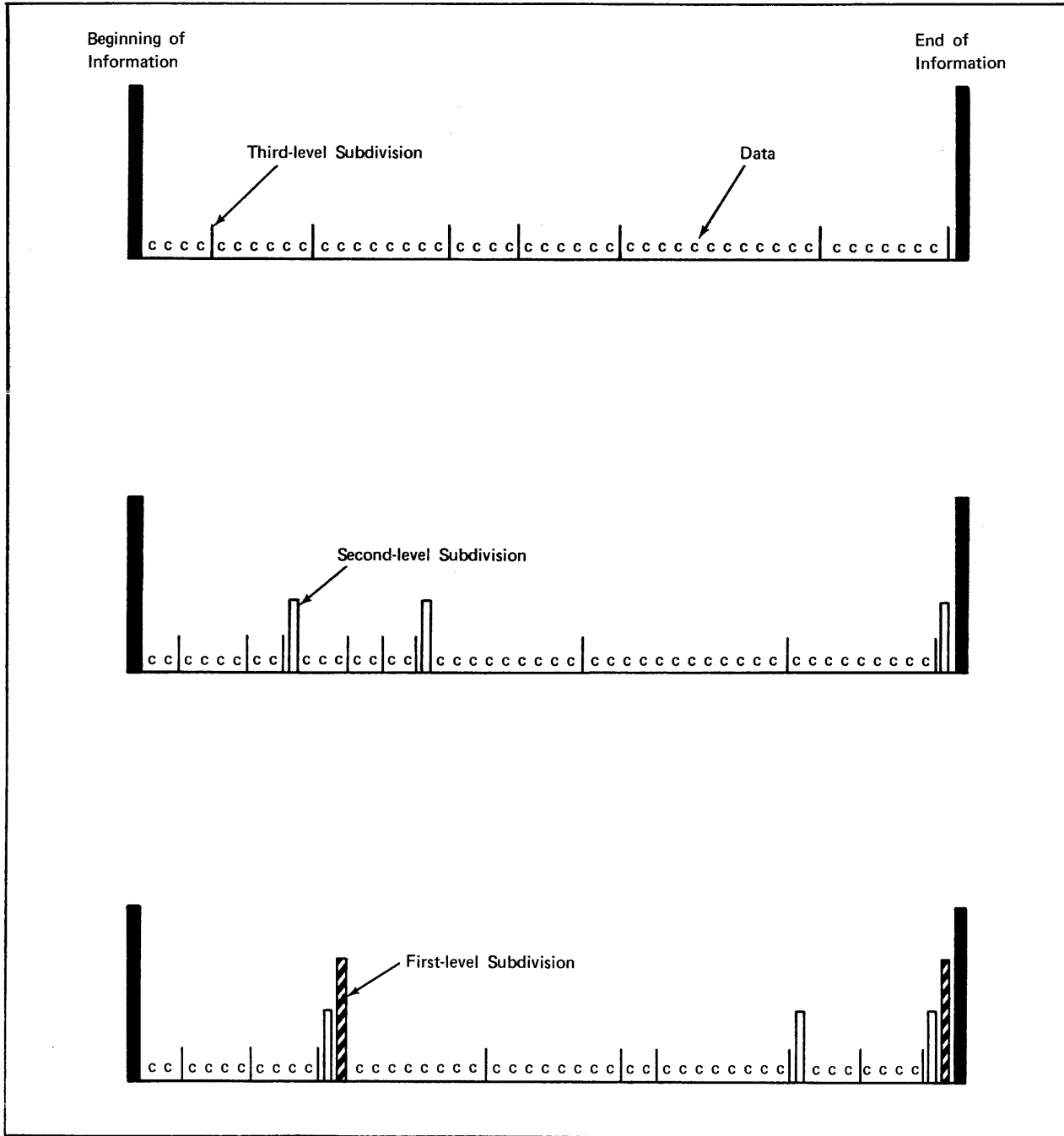


Figure 2-1. Logical Structure of Files

CYBER RECORD MANAGER FILE STRUCTURE

CRM recognizes all three levels of file subdivision. It calls a first-level subdivision a partition, a second-level subdivision a section, and a third-level subdivision a record. Through CRM, you can specify a file organization, a blocking type, and a record type for your data. The file organization determines how records are accessed, the blocking type determines how CRM records are grouped on their storage media, and the record type defines the smallest unit of data CRM can retrieve. Refer to the CRM manuals listed in the preface for a detailed description.

NOS FILE STRUCTURE

NOS recognizes only two levels of file subdivision, the first and second levels.[†] NOS calls a first-level subdivision a file and a second-level subdivision a record. The record is the smallest recognizable unit for both CRM and NOS but the term does not always refer to the same subdivision for the two processors. Figure 2-2 illustrates the differences in terminology.

file processor	first-level subdivision	second-level subdivision	third-level subdivision
NOS	file	record	-----
CRM	partition	section	record

Figure 2-2. NOS and CRM File Terminology

Since NOS terms a first level subdivision a file, a NOS file can contain more than one logical file; if it does, it is called a multifile file. A multifile file begins at beginning-of-information (BOI) and ends at end-of-information (EOI). A file within a multifile file begins either at BOI or after the end-of-file (EOF) of the preceding file. It ends at its EOF.

Each file consists of zero or more logical records of information. A record is zero or more 60-bit CM words. A record begins at the BOI, after an EOF, or after the end-of-record (EOR) of the preceding record. It ends at its EOR. The following is the structure of a single-record file.

BOI data EOR EOF EOI

or

BOI data EOR EOI

Most simple files, like text files, have one of these two formats.

[†]This is true for NOS commands but, at the program level, NOS does recognize third-level subdivisions.

The following is the structure of a multirecord, multifile file.

BOI data EOR data EOR EOF data EOR data EOR EOF EOI

The last EOF in a file may or may not be present depending upon the program used to create the file.

PHYSICAL FILE STRUCTURE

When NOS stores a file, it automatically converts the file to a structure that will conform to the physical characteristics of the storage medium. The logical file and record marks are converted to physical BOI, EOR, EOF, and EOI indicators.

The basis of all physical file structures is the physical record unit (PRU), the minimum amount of data that can be read or written in a single device access. Table 2-1 lists the PRU size, and record and file mark indicators for each supported storage device.

Table 2-1. Physical File Structure On Different Devices

Device		PRU Size	Record and File Mark Indicators			
			BOI	EOR	EOF	EOI
Magnetic disk or extended memory		64 CM words.	PRU 1 or random address 1 of the disk (counting from 0).	PRU of less than 64 words with a link to the next PRU.	Zero-length PRU (no data) with special link to next PRU.	Zero-length PRU with no forward link.
Card decks†		One card.	First card in the deck.	Card with a 7/8/9 punch in column 1. Remote Batch Facility (RBF)/HASP can also use /*EOR.	Card with 6/7/9 punch in column 1.††	Card with 6/7/8/9 punch in column 1. RBF/HASP can also use /*EOI.
Interactive Terminal		One line.	None.	On input, one nonnull line of data with EOR level 1.	On input, one null line of input.	None.
Magnetic tape†††	I (Internal)	512 CM words; each PRU includes a 48-bit terminator.	If labeled, tape mark following HDRI label. If unlabeled, load point.†††	A PRU of less than 512 words with level number of 0.	Zero-length PRU whose terminator contains a level number of 17g.	Tape mark followed by an EOF1 label.
	SI (System internal)	512 CM words; each PRU of less than 512 words has a 48-bit terminator.	If labeled, tape mark following HDRI label. If unlabeled, load point.†††	A PRU of less than 512 words with level number between 0 and 16g.	Zero-length PRU whose terminator contains a level number of 17g.	Tape mark followed by an EOF1 label.
	S (Stranger)	Maximum of 512 CM words; (refer to BS parameter on COPY command in section 9 and to appendix J in volume 4).	If labeled, tape mark following HDRI label. If unlabeled, load point.†††	End of each PRU.	Tape mark.††††	If labeled, a tape mark followed by an EOF1 label. If unlabeled, there is no EOI indicator.
	L (Long block stranger)	No maximum defined (refer to BS parameter on COPY command in section 9 and to appendix J in volume 4).	If labeled, tape mark following HDRI label. If unlabeled, load point.†††	End of each PRU.	Tape mark.††††	If labeled, a tape mark followed by an EOF1 label. If unlabeled, there is no EOI indicator.
	F (Foreign)	Determined by C or FC parameter on ASSIGN, LABEL, or REQUEST command.	Load point.	None.	Tape mark.	None.

† For more information, refer to appendix F.
†† The 6/7/9 card is not recognized in a remote batch job. In an RBF job the end-of-file marker is a card with a 7/8/9 punch in column 1 and a level number of 17g in columns 2 and 3. RBF/HASP can also use a card with /*EOR in columns 1 through 5 and a level number of 17g in columns 6 and 7.
††† In the case of multivolume tape files, BOI is the HDRI label of the first volume. In the case of an ANSI multifile, BOI is the HDRI label associated with the beginning of the multifile.
†††† For more information, refer to section 10 and appendix G.
††††† On labeled tapes, interchange standards require that tape marks be used only to delimit tape label groups.

Card Files

The physical file and record indicators in a card file are shown in figure 2-3 and listed in table 2-1. Although card decks do not have a defined PRU size, a card is the minimum data unit. NOS can read and punch cards in coded (Hollerith), binary, and absolute binary formats as described in appendix F. Coded cards are punched in 026 or 029 keypunch mode. The system uses the installation default keypunch mode (chosen by the installation) for reading cards unless a 26 or 29 is punched in columns 79 and 80 of a job, EOR, or EOF card, indicating that the subsequent cards are punched in that mode. NOS can punch up to 80 characters on a coded card and up to 150 characters (15 CM words) on a binary card.†

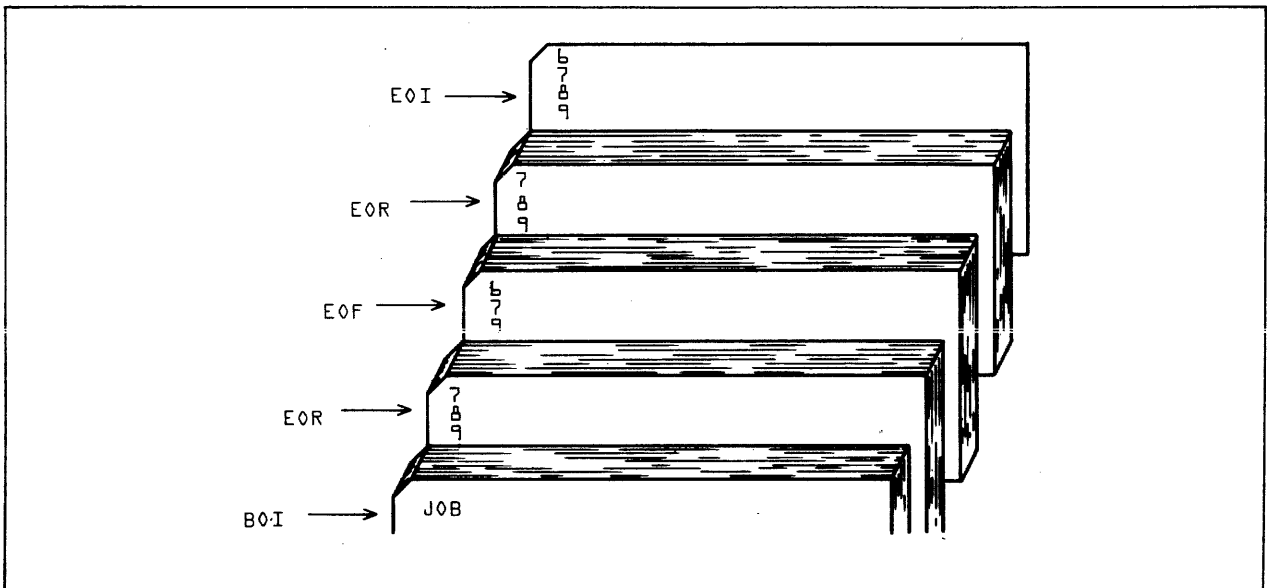


Figure 2-3. Sample Card File Structure

Mass Storage Files

Mass storage files are stored on disk or extended memory.

To use NOS, you do not need to know the physical structure of mass storage; you do need to know its logical structure, which contains the concepts of logical devices and logical tracks. A logical device is one or more physical disk units known to the system as a single device. A logical track is a file allocation unit determined by the device type.

† Keypunch mode selection is not supported for jobs entered through a 200 User Terminal or similar remote batch terminal except for HASP.

Each permanent file on mass storage is accessed via a catalog track containing the permanent file catalog of its owner. Indirect access files (refer to Permanent Files) must reside on the same device as their catalog; direct access files may reside on another device. Space is allocated for mass storage files in units called reservation blocks. An indirect access file reservation block is always 64 words (one PRU). A direct access file reservation block is a logical track. The maximum size of your mass storage file is determined by your validation limits (refer to LIMITS Command in section 6).

Magnetic Tape Files

You may write information on magnetic tapes in either binary or coded mode. You can select the recording mode in various ways. You can select the binary mode by using the COPYBF command, by specifying the CM=NO parameter on the FILE command, by specifying the BUFFER OUT(n,1) statement in a FORTRAN program, or by specifying the RECORDING MODE IS BINARY statement in a COBOL program. You can select the coded mode by using the COPYCF command, by specifying the CM=YES parameter on the FILE command, by specifying the BUFFER OUT (n,0) statement in a FORTRAN program, or by specifying the RECORDING MODE IS CODED statement in a COBOL program.

The representation of data in memory and on tape depends on the number of tracks (seven or nine), the recording mode (binary or coded), and the specification of the CV=cv parameter on the LABEL command. The following shows how the data representation varies with these factors.

<u>Number of Tracks</u>	<u>Binary Mode</u>	<u>Coded Mode</u>
9	<ol style="list-style-type: none"> 1. Odd parity. 2. Groups of four 6-bit characters in memory are written without conversion on three-frame units on tape. 	<ol style="list-style-type: none"> 1. Odd parity. 2. Each 6-bit character in memory is translated into an 8-bit equivalent on tape. You can choose EBCDIC or ASCII by specifying CV=EB or CV=AS on a LABEL command.
7	<ol style="list-style-type: none"> 1. Odd parity. 2. Each 6-bit character in memory is written without conversion. 	<ol style="list-style-type: none"> 1. Even parity. 2. Each 6-bit character in memory is translated to its BCD equivalent on tape. The % character is lost in the translation.

You can select 200- (read only), 556-, or 800-bit-per-inch (bpi) density for seven-track tapes or 800-, 1600-, or 6250-character-per-inch (cpi) density for nine-track tapes, provided these densities are available with the site hardware. NOS automatically processes tape parity errors and end-of-tape conditions unless you select other processing options (refer to Processing Options in section 12). You can also control error processing by using the EO parameter on a FILE command.

Tape Labels

Tape labels identify and delimit tape volumes and tape files. Tape marks begin and end most tape labels. A tape mark is a special bit sequence written and recognized by a tape unit.

NOS processes ANSI standard and nonstandard labeled tapes. Nonstandard labeled tapes are those whose format or content do not conform to the ANSI standard described in appendix G. NOS skips to the first tape mark when reading a nonstandard labeled tape if the tape assignment statement specifies the LB=NS parameter (refer to section 12). All information after the first tape mark is then handled as data.

ANSI standard labels as defined by NOS are those that conform to the American National Standard Magnetic Tape Labels for Information Interchange X3.27-1969 standard. NOS can create or verify ANSI labels if the LABEL command assigns the tape file. Label verification ensures that the correct volume has been mounted. ANSI labels separate multifile set files and indicate if a file continues on another volume.

The ANSI label EOF indicates end-of-information for a file within a file set. The use of ANSI labels to delimit files within file sets is illustrated in figure 2-4.

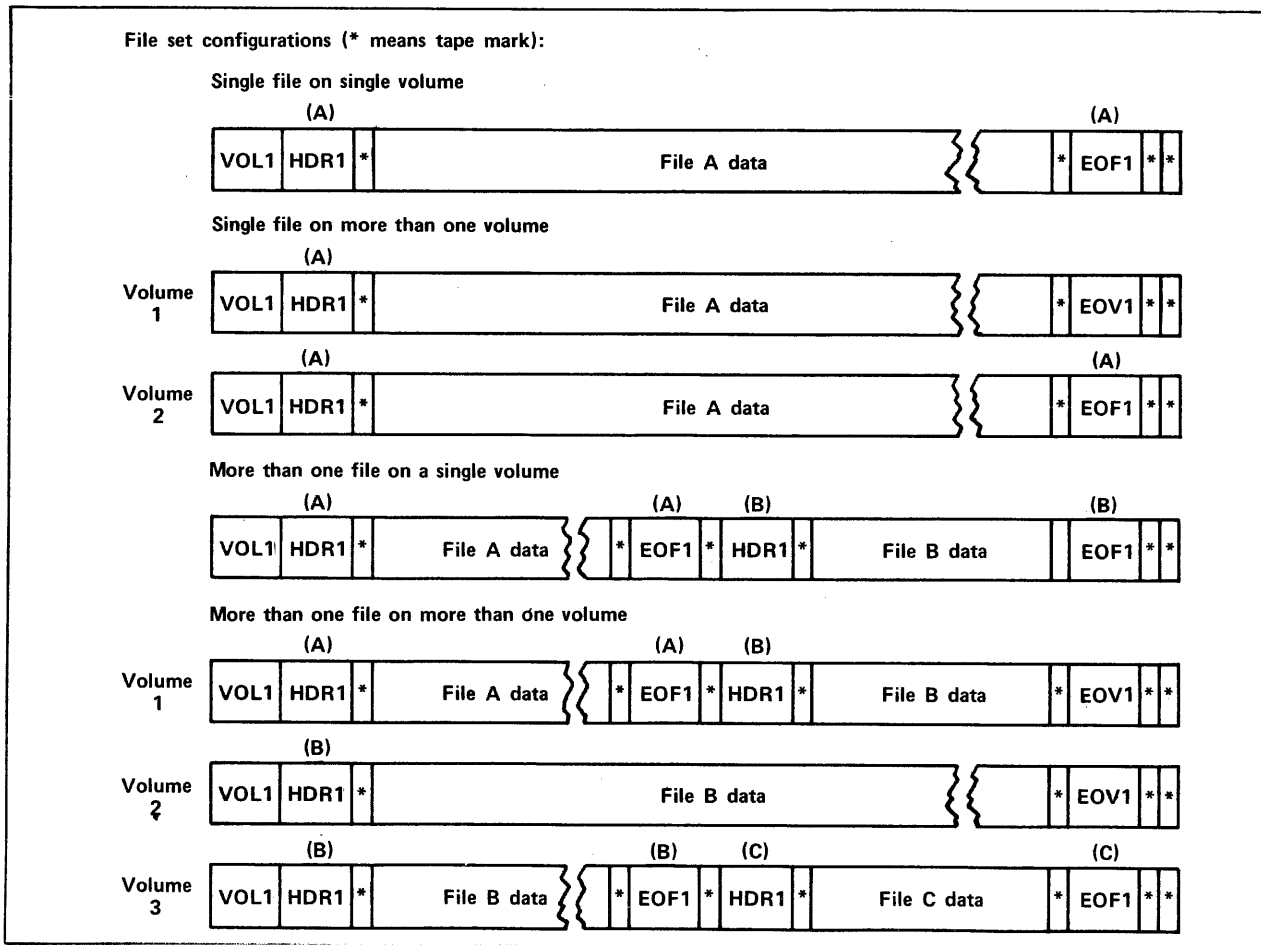


Figure 2-4. Use of ANSI Labels

An ANSI-labeled tape must have the following labels. Other optional labels are described in appendix G.

<u>Label</u>	<u>Location</u>
VOL1	Beginning of volume.
HDRI	Beginning of information. If the file continues on to another volume, the HDRI label is repeated. It must follow the VOL1 label and precede the continuation of the file information.
EOF1	End of information.
EOV1	End of volume (required only if the file continues on another volume).

Appendix G gives the tape label formats.

Tape Data Formats

NOS can read and write data on magnetic tape in any of the following formats.

<u>Format</u>	<u>Mnemonic</u>
Internal (NOS default)	I
System internal†	SI
Stranger	S
Long block stranger	L
Foreign	F

These data formats differ in their PRU (block) size and in their record and file mark indicators (refer to table 2-1). Other format differences are:

<u>Tape Format</u>	<u>Labels</u>	<u>I/O Mode</u>	<u>Tape Parity</u>	<u>Noise Size††</u>
I	Labeled or unlabeled	Binary or coded†††	Odd	Seven-track: <eight frames
			Odd	Nine-track: <six frames
SI	Labeled or unlabeled	Binary only††††	Odd	Seven-track: <eight frames
			Odd	Nine-track: <six frames

† NOS/BE system default tape format (binary mode only).

†† Tape blocks read that are smaller than the noise size are discarded. An attempt to write a block smaller than the noise size produces an error message. Refer to the NS=ns parameter of the LABEL command.

††† Specification of coded mode aborts the job step; refer to TCOPI Command in section 9.

†††† I/O mode bit is ignored.

<u>Tape Format</u>	<u>Labels</u>	<u>I/O Mode</u>	<u>Tape Parity</u>	<u>Noise Size</u>
S	Labeled or unlabeled	Binary Coded 7-track Coded 9-track	Even Odd Odd	User-selected; default is <18 frames
L	Labeled or unlabeled	Binary Coded 7-track Coded 9-track	Odd Even Odd	User-selected; default is <18 frames
F	Unlabeled (labels read as data)	Binary Coded 7-track	Odd Even†	User-selected; default is <18 frames

NOS terminates all blocks on I format types and all blocks that are not a full PRU on SI format tapes with a 48-bit block (PRU) terminator. The terminator contains the total number of bytes in the block (including the terminator itself), the number of blocks since the last HDR1 label, and the level number of the block. This terminator enables read operations on I format tapes to check whether the number of bytes read and the block number expected match the byte count and block number in the terminator. If either does not match, the system attempts to recover the missing data. This feature prevents dropped or fragmented blocks and provides a higher degree of reliability than other data formats.

Tapes should be read with the same format specified as when they were written. Data is then recovered in its original form. For some formats, NOS writes extra bits which are discarded when the tape is read. I format nine-track tapes are always written with an even multiple of 12-bit bytes per block. SI format nine-track tapes may have an extra 4 bits written per block to preserve the lower 4 bits of a CM word. (A 60-bit CM word would be written in eight frames, 8 bits per frame.)

All nine-track tapes are written with odd parity. Binary seven-track tapes have odd parity; coded seven-track tapes have even parity. If a parity error is detected on an F format seven-track tape, the recording mode (binary or coded) is automatically switched to allow reading of mixed mode tapes.

Volume 4, Program Interface, describes tape formats in greater detail.

†I/O mode on write for F format is specified in the control word and is returned in the control word on read.

FILE TYPES

The system maintains several tables for each job. Every file assigned to a job has an entry in one or more of these tables. A table entry for a file includes the file name, the device on which the file resides, file type, and its current position and status.

The system assigns one or more of the following file types to files associated with a job:

- Input file type
- Queued file type
- Local file type
- Primary file type
- Direct access file type
- Library file type

INPUT FILES

An input file is also called a job file because it contains user-supplied commands and data for a job (refer to section 3). Initially, input files exist on mass storage in the input queue. A file enters the input queue directly when a local or remote batch job enters the system or indirectly when a user job submits another job via an LDI, ROUTE, or SUBMIT command. The input file for an interactive job is normally assigned to the terminal and is used by an interactive job to read data entered at the terminal into the program. Commands do not appear in the input file of an interactive job. A user job refers to its input file by the file name INPUT (refer to Input File Control in section 3).

QUEUED FILES

A file is queued as a result of one of the following events:

- Generally, whenever you enter a LDI, OUT, ROUTE, or SUBMIT command. Refer to the description of each individual command for cases where the command fails to queue a file.
- Job termination. The system queues files OUTPUT, PUNCH, PUNCHB, P8, and other system-reserved output files if they are assigned to the job.

A queued file can have one of five dispositions:

- Input
- Plot
- Print
- Punch
- Wait

Your job's origin type determines the default processing of your queued job output. Jobs originating at a central site card reader are queued to a line printer with the same system-defined identifier as the card reader. Similarly, the system queues remote batch output to the remote batch terminal where the job originated. The system gives each remote batch terminal a unique terminal identifier (TID) when it logs in, and uses this TID for queuing files originating from the terminal.

You can override the default queuing of job output with the LDI, QGET, ROUTE, SETJOB, or SUBMIT command (refer to section 7).

Queued files with an input disposition wait until they are scheduled for execution. The service class of the job determines the scheduling priority of an input file.

For queued files with plot, print, or punch dispositions, their priority grows as they wait in the queue. Such files are processed when the appropriate device becomes available and when the file's priority is higher than all other files queued to that device. Refer to the ROUTE command in section 9 for an enumeration of disposition codes for plotting, printing, and punching.

A file with a print disposition must be formatted for line printing. Most system utilities format file OUTPUT for printing, but you must include the appropriate printer control characters when you create a file to route to a printer (refer to appendix I). Appendix D contains the printer output from a sample job.

A queued file with a wait disposition awaits further specifications from you. The system queues output from detached interactive jobs with a wait disposition unless you specify a different disposition. You can explicitly queue files with a wait disposition with the LDI, ROUTE, SETJOB, and SUBMIT commands.

A file remains queued on mass storage until one of the following occurs:

- The system resource it requires becomes available and the file's priority is the highest of all queued files with the same disposition.
- You remove the file with the QGET or DROP command.
- Site personnel remove it if it resides in the queue too long (site-determined).

You can get a list of your queued files with the ENQUIRE command (refer to section 7).

LOCAL FILES

Files of local file type are temporary files. Although all files assigned to a job are considered local to the job, the local file type includes only those files assigned to your job that are not given one of the other file types.

You can create a local file in many ways. The following list shows some implicit methods of creating a local file:

- Name the file for the first time in a copy command or in a read or write command within a program. A local file created in this manner always resides on mass storage.
- Name the file for the first time in an ASSIGN or REQUEST command that assigns the local file to mass storage or to an interactive terminal or name the file in an ASSIGN, LABEL, or REQUEST command that assigns the local file to magnetic tape.

- Name the file in a GET command generating a local mass storage file.
- Name the file as the output file in a compiler-or assembler-call command.
- Name the file as the list or output file in one of the file editing commands (MODIFY or EDIT, for example).

To save the contents of a local mass storage file, you issue a SAVE or REPLACE command to copy the local file to a permanent indirect access file or an APPEND command to copy the local file onto the end of an existing permanent indirect access file. Data written on a local file assigned to magnetic tape is written on the tape for later access. Local files are released upon job completion.

PRIMARY FILES

The primary file is a temporary mass storage file designated as the primary file by a PRIMARY, NEW, or OLD command. Only one primary file can exist for a job at a time. Some commands use the primary file as the default file when a file name is not specified. NOS rewinds the primary file before each job step.

DIRECT ACCESS FILES

You assign a direct access permanent file to your job by issuing an ATTACH or DEFINE command. When you define the file or attach the file in a mode permitting file modification, you can write on the permanent file. Refer to Permanent Files in this section.

LIBRARY FILES

A library file is a read-only file that several users can access simultaneously. This file type should not be confused with system library programs or with public permanent files stored under user name LIBRARY. Refer to Libraries in this section for a description of the uses of the term library in NOS.

You must be authorized to access a library file.

If authorized, you can read a library file after naming it in a COMMON command.

PERMANENT FILES

Your permanent files reside on mass storage until you purge them or until your site archives them (contact your computer center for their archiving procedures). The system maintains a separate permanent file catalog for each user name and it contains entries for all of your permanent files created under the user name you are using to access the system. A catalog entry includes the file name, its location on mass storage, its length, permission modes, and access history. Unless another user name is specified, the system directs all permanent file requests to the files of the user specified on the last USER command (or specified in the login of an interactive job).

User names (refer to Validation in section 3) that contain asterisks represent users with automatic read-only permission to files in the catalogs of other users. The user name must match the other user name in all characters not containing asterisks. For example, the user with user name *AB*DE* can access the catalogs of the following users.

- UABCDEF
- UABDDEE
- MABCDEI

The device residence of permanent files and their backup copies are described under Mass Storage File Residence in this section.

The two types of permanent files, indirect access permanent files and direct access permanent files, are described in the following paragraphs.

INDIRECT ACCESS PERMANENT FILES

You access an indirect access permanent file by naming it in an OLD or GET command. The system copies the permanent file from mass storage to a temporary file (primary or local file type). To alter an indirect access permanent file, you make the changes to the temporary copy and then enter the REPLACE command which writes the temporary copy over the indirect access permanent file. You create an indirect access permanent file by naming a temporary file in a SAVE or REPLACE command.

Mass storage for indirect access permanent files is allocated in 640-character blocks (64 CM words). Because of its small allocation block size, indirect files are most economical for relatively small files.

The maximum size of an indirect access file is determined either by your validation limits (refer to the LIMITS command), or by the device limitations described in Mass Storage Files in this section.

DIRECT ACCESS PERMANENT FILES

You access a direct access permanent file directly, not through a temporary copy. Data is written directly on the permanent file.

You create a direct access permanent file with a DEFINE command, which determines its name and residence and the default access mode available to other users. Other users can access the file with an ATTACH command. A number of users can attach the file concurrently, but only one user at a time can change the file. To change the file, you must attach it in modify, append, or write access mode. If you attach the file in write mode, no other user can attach that file concurrently.

Even if a file is currently attached to a job, you can purge the file from the permanent file catalog with a PURGE command. However, the purged direct access file remains attached to the job until it is released by a RETURN, CLEAR, UNLOAD, OLD, or NEW command or until the job ends.

Mass storage for direct access permanent files is allocated in large blocks; the block size depends on the mass storage device type on which the file resides (refer to Mass Storage Files in this section). Because of their large allocation block size and the write interlock feature, direct access files are often used for database files.

The maximum size of a direct access file is determined by your validation limits (refer to LIMITS Command in section 7), or by the device limitations described in Mass Storage Files in this section.

MASS STORAGE FILE RESIDENCE

For most mass storage file operations, you need not be concerned about the specific device on which your file resides. However, under certain circumstances, you may wish to override the default device residence for local or permanent files.

With the ASSIGN command, you can assign a local file to either a specific device or to a device category if you are so authorized.

Every permanent file you create resides either in your family of permanent file devices, on an auxiliary device, or on the Mass Storage Facility. Unless you specify otherwise, all permanent files are saved in your family. Refer to the NOS Installation Handbook for further details.

FAMILY DEVICES

A family consists of a set of mass storage devices. Normally, a system has only one family (the default family) of permanent file devices. However, because families are interchangeable between NOS systems, several families may be active on one system, or a system may be part of a multimainframe system. For example, consider an installation with two systems, A and B. System B provides backup service to system A. If system A failed, its family of permanent file devices could be introduced into system B without interrupting current operations on system B.

You identify your family by supplying a one- to seven-character family name. You can specify a family name on the USER command. If you are authorized to use secondary USER commands (refer to the USER command), you also specify a family name on a USER command in interactive jobs. If your family is the system's default family, you may, but need not, supply the family name. When the family name is omitted, the system uses the system default family name. If your family has been introduced into another system, you must supply your family name.

If you choose to save files on family devices, you have the option of either using the system default device type or specifying another type of permanent file device.

AUXILIARY DEVICES

An auxiliary device is a supplement to the mass storage provided by family devices. It is identified by a one-to seven-character pack name. An auxiliary device is not necessarily a disk pack that can be physically removed as the pack name implies. Rather, an auxiliary device can be any mass storage device supported by the system and defined as such by the installation. Each auxiliary device is a self-contained permanent file device; all direct and indirect access files represented by the catalogs on the device reside on the device. Auxiliary devices may be defined as public or private. Anyone permitted to use auxiliary devices who supplies the appropriate pack name can create, replace, and access files on a public device. Only one user, the owner, can create and replace files on a private auxiliary device, but others may access or replace those files as permitted by the owner.

MASS STORAGE FACILITY (MSF)

Rotating mass storage is the usual residence of permanent mass storage files. However, if the installation has an MSF, some direct access files can be stored there. An MSF is suited for the storage of large, direct access files that are accessed infrequently. Attaching a file residing on the MSF takes at least 10 seconds, because the file must be retrieved and copied (staged) to disk. You can specify the preferred residence of your direct access files with the PR parameter on the DEFINE or CHANGE commands. You can ascertain the actual residence of your files with the CATLIST(LO=F) command.

Usually, when attaching an MSF file, the system suspends the job until the file has been staged and assigned to the job. (You can determine the status of your MSF file staging request with the ENQUIRE or cE command.) However, if you specify the RT parameter on the ATTACH command, the job continues processing while the MSF file is being staged to disk. You must then issue a second ATTACH command to assign the file to your job after staging. You can then check that the file has been attached by using either the FILE function (refer to section 6), or a LENGTH or ENQUIRE command.

If a permanent file is lost or destroyed, site personnel can recover the file by loading its backup copy. Generally, sites perform regular dumps of permanent files to magnetic tape to serve as the permanent file backup. By specifying the BR parameter on the DEFINE or CHANGE command, you can choose to have a tape backup copy of his direct access file kept even if the file resides on the MSF. You also can choose to have the MSF file copy serve as backup, or you can require no backup copy of your direct access file.

LIBRARIES

As defined in the glossary (appendix C), the term library has several meanings. The applicable meaning for the term must be determined from its context. The following describes some NOS libraries.

USER NAME LIBRARY

Files stored under user name LIBRARY need not be libraries themselves. An installation saves programs or text as files under user name LIBRARY so that authorized users can access them from a centralized location. You can access those files by specifying the file name and the alternate user name LIBRARY on their permanent file request or by issuing a LIB command.

PROGRAM LIBRARIES

A program library is a collection of compressed source deck images stored in Modify or Update format. You access these compressed source decks through MODIFY or UPDATE commands (refer to section 15).

USER LIBRARIES

User libraries are the files named in the LIBRARY, SATISFY, LIBLOAD, or LDSET loader command or in program binaries. These files are searched by CYBER Loader to satisfy external references within a program it is loading. They contain compiled or assembled routines. The first record of a user library is a ULIB record; the last record is an OPLD directory record. Section 16 describes commands that create, catalog, and manipulate user libraries.



The primary job types are batch and interactive. A batch job is a file of commands and data. Its first record, called the command record, contains lines of text that specify NOS commands. Every batch job begins with a Job command and a USER command. The end of the command record is marked by an EOR (or an EOI if there is no data in the job).

If records follow the command record, they contain program, data, or directive input for the commands. As each command requiring user input is processed, the system reads the next record in the input file (unless the command specifies otherwise). These records must be in the same order as the commands that use them.

For example, figure 3-1 illustrates a job file. In this job file, the first three commands are processed by system routines that require no additional user input. The fourth command, FTN5(GO), requests the compilation and execution of a FORTRAN 5 program. The compiler reads the FORTRAN 5 source program from the second record of the input file. After successful compilation, the system executes the program. Because the program reads data from the input file, it takes input data from the third record of the input file. Normal job termination occurs when the system reads the command record EOR (the first 7/8/9 card).

You can also consider an interactive job to be a file of commands and data. However, you do not submit the file to the system as a unit, but enter input one command at a time from a terminal. Generally, the system processes each command as soon as you enter it.

JOB INITIATION AND TERMINATION

The methods of job entry and termination differ for interactive and batch jobs.

BATCH JOBS

You can enter a batch job in the following ways:

- Enter a job file through a local card reader.
- Enter a job file through a remote batch terminal.
- Enter a job file using the LDI, ROUTE, or SUBMIT command from an interactive or another batch job.

To enter a job file through a remote batch terminal, you must use the Remote Batch Facility (RBF). Consult the Network Products Remote Batch Facility Version 1 Reference Manual for details. Section 7 describes the LDI, SUBMIT, and ROUTE commands.

Unless abnormally terminated, your batch job ends after the system executes the last command in the command record of your job file.

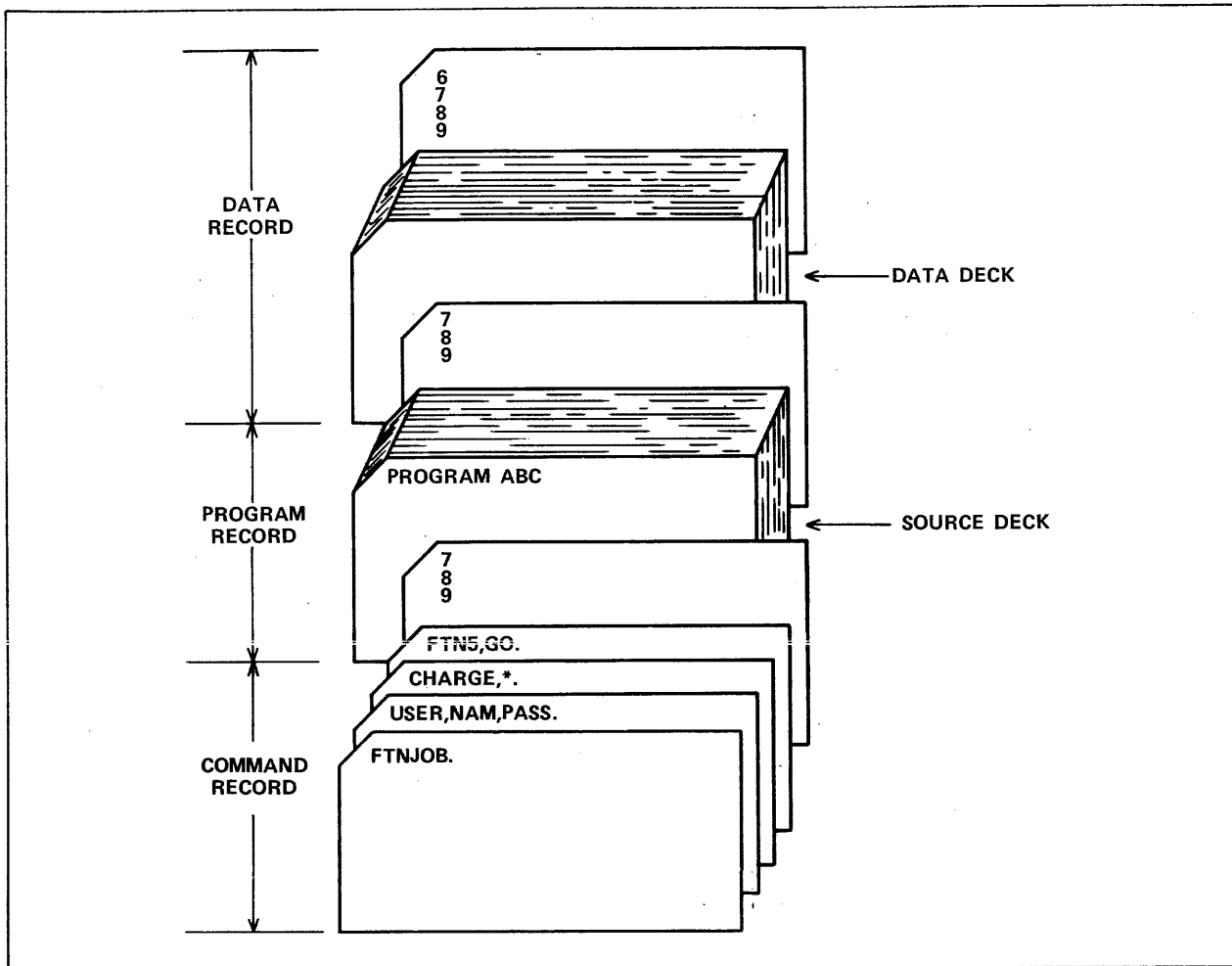


Figure 3-1. FORTRAN Compile and Execute Deck

INTERACTIVE JOBS

To enter an interactive job, you must use the Interactive Facility (IAF). Volume 1, Introduction to Interactive Usage, gives a detailed description of login and logout procedures for interactive jobs. Section 17 describes terminals and input/output conventions.

Login

The following example shows an abbreviated login sequence. Your entries appear in lowercase characters.

```
81/12/18. 14.05.12. LE0T1
CDC NETWORK OPERATING SYSTEM
FAMILY: ,username,password
JSN: ABMK, NAMIAF
```

```
NOS 2
```

```
READY.
```

After the system validates your user name and password, it assigns your job a job sequence name (JSN). The JSN in the preceding example is ABMK.

If you have recoverable jobs in the system, the system initiates a recovery dialogue with you immediately after it logs you in. The following example shows a login sequence where you have recoverable jobs. Again, your entries appear in lowercase characters.

```
81/12/18. 14.11.57. LGOT1
CDC NETWORK OPERATING SYSTEM          NOS 2
FAMILY: ,username,password
JSN: ABMV, NAMIAF
```

RECOVERABLE JOB(S)

JSN	UJN	STATUS	TIMEOUT
ABMK	AJLA	SCH ROLLED	
ABMQ	AJLA	SUSPENDED	8 MIN.

```
ENTER GO TO CONTINUE CURRENT JOB,
      RELIST TO LIST RECOVERABLE JOBS,
      OR DESIRED JSN: go
READY.
```

Additionally, some sites may require you to enter charge information during login (refer to CHARGE command in section 7).

Logout and Application Switching Procedure

When you have finished using IAF, you can either log out or switch to another application. The logout procedure ends your session at the terminal. Application switching ends your session with IAF but allows you to continue processing under the control of another application (such as RBF).

To terminate your terminal session, enter one of the following:

GOODBYE

or

BYE

or

LOGOUT

The system responds by printing

```
UN=username  LOG OFF      hh.mm.ss.
JSN=jsn      SRU-S        s.sss
IAF          CONNECT TIME hh.mm.ss.
LOGGED OUT.
```

The username is the user name you entered at login, the first hh.mm.ss. is the time of logout, jsn is your job sequence name, s.sss is a measure of the system resources used while connected to IAF, and the second hh.mm.ss. indicates how long the terminal was connected to IAF.

IAF automatically logs out the terminal if no activity has been registered in a site-defined timeout period (the default is 10 minutes), unless you have a no-timeout terminal status. Refer to the discussion of the LIMITS command for further information.

When IAF logs out a terminal connected to a packet switching network (PSN), the PSN decides whether to disconnect the line. When IAF logs out a dial-up terminal directly connected to the system, the system automatically disconnects the line.

You may wish to log out without disconnecting the terminal. To log out of the system and reinitialize the login sequence, enter:

HELLO

or

LOGIN

IAF logs out the current job, issues the normal logout messages, and then automatically causes the network to initiate a new login sequence. Any terminal characteristics, such as page width or terminal class set in the previous job, remain in effect.

If you wish to leave the IAF application but remain connected to the network and use another application, you can enter one of the following:

BYE,application

GOODBYE,application

HELLO,application

LOGIN,application

LOGOUT,application

In these commands, the term application means a product that uses the network for terminal communications. Other applications include TAF, MCS, and RBF (refer to the BYE command in section 8). Other site-provided applications may also be available. All terminal characteristics in effect under IAF remain in effect under the new application.

After you enter one of the above commands, the system prints the logout message and disconnects the terminal from IAF. If the site has authorized you to access the application, the system connects the terminal to the named application. If the named application is not present or if the site has not granted you access to this application, the system issues an error message and the prompt:

terminalname - APPLICATION:

where terminalname is the name the network has given your terminal.

You can again enter the name of an application, or you can enter BYE or LOGOUT to log out or HELLO or LOGIN to reinitiate the login sequence.

JOB ORIGIN TYPES

When a job enters the system, the system determines the job origin type according to the means used for job initiation. Its origin identification remains with the job throughout job processing. The job origin type determines how the job is handled and how it exits from the system.

There are four origin types:

- System origin
- Interactive
- Local batch
- Remote batch

System origin jobs originate at the system console, interactive jobs all enter through IAF, local batch jobs enter through central site batch devices, and remote batch jobs enter through RBF.

If you are authorized (refer to the LIMITS command), you can also initiate jobs using the LDI, ROUTE, or SUBMIT commands. Depending on how you specify the parameters of these commands, the resulting jobs can either be local batch or remote batch in origin type.

JOB SERVICE CLASSES

When your job enters the system, the system assigns the job a service class based on the job's origin type. The system uses this service class to determine the job's scheduling priority as the job flows through the system.

The system assigns your job one of the following service classes:

- Batch (B)
- Remote batch (R)
- Interactive (T)
- Detached interactive (D)

You can use the ENQUIRE command to ascertain the service class of any of your jobs (refer to section 7). In the ENQUIRE command output, the one-character mnemonics given in the preceding list indicate the service class of a job.

Unlike a job's origin type, a job's service class can change during job processing. Most notably, when you detach an interactive job (refer to the Detach Command in section 8), its service class changes from interactive to detached interactive and when you recover it, its service class becomes interactive again.

JOB NAMES

Every job in the system receives a job sequence name (JSN) and a user job name (UJN). This includes not only your job file, but files you explicitly route for disposal during your job, such as files routed to a line printer. These job names are the primary job identifiers for both the system and you. Many system commands use the JSN and UJN as parameters. You can get a list of the JSN and UJN of all your active jobs by using the ENQUIRE command.

JOB SEQUENCE NAME (JSN)

The system assigns every job a unique JSN. Each JSN consists of four alphabetic characters. The banner page of batch output prominently displays the job's JSN: the JSN appears as the last four characters of the eight characters in block letters (refer to appendix D for a sample banner page).

USER JOB NAME (UJN)

The UJN is for your convenience. You can select a meaningful name by which to identify your jobs. For batch jobs, you can set the UJN with the SETJOB or ROUTE command. If not otherwise specified, the UJN for batch jobs defaults to the UJN specified on the Job command. For interactive jobs, you can specify the UJN with the SETJOB command. The UJN for interactive jobs defaults to the user index hash (refer to the ENQUIRE command).

VALIDATION

In batch jobs, the USER command follows the Job command and is used to identify you as an authorized user. If you are an authorized user, a set of control values associated with your user name is used by the system to control all system requests from your job. If you are not permitted to perform specific functions (such as access nonallocatable devices) and you attempt to use them, your job will be terminated.

To get a listing of resources at your disposal and of special permissions you possess, enter the LIMITS command. To change your resource allocation or to get additional permissions, contact installation personnel.

ACCOUNTING

The unit of accounting for the system is the system resource unit (SRU). The SRU is a composite value of central processor time, I/O activity, and memory usage. SRU operations are initiated at the beginning of a job and reinitiated whenever another CHARGE command is encountered. SRU information includes:

- Central processor time
- Mass storage activity
- Adder activity (fixed charges for some system requests whose resource requirements are highly variable and beyond your control)
- Magnetic tape activity
- Permanent file activity
- Central memory and extended memory usage
- SRU value
- Matrix Array Processor (MAP) activity
- Application account charges†

This information is written to the job's dayfile at the end of the job or whenever this job's service class changes. You may request SRU information to be written to your output file at any time during the job by issuing the ENQUIRE command. The format of SRU information written in the dayfile is given under Job Completion in this section.

JOB SCHEDULING

When a job enters the system, it is queued for input and waits for the required system resources to become available or its priority to grow. The job's priority depends on its service class. The system priorities are system-defined and can be altered only by the system operator. The scheduling priority of the job is advanced as the job waits. The priority ages to a system-defined limit. The job scheduler periodically scans the queued jobs and active jobs to determine whether action is necessary to ensure that the highest priority jobs are being serviced. This action may include rolling out low priority jobs or rolling in higher priority jobs. The job scheduler is also activated to analyze the system status whenever there are changes (for example, when the field length of a job is released, a job is queued, or a job completes).

Once a job is scheduled for execution, normal command processing begins. The general flow of the command processing is illustrated in figure 5-1.

† Not currently supported by the system but reserved for future use.

JOB CONTROL

While the job is executing, the system exercises the following controls over the job.

FIELD LENGTH CONTROL

The system controls the field length (central memory) assigned to a job, adjusting it according to the requirements of each job step. CYBER Loader further adjusts the field length during both program loading and program execution. Memory may be added or removed as the needs of the program change. Refer to the description of the REDUCE command in the CYBER Loader Reference Manual. You can further influence the field length assigned to your job by using the CM parameter of the Job command and the MFL and RFL commands.

The maximum field length for a job (MAXFL) is set at the smallest of the following values.

- The value of the CM parameter of the Job command if specified
- Maximum field length you are authorized to use
- Maximum field length available for user jobs (dependent on machine size)

For any job step, the maximum field length is the smaller of MAXFL or the value you specified with the MFL command.

The running field length (RFL) is initially set to zero, indicating system control of field length. The RFL command changes RFL. RFL cannot exceed the current MFL.

To set the initial field length for a job step, the system uses the first value set by one of the following.

- Predefined initial field length for a system routine or on a global library set (RFL= or MFL= special entry point).
- Highest high address (HHA) from EACP loader table (54 table) (refer to the CYBER Loader Reference Manual).
- RFL value, if nonzero.
- The smaller of the MFL or the installation-defined default value (release value 50000B).

NOTE

The system automatically assigns a field length for CM only. To set the initial field length for a job step in extended memory, use the RFL command or the MEMORY macro (refer to Volume 4, Program Interface).

The following example shows a command record, the MAXFL, MFL, and RFL settings, and the actual field length (expressed in octal) used to process each command.

<u>Command</u>	<u>MAXFL</u>	<u>MFL</u>	<u>RFL</u>	<u>Field Length</u>	<u>Explanation</u>
JOB,CM60000. USER,USERABC,1234. CHARGE,*.	60000	60000	0	700	The CM parameter sets the MAXFL and MFL values. The system sets the field length as required for processing the command.
GET,ABSPROG,RELPROG.	60000	60000	0	1700	GET command retrieves copies of an absolute program and a relocatable program.
RFL,40000.	60000	60000	0	1500	The user issues an RFL command to set the field length for execution of the absolute program that follows.
ABSPROG.	60000	60000	40000	40000	The absolute program on file ABSPROG is executed within a 40000-word field length.
MFL,50000.	60000	60000	40000	1500	The user issues an MFL command to set the maximum field length for the following relocatable load.
RELPROG.	60000	50000	0	50000	If more than a 50000-word field length is required, the job aborts.

INPUT FILE CONTROL

Jobs, when initiated, have a file named INPUT (input file type). This file contains the commands and other input records required for job execution. INPUT is a locked file. As a result, you may read from it and reposition it, but the system does not allow you to write on it. If for some special reason you need to write on a file named INPUT, you should first issue a RETURN(INPUT) command. This command changes the name of the file from INPUT to INPUT* and leaves it assigned to your job. You may then write on file INPUT. The change of name caused by the RETURN command applies only if the file has an input file type (refer to File Types in section 2).

TIME LIMIT CONTROL

The system sets a default time limit for each job step unless the Job command or the SETTL command specifies a job step time limit. This time limit is the amount of central processor time that any one job step is allowed. You cannot increase the limit beyond that for which you are authorized.

While a job is using the central processor, the CPU time is accumulated and checked against the time limit for each job step. If the job's origin type is not interactive, the job in execution terminates or enters exit processing when the time limit is reached. Interactive jobs are suspended, after which you can increment the time limit and resume execution from the point of suspension (refer to SETTL Command for an example). In the case of a detached interactive job, the system suspends or terminates the job according to the parameter values of the SETJOB command.

SRU LIMIT CONTROL

The system sets a limit on the number of system resource units (SRU) that a job step or an account block can accumulate. An SRU includes central processor time, central memory usage, permanent file activity, and mass storage and tape I/O. An account block is that portion of a job from one CHARGE command to the end of the job or to another CHARGE command. You may alter these limits through the SETJSL and SETASL commands or macros; however, you may not set either limit beyond that for which you are validated.

While a job is in the system, SRUs are accumulated and checked against the SRU step and account block limits. If either limit is reached, the system treats the job as if it exceeded its time limit for a job step, which is described in the preceding subsection.

COMMAND LIMIT CONTROL

The system sets a limit on the number of commands you can enter in a job. You can ascertain this limit by entering the LIMITS command. If your job exceeds this limit, the system so informs you and allows you seven additional commands before terminating your job.

ROLLOUT CONTROL

Each executing program is allowed to reside in CM for a certain amount of time before relinquishing its space to another program. When this CM time slice is exceeded, the program may be rolled out. This means that the contents of the job field length (both CM and extended memory), the job control area, and the control registers (exchange package) are written to mass storage. The program remains on mass storage until it is rolled back into memory. Execution resumes from the point where rollout occurred. The amount of time the job is allowed to occupy CM is called the central memory time slice. The central memory time slice is a system parameter that can be changed only by the system operator; time slices vary for each origin type. Whether a job is rolled out when its time slice expires depends on several factors.

- Whether there are jobs waiting for execution.
- Whether the jobs that are waiting have a lower priority.
- Whether jobs that are waiting require more field length than would be available if all jobs of lower priority were rolled out.

When a job is rolled out, it is assigned a priority. The priority assigned is a system parameter that depends on a job's service class and can be changed only by the system operator. The job's priority increases while the job waits. Normally, all other factors being equal, the job with the highest priority is selected to be rolled in.

ERROR CONTROL

When job step activity ceases, the system must determine the next command to process. If activity ceased due to normal termination, the next command processed is the next command in sequence. If an error caused activity to cease, the system issues the appropriate dayfile message and exits from the job.

Errors may be detected by system software or hardware. When the system hardware detects an error condition, NOS issues two or more dayfile messages. The first message gives the address where the error was detected; the second and following messages give the types of errors that occurred. NOS then dumps the exchange package for the job either to OUTPUT, for local batch and remote batch origin jobs, or to local mass storage file ZZZDUMP, for interactive jobs (refer to section 14). ZZZDUMP is not rewound before or after the dump.

After issuing the appropriate dayfile message(s) for the error(s), the system searches for an EXIT command. If an EXIT command is found, processing continues with the command following EXIT. If an EXIT command is not encountered, the system terminates the job. (Exit processing is further described in section 5.) If you issue a NOEXIT command, the system does not search for an EXIT command on subsequent errors, and processing continues with the next command.

You can specify the error exit mode on which the system is to abort a job step with the MODE command. For example, you can specify that address out of range, operand out of range, and/or indefinite operand errors are allowed and program execution continues (refer to section 6). The default error exit mode specifies that all errors terminate the job.

The EREXIT, RECOVER, REPRIEVE, and MODE macros control error processing in COMPASS programs. The SETLOF macro specifies file completion procedures when a job step abort occurs. These macros are described in Volume 4, Program Interface.

SECURITY CONTROL

A job cannot dump or directly change the contents of the job field length immediately after processing a protected command or user program. A COMPASS program can request protection through the SETSSM macro (refer to Volume 4, Program Interface). These security restrictions do not apply if the job is of system origin or if you have system origin privileges and debug mode has been set at the system console.

A load/dump memory command cannot immediately follow a protected command.

Protected Commands

ASSIGN	COPYBF	EDIT	REQUEST	TCOPY
BEGIN	COPYBR	ELSE	RESOURC	UNLOAD
BLANK	COPYCF	ENDIF	RESTART	USER
CATALOG	COPYCR	ENDW	RETURN	VERIFY
CATLIST	COPYEI	ENQUIRE	REVERT	VFYLIB
CHARGE	COPYSBF	IFE	REWIND	WHILE
CKP	COPYX	LABEL	SET	
CLEAR	DAYFILE	LIBEDIT	SKIP	
COPY	DISPLAY	RECOVER	SORT	

Load/Dump Memory Commands

DMB	DMDECS	DMPECS	LOC	RBR
DMD	DMP	LBC	PBC	WBR

In an interactive job, you cannot enter a load/dump memory command, CKP command, or RESTART command individually or as the first command of a procedure.

If you attempt to change or dump protected memory, the system issues an informative message to the dayfile and terminates the job step.

The programs you execute can affect the security of your files and validation information. When you execute a program, it can access the command record of your job. It is possible that a program belonging to another user can obtain privileged information like your user name, passwords, and charge numbers.

JOB COMPLETION

When there is no more activity at a control point, no outstanding central processor requests, and no commands to process, the job is completed in the following manner.

1. All CM assigned to the job is released.
2. Extended memory assigned to the job is released.
3. All equipment assigned to the job is released.
4. All library files attached to the job are released.
5. All scratch (local) file space used by the job is released.
6. All direct access permanent files attached to the job are released; the status information for these files is updated.

7. The following summations of job activity are added to the end of the job dayfile. This information is also issued to the associated account dayfile for site usage.

- Adder activity in kilounits (incremented by USER commands, CHARGE commands, and resource assignments).

hh.mm.ss.UEAD, xxxxxx.xxxKUNS.

- Permanent file activity in kilounits:

hh.mm.ss.UEPF, xxxxxx.xxxKUNS.

- Mass storage activity in kilounits:

hh.mm.ss.UEMS, xxxxxx.xxxKUNS.

- Magnetic tape activity in kilounits:

hh.mm.ss.UEMT, xxxxxx.xxxKUNS.

- Accumulated central processor time in seconds:†

hh.mm.ss.UECP, xxxxxx.xxxSECS.

- SRU value in units for total job usage including CPU time, I/O activity, and memory usage:

hh.mm.ss.AESR, xxxxxx.xxxUNTS.

- Matrix Array Processor (MAP) III accumulator

hh.mm.ss.AEMP, xxxxxx.xxxUNTS.

†If the installation defines a CPU multiplier value, the value given is the product of the actual CPU seconds and the multiplier. The installation may assign a different CPU multiplier value to each CPU within a dual-processor machine (refer to the NOS 2 System Maintenance Reference Manual).

8. The following information is printed at the end of files queued for printing.

- Lines printed in kilolines:

hh.mm.ss.UCLP, mi,es, xxxxxx.xxxKLNS.

or

hh.mm.ss.UCLV, mi,es, xxxxxx.xxxKLNS.

mi Machine identifier.

es Equipment status table (EST) ordinal of the output device.

The UCLV summation is issued if the V carriage control character was used (refer to appendix I).

9. The following information is issued to the account dayfile only.

- Cards punched in kilocards:

hh.mm.ss.jsn. UCPC. mi,es. xxxxxx.xxxKCDS.

10. For batch jobs, the job dayfile is copied to the end of file OUTPUT. If OUTPUT does not exist or if it is a deferred routed file with EC=A9 specified, the dayfile is copied to another file that the system queues for printing.†
11. All deferred routed files are queued for disposal. The files named OUTPUT, PUNCH, PUNCHB, and P8 are also queued for disposal in batch jobs.† The system does not queue output in interactive jobs.

† This step is done only if job output is to be queued. You can prevent the queuing of job output by specifying the appropriate parameter on the SETJOB, ROUTE, SUBMIT, or LDI command.

A procedure is a group of commands in a file apart from the job command record. A procedure call command executes a procedure as a new command record. Control returns to the command record of the procedure call when the procedure terminates. A procedure is to a command record as a subroutine is to a program. Like a subroutine, a procedure usually contains the commands required to perform a single function within a job.

A procedure is stored as a record on a file. Several procedures can exist on one file. The file may be a local file, an indirect access permanent file, or a direct access permanent file. A procedure file can reside on magnetic tape as well as on mass storage.

You can alter the commands in a procedure by entering parameters on the procedure call. If you incorrectly specify parameters, the system prompts you for the required parameters if parameter prompting has been enabled.

This section describes the structure of a procedure, how to call a procedure, how to end a procedure, and the two modes of parameter substitution.

PROCEDURE STRUCTURE

A procedure consists of a procedure header directive and a procedure body. The procedure header directive must be the first line in the procedure. It names the procedure, indicates whether the system will prompt you for incorrect parameters entered on the procedure call, and establishes the requirements for parameter specification.

The procedure body consists of all commands, procedure directives, and data lines between the procedure header directive and the end-of-record or end-of-file. Any system command is allowed within a procedure.

PROCEDURE DIRECTIVES

The following list gives a brief description of each of the procedure directives:

<u>Directive</u>	<u>Function</u>
.PROC	Specifies the procedure name, controls parameter specifications, and enables parameter prompting.
.HELP	Precedes text explaining the procedure or a parameter.
.ENDHELP	Terminates the explanation text for a parameter.
.DATA	Creates a local data file.
.EOR	Inserts an end-of-record mark in the procedure command record or a data file created by the procedure.
.EOF	Inserts an end-of-file mark in the procedure command record or a data file created by the procedure.
.*	Identifies comments in the procedure.

Procedure directives can span more than one line. No line can contain more than 160 characters and each line to be continued must end with a separator.

.PROC Directive

Every procedure requires a procedure header directive. The other directives are optional. The .PROC directive must be the first line of a procedure. It names the procedure, controls parameter prompting, and defines the parameters that can appear on a call to the procedure. The .PROC directive has a parameter-prompting format and a passive format. Appendix M describes the passive format. The following description applies only to the parameter-prompting format.

This directive enables the interactive processing of a procedure call (refer to calling a Procedure Interactively later in this section). The directive controls parameter specifications in the following manner:

- Designates procedure parameters as optional or required.
- Designates permissible values and correct syntax for each parameter through a checklist.
- Supplies descriptions for each parameter.

Parameter-prompting format:

`.PROC, pname*I, p1, p2, ..., Pn.`

A comma must separate `.PROC` and the procedure name. The separator between the procedure name and the procedure parameters can be a comma, reverse slant (`\`), or slant (`/`). Reverse slants and slants have special significance. Refer to Parameter Matching Modes described later in this section.

`pname*I` Specifies the procedure name; `pname` can be one to seven alphanumeric characters. The procedure name should begin with an alphabetic character. You must append `*I` to the procedure name to enable parameter prompting.

`pi` Specifies optional parameters. The maximum number of parameters is 50. Each parameter `pi` has the following form:

`keywrddescription"=(checklist)`

`keywrdd` Specifies the keyword of the parameter. `Keywrdd` can be a string of 1 to 10 characters. The occurrences of `keywrdd` in the procedure body are replaced by a value that conforms to the specifications made in the checklist. This value is specified by a parameter on the procedure call or by default according to checklist specifications..

`description` Specifies an optional 1- to 40-character text string that must be within quotation marks. The system displays this text string when prompting for a parameter. Description may be null (`""`) or omitted.

`checklist` Specifies a list of the acceptable values and syntax for `pi`. The checklist must be surrounded by parentheses. The value specified for a parameter in a procedure call is compared to each of the entries in the checklist in left to right order and a match must occur for a value to be acceptable.

If the checklist is omitted, the system assumes that any 1- to 40-character value is valid as long as you delimit special characters with dollar signs.

A description of acceptable checklist entries follows.

*N=value Specifies the substitution for
*N= keywrđ when there is no parameter
*N entry for p_i on the procedure call.
 If *N=value is in the checklist, value
 replaces keywrđ in the procedure
 body. If value is #DATA, all
 occurrences of keywrđ in the procedure
 body are replaced by the name of the
 data file created by the .DATA
 directive (refer to .DATA Directive
 for the name of this file). If value
 is #FILE, all occurrences of keywrđ in
 the procedure body are replaced by the
 name of the file that contains the
 procedure. Any data read from this
 file begins with the record
 immediately following the record that
 houses the procedure.

If *N= is in the checklist, a null
value replaces keywrđ. If *N is in
the checklist, no substitution occurs.

If no format of the *N entry is in the
checklist, P_i is a required
parameter and interactive prompting
occurs when it is omitted from the
procedure call.

Specifying some form of the *N entry
more than once in a single checklist
is an error.

Example:

Procedure SUB is on local file PROCFIL.

```
.PROC,SUB*I,P1=(*N),P2>(*N=X),P3>(*N=).  
COMMENT. #P1=P1, #P2=P2, #P3=P3, P4
```

SUB is called:

```
begin,sub.
```

The procedure body becomes:

```
COMMENT. P1=P1, P2=X, P3=, P4
```

***K=value** Specifies the substitution for
***K=** keyword when the parameter entry for
***K** P_i on the procedure call is only the
keyword keyword. If *K=value is in the
checklist, value replaces keyword in
the procedure body. If value is
#DATA, all occurrences of keyword in
the procedure body are replaced by the
name of the data file created by the
.DATA directive (refer to .DATA
Directive for the name of this file).
If value is #FILE, all occurrences of
keyword in the procedure body are
replaced by the name of the file that
contains the procedure. Any data read
from this file begins with the record
immediately following the record that
houses the procedure.

If *K= is in the checklist, a null
value replaces keyword. If *K is in
the checklist, no substitution occurs.

Specifying some form of the *K entry
more than once in a single checklist
is an error.

Example:

Procedure KEY is on local file PROCFIL.

.PROC,KEY*I,P1=(*K),P2=(*K=X),P3=(*K=).

COMMENT. #P1=P1, #P2=P2, #P3=P3

KEY is called:

BEGIN,KEY,,P1,P3,P2.

The procedure body becomes:

COMMENT. P1=P1, P2=X, P3=

*F=value Specifies that the parameter entry for P_i on the procedure call be a file name that conforms to the operating system format for a local file name. If *F=value is in the checklist, value replaces keywrd in the procedure body. If value is #DATA, all occurrences of keywrd in the procedure body are replaced by the name of the data file created by the .DATA directive (refer to .DATA Directive for the name of this file). If value is #FILE, all occurrences of keywrd in the procedure body are replaced by the name of the file that contains the procedure. Any data read from this file begins with the record immediately following the record that houses the procedure.

If *F= is in the checklist, a null value replaces keywrd. If *F is specified, the file name specified on the procedure call replaces keywrd.

If you specify some form of the *F entry more than once in a single checklist, the system uses the left-most entry.

Example:

Procedure EXEC is on local file EXEC:

```
.PROC,EXEC*I,I>(*F),B>(*N=LGO,*F),  
L(*F=OUTPUT).  
FTN5,#I=I,#B=B,#L=L.
```

EXEC is called:

```
EXEC,I=CARDS,L=PRINT.
```

The procedure body becomes:

```
FTN5,I=CARDS,B=LGO,L=OUTPUT.
```

*A=value Specifies the substitution for
*A= keywrđ regardless of the
*A specifications for p_i on the
procedure call. If *A=value is
in the checklist, value replaces
keywrđ in the procedure body.
Value can be a string of 1 to 40
characters (special characters
must be \$-delimited). If value
is #DATA, all occurrences of
keywrđ in the procedure body are
replaced by the name of the data
file created by the .DATA
directive (refer to .DATA
Directive for the name of this
file). If value is #FILE, all
occurrences of keywrđ in the
procedure body are replaced by
the name of the file that
contains the procedure. Any data
read from this file begins with
the record immediately following
the record that houses the
procedure.

If *A= is in the checklist, a
null value replaces keywrđ. If
*A is specified, whatever is
specified on the procedure call
replaces keywrđ.

If you specify some form of the
*A entry more than once in a
single checklist, the system uses
the left-most entry.

*Sn(set)=value Specifies that the parameter entry
 for p_i on the procedure call
 *Sn(set)= contain 1 to n characters from set. The
 *Sn(set) string set can contain 1 to 40
 characters. Special characters must be
 \$-delimited. n is the maximum number of
 characters the parameter entry for p_i
 can contain on the procedure call. n must
 be greater than 0 and 1 is its default.

If *Sn(set)=value is in the checklist,
 value replaces keywrđ in the procedure
 body. If value is #DATA, all occurrences
 of keywrđ in the procedure body are
 replaced by the name of the data file
 created by the .DATA directive (refer to
 .DATA Directive for the name of this
 file). If value is #FILE, all occurrences
 of keywrđ in the procedure body are
 replaced by the name of the file that
 contains the procedure. Any data read
 from this file begins with the record
 immediately following the record that
 houses the procedure.

If *Sn(set)= is in the checklist, a null
 value is substituted for keywrđ. If
 *Sn(set) is in the checklist, the
 parameter entry for p_i on the procedure
 call replaces keywrđ.

Example:

Procedure COPIL is on local file COPI

```

.PROC,COPIL*I,O"OLD FILE NAME"=(*F,*N=OLD),
R"REPLACEMENT FILE NAME"=(*F,*N=LGO),
N"NEW FILE NAME"=(*F,*N=NEW),
L"LAST RECORD"=(*F,*N=),
F"FLAG"=(*S4(ARTE),*N=).
COPYL,O,R,N,L,F.
REVERT.
  
```


COPIL is called:

```
BEGIN, COPIL, COPI, O=OLD, R=MODIFIED,  
N=NEW, F=AE.
```

The procedure body becomes:

```
COPYL, OLD, MODIFIED, NEW, , AE.
```

Procedure COPIL will accept one to four letters for the F parameter.

More than one set may be specified for a parameter. For example, procedure SET has two sets specified for the P parameter.

```
.PROC, SET*I, P=(*S3(ABC), *S3(XYZ)).
```

Parameter entries could include P=BB or P=XZY but not P=AZ or P.

Null sets are not allowed. For example, all the set entries for P1 in the following procedure header are in error.

```
.PROC, SET*I, P1=(*S, *S(), *S3).
```

string=value Specifies the substitution for
string= keyword when the parameter entry
string for p_i on the procedure call
matches string. String must be 1
to 40 characters long. Special
characters must be \$-delimited. If
string=value is in the checklist,
value replaces keyword in the
procedure body. If value is #DATA,
all occurrences of keyword in the
procedure body are replaced by the
name of the data file created by
the .DATA directive (refer to .DATA
Directive for the name of this
file). If value is #FILE, all
occurrences of keyword in the
procedure body are replaced by the
name of the file that contains the
procedure. Any data read from this
file begins with the record
immediately following the record
that houses the procedure.

If string= is in the checklist, a
null value replaces keyword. If
string is in the checklist, string
replaces keyword.

Example:

Procedure LABL on a global library file:

```
.PROC,LABL*I, FN"FILE NAME"=(*F),  
VSN" OF TAPE"=(*A),  
WRITE "YES OR NO"=(YES=W,NO=R).  
LABEL, FN, #VSN=VSN, #PO=WRITE, WRITE.  
REVERT.
```

Procedure LABL is called:

```
LABL, FN=STATS, VSN=TAPE24, WRITE=NO.
```

Procedure body becomes:

```
LABEL, STATS, VSN=TAPE24, PO=R, R.
```

.HELP Directive

The .HELP directive indicates that the text that follows it is information about the procedure itself or its parameters. The procedure caller can access this information by entering a question mark as a parameter in a procedure call or by appending one to the procedure name or a parameter name. When the system encounters a question mark in a procedure call, it stops processing the call and starts help processing. Therefore, anything entered after the question mark will not be read. Refer to calling a Procedure Interactively later in this section.

When you request information about the procedure itself, the system displays the following:

- The text following the procedure's .HELP directive.
- The parameters that follow the procedure name on the .PROC directive (unless suppressed by the NOLIST parameter of the .HELP directive).
- A prompt for a value. The prompt includes the descriptive string specified with the parameter in the .PROC directive.

When you request information about a parameter on a procedure call, the system displays the following:

- Parameter values that are acceptable according to the parameter's checklist (the presence of NOLIST on the parameter's .HELP directive suppresses the display of these values).
- The text that follows the parameter's .HELP directive.
- The current value, if any, of the parameter.
- A prompt for a value. The prompt includes the descriptive string specified with the parameter on the .PROC directive.

All .HELP directives, if specified, must immediately follow the .PROC directive. You must not specify a terminator on a .HELP directive.

Formats:

```
.HELP  
.HELP,,NOLIST  
.HELP,keywr  
.HELP,keywr,NOLIST
```

keywr Specifies one of the keywords that appear in the .PROC directive.

The first and second formats indicate that the text that follows the directives describes the procedure itself. The NOLIST parameter suppresses the display of a parameter list when the caller requests information about the procedure itself.

The third and fourth formats indicate that the text that follows the directive describes the parameter associated with keywr. The NOLIST parameter suppresses the display of acceptable parameter values as given in the parameter's checklist.

The text information starts on the line following the .HELP directive. The text can span multiple lines and must end information with another .HELP directive or an .ENDHELP directive. The HELP text is not subject to parameter substitution as is the remainder of the procedure body.

Examples of the .HELP directive are in the .ENDHELP Directive subsection.

.ENDHELP Directive

The .ENDHELP directive specifies the end of the help text in the procedure body. Only one .ENDHELP directive is allowed in a procedure. You must not specify a terminator on the .ENDHELP directive and nothing else can appear on the same line.

Format:

```
.ENDHELP
```

Example:

Procedure PRINT, on file PRINT, verifies that the selected file is local before it routes the file to a printer:

```
.PROC,PRINT*I,F"FILE NAME"=(*F),
DC>(*N=LP,LP,PR,LR,LS,LT).
.HELP
THIS PROCEDURE ROUTES A PERMANENT FILE TO THE SELECTED LINE PRINTER.
.HELP,F
THE NAME OF THE PERMANENT FILE TO BE ROUTED.
.HELP,DC
THE DISPOSITION CODE. DC ACCEPTS ONLY THE LINE PRINTER OPTIONS.
.ENDHELP
IFE,FILE(F,.NOT.LO),PF.
GET,F.
ELSE,PF.
REWIND,F.
ENDIF,PF.
COPYSBF,F,Z.
ROUTE,Z,#DC=DC.
REVERT,NOLIST.
```

To get a description of procedure PRINT, either append a question mark to the file name or enter a question mark as the first parameter. If PRINT is a local file, you can get such a description by entering any of the following:

```
PRINT?
PRINT,?
BEGIN,PRINT,PRINT?
BEGIN,PRINT,PRINT,?
```

If PRINT is not local, enter any of the following:

```
BEGIN,PRINT,PRINT?
or
BEGIN,PRINT,PRINT,?
or
GET,PRINT
PRINT,?
```

The following is a sample of the dialogue generated by the .PROC and .HELP directives:

```
begin,print,print?
THIS PROCEDURE ROUTES A PERMANENT FILE TO THE SELECTED LINE PRINTER.
PARAMETERS FOR PRINT ARE F, DC
ENTER F FILE NAME? f?
ALLOWABLE VALUE(S)
MUST BE A FILE NAME
THE NAME OF THE PERMANENT FILE TO BE ROUTED.

ENTER F FILE NAME? datafil
ENTER DC ? dc?
ALLOWABLE VALUE(S)
LP
PR
LR
LS
LT
PARAMETER MAY BE OMITTED
THE DISPOSITION CODE. DC ACCEPTS ONLY THE LINE PRINTER OPTIONS.

ENTER DC ? pr
```

.DATA Directive

A .DATA directive in a procedure marks the beginning of a sequence of data lines to be written to a separate file when the procedure is called. File marks generated by .EOR and .EOF commands can subdivide the lines written to the data file into records or files. The sequence of data lines is terminated by one of the following:

- Another .DATA directive.
- An end-of-record (not an .EOR directive) in the procedure file.
- An end-of-file (not an .EOF directive) in the procedure file.
- An end-of-information in the procedure file.

The data file created does not include the .DATA directive. Keyword substitution continues within the data file.

Format:

.DATA,lfn

lfn Specifies the optional name of the file on which the data lines are to be written. If a file named lfn is already assigned to the job, it is returned, and new local file lfn is created. After the data file is written, it is rewound.

If lfn is omitted, the default file name referenced by special default #DATA is used. At the first procedure level, the system calls this file ZZCCLAA; at the second procedure level it is called ZZCCLAB; and so on.

You cannot specify a terminator on the .DATA directive.

The following examples show three different ways of inserting a FORTRAN 5 program into a procedure.

Example 1: Procedure accesses program data with .DATA directive

The following procedure is on an indirect access permanent file named DATAFIL.

```
.PROC,ALPHA*I,P1>(*F,*N=#DATA),X>(*N=FTNOUT).
FTN5,I=P1,L=X.
LGO.
REPLACE,X=LISTFIL.
.DATA
  PROGRAM X
    .
    .
    statements
    .
    .
  END
```

The following BEGIN command in a command record of the job accesses procedure ALPHA on file DATAFIL.

```
BEGIN,ALPHA,DATAFIL.
```

A sample of a resulting dayfile is:

```
11.07.59.BEGIN,ALPHA,DATAFIL.
11.07.59.FTN5,I=ZZCCLAA,L=FTNOUT.
11.08.00.    61000 CM STORAGE USED.
11.08.00.    0.015 CP SECONDS COMPILATION TIME.
11.08.00.LGO.
11.08.00.    STOP
11.08.00.    5600    FINAL EXECUTION FL.
11.08.00.    0.000 CP SECONDS EXECUTION TIME.
11.08.00.REPLACE,FTNOUT=LISTFIL.
11.08.01.$REVERT.CCL
```

All input after the .DATA directive (the FORTRAN 5 source program) is written onto the default temporary file ZZCCLAA. Parameter substitution also occurs in ZZCCLAA.

Example 2: Procedure accesses program data with #FILE

The following procedure is on an indirect access permanent file named PFILE. The record immediately following procedure BETA contains the program data. The #FILE default tells the FTN5 compiler to search for input from the next record on file PFILE.

```
.PROC,BETA*I,P1>(*N=#FILE),X>(*N=FTNOUT).
FTN5,I=P1,L=X.
LGO.
REPLACE,X=LISTFIL.
-EOR-
      PROGRAM X
      .
      .
      statements
      .
      .
      END
```

The following call accesses procedure BETA on file PFILE.

```
BEGIN,BETA,PFILE.
```

The following is a segment of the resulting dayfile. Parameter substitution occurred within the procedure but not within the FORTRAN 5 program.

```
11.08.35.BEGIN,BETA,PFILE.
11.08.36.FTN5,I=PFILE,L=FTNOUT.
11.08.36.    61000 CM STORAGE USED.
11.08.36.    0.013 CP SECONDS COMPILATION TIME.
11.08.36.LGO.
11.08.37.    STOP
11.08.37.    5600    FINAL EXECUTION FL.
11.08.37.    0.000 CP SECONDS EXECUTION TIME.
11.08.37.REPLACE,FTNOUT=LISTFIL.
11.08.37.$REVERT.CCL
```

Example 3: Procedure accesses program data from another file

A procedure may access data outside of the procedure file. The following procedure is in the default file PROCFIL. It uses a GET command to access the program data on file TEST and writes the binary output.

```
.PROC,GAMMA*I,P1=(*F),X>(*N=FTNOUT).  
GET,P1.  
FTNS,I=P1,L=X.  
LGO.  
REPLACE,X=LISTFIL.
```

The following call accesses procedure file GAMMA.

```
BEGIN,GAMMA,,P1=TEST.
```

Parameter substitution occurred within the procedure but not within the FORTRAN 5 program, as shown in the following dayfile segment.

```
11.06.48.BEGIN,GAMMA,,P1=TEST.  
11.06.48.GET,TEST.  
11.06.49.FTNS,I=TEST,L=FTNOUT.  
11.06.50. 61000 CM STORAGE USED.  
11.06.50. 0.013 CP SECONDS COMPILATION TIME.  
11.06.50.LGO.  
11.06.51. STOP  
11.06.51. 5600 FINAL EXECUTION FL.  
11.06.51. 0.000 CP SECONDS EXECUTION TIME.  
11.06.51.REPLACE,FTNOUT=LISTFIL.  
11.06.51.$REVERT.CCL
```

.EOR Directive

The .EOR directive is used to separate records in a data file created with a .DATA directive. Wherever an .EOR is placed, an actual end-of-record is recorded when the data file is written on the file specified in the .DATA directive. The system always writes at least one end-of-record at the end of a data file, whether or not you specify an .EOR directive. The .EOR directive is valid only after a .DATA directive. A terminator must not be used and nothing else can appear on the same line.

.EOF Directive

The .EOF directive generates an end-of-file on the data file created with a .DATA directive. Wherever an .EOF appears, an actual end-of-file is recorded when the data file is written on the file specified in the .DATA directive. The system always writes an end-of-record at the end of a data file. If you want an end-of-file also, you must end the file with an .EOF directive. The .EOF directive is valid only after a .DATA directive. A terminator must not be used, and nothing else can appear on the same line.

. * Directive

The `.*` directive enables you to document a procedure with internal comments. These comments do not appear in the dayfile when the procedure is processed. The comment, which follows the `*`, can contain any combination of characters.

PARAMETERS AND THE PROCEDURE BODY

The `.PROC` directive specifies keywords used in the procedure body. You can change the value of these keywords each time you call the procedure by changing your parameter specifications in the procedure call. The values of these parameters must conform to the restrictions specified by the parameter checklists.

Commands in the procedure body sometimes expand beyond 80 characters after the substitutions for keywords. The system does not allow this for most commands (refer to Command Format in section 5 for exceptions). Ensure that a line containing keywords is short enough to accommodate parameter substitution without extending beyond the 80th character.

When specifying keywords in the procedure body, two special characters, ASCII graphics `#` and `—` (or CDC graphics `≡` and `⌈`), are used to inhibit keyword substitution and to combine parts of a parameter after keyword substitution.

A single `#` character placed immediately before a keyword in a procedure command inhibits substitution for that keyword. This character is removed when the procedure is processed. If you want a `#` character to remain in the procedure body after the parameter substitutions, use two such characters (`##`). In this case, keyword substitution still occurs. If `#` characters precede nonkeywords, they are removed in a similar fashion.

The linking character, underline, is used in a procedure body to temporarily separate two parameters (keyword or nonkeyword). After possible substitutions are made, the underline character is removed and the two parameters are merged into one. `#` before `—` retains `—` and allows substitution. `—` before `#` does not affect the inhibiting action of `#`.

Examples of use of the `#` and `—` characters in a procedure are shown in the following examples.

Example 1: `#` Character

The following procedure resides on file `PROCFIL`.

```
.PROC,INHIBIT*I,I=(*N=TEST).
GET,I.
FTN5,#I=I,L=0.
LGO.
COMMENT. I, #I, I#I, #I#I.
```

On the left is a call to procedure INHIBIT. On the right is the resulting dayfile.

```
10.31.00.BEGIN,INHIBIT.
10.31.00.GET,TEST.
10.31.00.FTN5,I=TEST,L=0.
10.31.00.      56000 CM STORAGE USED.
10.31.00.      0.011 CP SECONDS COMPILATION TIME.
10.31.00.LGO.
10.31.01.      STOP
10.31.01.      5600   FINAL EXECUTION FL.
10.31.01.      0.000 CP SECONDS EXECUTION TIME.
10.31.01.COMMENT. TEST, I, TESTI, II.
10.31.01.$REVERT.CCL
```

Example 2: __ Character

Procedure LINK resides on file FILE1.

```
.PROC,LINK*I,SUFFIX"FOR COPY"=(*N=,BF,BR,CF,CR,EI,SBF,X),LFN1=(*F),LFN2=(*F).
REWIND,LFN1.
COPY_SUFFIX,LFN1,LFN2.
```

The first BEGIN command does a COPYSBF of file PLAN to file SCHEME. The next BEGIN command does a COPYEI of file MAZE to file TAXES. Each resulting dayfile follows the BEGIN command.

```
begin,link,file1,suffix=sbf,lfn1=plan,lfn2=scheme.
```

```
08.00.17.$BEGIN,LINK,FILE1,SUFFIX=SBF,LFN1=PLAN,LFN2=SCHEME.
08.00.18.REWIND,PLAN.
08.00.18.COPYSBF,PLAN,SCHEME.
08.00.18. EOI ENCOUNTERED.
08.00.18.$REVERT.CCL
```

```
begin,link,file1,suffix=ei,lfn1=maze,lfn2=taxes.
```

```
08.03.23.$BEGIN,LINK,FILE1,SUFFIX=EI,LFN1=MAZE,LFN2=TAXES.
08.03.23.REWIND,MAZE.
08.03.24.COPYEI,MAZE,TAXES.
08.03.24. EOI ENCOUNTERED.
08.03.24.$REVERT.CCL
```

CALLING A PROCEDURE (BEGIN COMMAND)

A BEGIN command executes a procedure as a new command record. Control returns to the command record of the procedure call when the procedure terminates. After the final command in the procedure is processed, a system- or user-supplied REVERT command is executed and job processing continues with the command following the BEGIN command. Use of a BEGIN command is illustrated in figure 4-1.

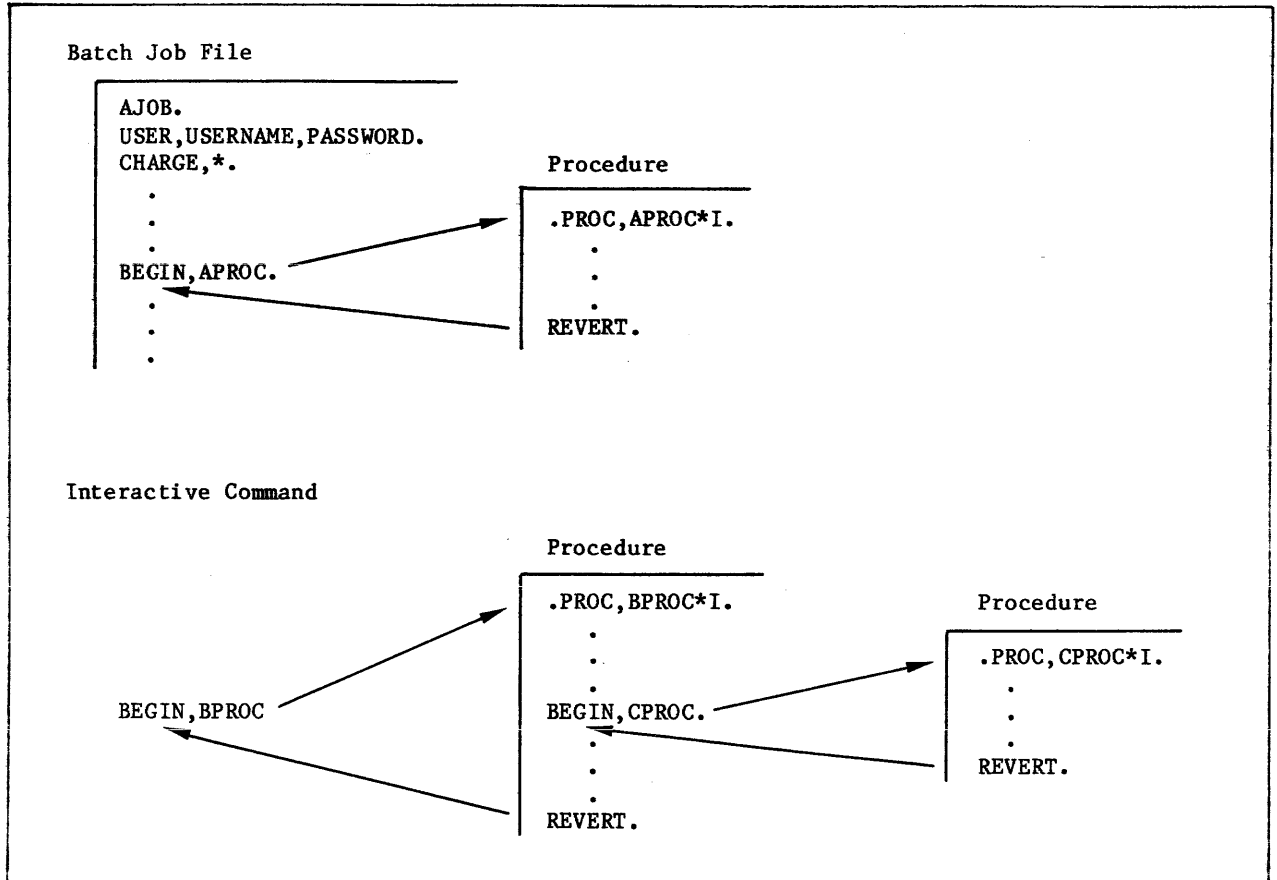


Figure 4-1. Calling a Procedure

You can alter the contents of a procedure by passing parameters to the procedure through the BEGIN command. The system substitutes the parameter values into the procedure body according to the rules specified in the procedure header (refer to .PROC Directive earlier in this section).

Formats:

```
BEGIN, pname, pfile, p1, p2, ..., pn.  
-pname, pfile, p1, p2, ..., pn.  
pname, p1, p2, ..., pn.
```

The second and third formats are referred to as the call-by-name formats. You can use the second format only in interactive jobs. You can use the third format only if the procedure is the first procedure in a local file named pname, is in an active global library file (refer to the LIBRARY command), or is in a system library.

<u>Parameter</u>	<u>Description</u>
pname	<p>Specifies the procedure name from the procedure header directive.</p> <p>In the first and second format, pname is the name of a procedure on pfile.</p> <p>If pname is omitted from the first format, two consecutive commas must be specified. The default procedure is the record at the current position of pfile. If pfile is at its end-of-information, the system rewinds pfile and uses the first procedure on pfile. If pfile is INPUT, the file is not rewound.</p>
pfile	<p>Specifies the name of the file containing the procedure. pfile must be the second parameter in the first format. Its omission is indicated by two consecutive commas following pname. To get a description of the procedure pname, you can append a question mark to pfile (refer to calling a Procedure Interactively later in this section).</p> <p>If pfile is omitted from the first format, the installation-defined default file name is used (PROCFIL is the default).</p> <p>When the BEGIN command is processed, the system looks for a file named pfile assigned to the job. If none exists, it looks for an indirect access file named pfile and retrieves a local copy. If pfile is a direct access permanent file, the system attempts to attach pfile as a direct access file. File pfile is not automatically returned after the procedure executes.</p>
P _i	<p>Specifies an optional parameter that may affect the substitution to be made for a keyword used in the procedure. Refer to the .PROC Directive in this section for more details on parameter substitution.</p>

The following parameter formats are available.

<u>Format</u>	<u>Description</u>
keywrđ	Specifies a parameter identical to a keyword on the procedure header. Substitution is controlled by the *K entry in the checklist for the parameter.
keywrđ? or ?	Specifies interactive BEGIN processing (refer to calling a Procedure Interactively later in this section).
keywrđ=	Removes keywrđ in the procedure body (null substitution) unless overridden by a checklist specification.
val	Unless overridden by a checklist specification, assigns this 1- to 40-character symbolic name or value to the keyword whose position in the procedure header parameter list matches the position of this parameter in the BEGIN command parameter list (refer to Parameter Matching Modes in this section). A value containing special characters must be \$-delimited. This parameter format has the following variations:

<u>Format</u>	<u>Meaning</u>
val	Substitutes the string val itself.
val+	Substitutes the decimal value associated with val.
val+D	Substitutes the decimal value associated with val.
val+B	Substitutes the octal value associated with val.

keywrđ=val Allows order-independent substitution (refer to Parameter Matching Modes in this section). val replaces keywrđ in the procedure body unless the associated checklist specifies otherwise. This parameter format has the following valid variations:

<u>Format</u>	<u>Meaning</u>
keywrđ=val	Substitutes the string val for keywrđ.
keywrđ=val+	Substitutes the decimal value of val for keywrđ.
keywrđ=val+D	Same as keywrđ=val+.
keywrđ=val+B	Substitutes the octal value of val for keywrđ.

When calling a procedure, a keyword can be named more than once if the keyword=val parameter format is used each time. The system issues a message informing you that a keyword is named more than once on the command. It uses the value specified with the last occurrence of the keyword.

Example:

The following procedure is accessed by a sequence of BEGIN commands in the command record of the job.

```
.PROC,TEST1*I,FK.  
COMMENT.    FK
```

The resulting dayfile shows each BEGIN command and the substitutions made. The relevant segment of the dayfile is as follows:

```
10.15.26.BEGIN,TEST1,FKTEST,20.  
10.15.27.COMMENT.    20  
10.15.27.REVERT.CCL  
10.15.27.SET,R2=100.  
10.15.27.BEGIN,TEST1,FKTEST,FK=R2+.  
10.15.28.COMMENT.    100  
10.15.28.REVERT.CCL  
10.15.28.BEGIN,TEST1,FKTEST,FK=R2+D.  
10.15.29.COMMENT.    100  
10.15.29.REVERT.CCL  
10.15.29.BEGIN,TEST1,FKTEST,FK=R2+B.  
10.15.30.COMMENT.    144  
10.15.30.REVERT.CCL  
10.15.30.BEGIN,TEST1,FKTEST.  
10.15.31.COMMENT.    FK  
10.15.31.REVERT.CCL  
10.15.31.BEGIN,TEST1,FKTEST,FK=.  
10.15.32.COMMENT.  
10.15.32.REVERT.CCL  
10.15.32.BEGIN,TEST1,FKTEST,VALUE.  
10.15.33.COMMENT.    VALUE  
10.15.33.REVERT.CCL  
10.15.34.BEGIN,TEST1,FKTEST,$VALUE-2$.  
10.15.34.COMMENT.    VALUE-2  
10.15.34.REVERT.CCL
```

CALLING A PROCEDURE (INTERACTIVELY)

After you enter a procedure call but before the system executes the specified procedure, you can have a dialogue with the system about the procedure. The procedure writer must have appended the string *I to the procedure name in the procedure header (refer to the .PROC directive described in this section). You can do any of the following:

- Request help for the procedure.
- Request the help for a procedure parameter.
- Supply omitted parameters.
- Reenter parameter values that are in error.

If there is an incorrect parameter specification in the procedure call, the system initiates the dialogue. You can also initiate the dialogue by requesting a description of the procedure or parameters in one of the following ways:

- Append a question mark to the procedure file name.
- Append a question mark to the name of a procedure parameter.
- Enter a question mark as a parameter on the procedure call.
- Enter a question mark in response to an interactive prompt.

In the following example, the first two calls request the description of procedure LIST on file FILE and initiate prompting for procedure parameters. The third call requests the description of the parameter KEY and all other parameters that you must specify.

```
-LIST,FILE?  
-LIST,FILE,?  
-LIST,FILE,KEY?
```

If you omit required parameters or any parameter is in error on the procedure call, the system prompts you for the required parameters and those in error. If the system prompts you for a parameter that need not be specified and you want to use the default for that parameter, enter a carriage return. If the format of a parameter entry is not correct or the parameter entry is not specified on the procedure header directive, you are reprompted for the parameter. Prompting for parameters terminates when:

- All parameter requirements are satisfied.
- You enter a parameter terminated by a period or a right parenthesis, enter just a period, or enter a right parenthesis. If all required parameters have been entered, the system executes the procedure. Otherwise, the system continues prompting until all required parameters are satisfied.
- You enter the termination character for your terminal (refer to appendix K). The BEGIN command processing ends and the system does not execute the procedure.

The following example shows the interactive entry of parameters.

Example:

Procedure F5 resides on local file F5.

```
.PROC,F5*I,I"INPUT"=(*F,*N=INPUT),  
B"BINARIES"=(*F,*N=LGO),  
L"OUTPUT"=(*F,*N=OUTPUT),  
LO"LIST OPTIONS"=(*N=O,O,O,R,A,M,S).  
FTN5,#I=I,#B=B,#L=L,#LO=LO.  
REVERT.
```

To elicit prompts for the parameters on the procedure, the user enters

```
f5,?
```

at the terminal.

The following dialogue ensues:

```
PARAMETERS FOR F5 ARE I, B, L, LO  
ENTER I INPUT? test  
ENTER B BINARIES ? bfile  
ENTER L OUTPUT ? listing  
ENTER LO LIST OPTIONS? s  
REVERT.
```

If your job is a batch job and you request a description of the procedure or its parameters, the system writes the description in the job dayfile. If any of the required parameter specifications are in error, error messages appear in the dayfile. In either case, the procedure is not executed.

ENDING A PROCEDURE (REVERT COMMAND)

The REVERT command terminates procedure processing.

Format:

REVERT,opt.com

opt Controls the revert options and whether the REVERT command appears at the terminal and in the job dayfile.

opt

Meaning

ABORT

Returns control to the next EXIT command in the command record unless a NOEXIT command has been processed. This parameter sets EF=CPE (CPU abort). It returns control to the command following the calling BEGIN command if a NOEXIT command has been processed. This parameter always causes the REVERT command to appear at the terminal and in the job dayfile.

EX

Returns control to the command record of the calling BEGIN command and command com is inserted as the next command to be executed. The EX parameter causes the REVERT command to appear in the job dayfile but not at the terminal.

NOLIST

Returns control to the command following the calling BEGIN command and suppresses the display of the REVERT command at the terminal and in the dayfile.

If you omit opt, control returns to the command following the calling BEGIN command. The REVERT command appears in the job dayfile but not at a terminal unless you are under the batch subsystem and the REVERT is not within a nested procedure call.

com Specifies a comment unless used with the EX parameter. In this case, com must be a command.

The system always appends the following commands to a procedure record.

\$REVERT.CCL
\$EXIT.CCL
\$REVERT,ABORT.CCL

These commands terminate procedure processing if no user REVERT commands are processed.

Example:

The following procedure (REVTST) is on a file called PROCFL. It reverts to the job calling it if the named file has no read permission and aborts (causing control to be transferred to the job EXIT command) if the named file has no read modify permission.

```
.PROC,REVTST*I,LFN1,LFN2.
IFE,FILE(LFN1,RD),LABEL1.
TDUMP,I=LFN1.
ELSE,LABEL1.
REVERT.NO READ PERMISSION
ENDIF,LABEL1.
IFE,FILE(LFN1,RM),LABEL2.
COPY,LFN2,LFN1.
ELSE,LABEL2.
REVERT,ABORT. NO READ/MODIFY PERMISSION
ENDIF,LABEL2.
```

The following two jobs (REVJOB1 and REVJOB2) call the REVTST procedure. REVJOB1 attaches an execute-only file; REVJOB2 attaches a read and/or execute file.

```
REVJOB1.                                REVJOB2.
USER,USERNAM,PASSWRD,FAMNAME.           USER,USERNAM,PASSWRD,FAMNAME.
CHARGE,*                                 CHARGE,*
ATTACH,FILE1/UN=ALTUSER,PW=PW1,M=E.     ATTACH,FILE2/UN=ALTUSER,PW=PW2,M=4.
BEGIN,REVTST,PROCFL,FILE1,XFIL.        BEGIN,REVTST,PROCFL,FILE2,XFIL.
COMMENT. RETURNS HERE                   COMMENT. RETURNS HERE
EXIT.                                     EXIT.
COMMENT. EXIT ON ERROR                   COMMENT. EXIT ON ERROR
```

The following are the dayfile segments produced by REVJOB1 and REVJOB2. REVJOB1 processes the REVERT command and terminates normally. REVJOB2 processes the REVERT,ABORT command and terminates via error processing.

```
10.09.51.REVJOB1.                        10.10.11.REVJOB2.
10.09.51.USER,USERNAM,,FAMNAME.          10.10.11.USER,USERNAM,,FAMNAME.
10.09.51.CHARGE,*                         10.10.11.CHARGE,*
10.09.51.* CHARGE(1514,5612PAY)           10.10.11.* CHARGE(1514,5612PAY)
10.09.51.ATTACH,FILE1/UN=ALTUSER,PW=,M=E. 10.10.11.ATTACH,FILE2/UN=ALTUSER,PW=,M=R.
10.09.52.BEGIN,REVTST,PROCFL,FILE1,XFIL. 10.10.12.BEGIN,REVTST,PROCFL,FILE2,XFIL.
10.09.53.IFE,FILE(FILE1,RD),LABEL1.      10.10.14.IFE,FILE(FILE2,RD),LABEL1.
10.09.53.ELSE,LABEL1.                    10.10.14.TDUMP,I=FILE2.
10.09.53.REVERT.NO READ PERMISSION        10.10.14. TDUMP COMPLETE.
10.09.53.COMMENT. RETURNS HERE           10.10.14.ELSE,LABEL1.
10.09.54.EXIT                             10.10.14.ENDIF,LABEL1.
                                           10.10.15.IFE,FILE(FILE2,RM),LABEL2.
                                           10.10.15.ELSE,LABEL2.
                                           10.10.16.REVERT,ABORT. NO READ/MODIFY
                                           PERMISSION
                                           10.10.16.EXIT.
                                           10.10.16.COMMENT. EXIT ON ERROR
```

PARAMETER MATCHING MODES

When you call a procedure, the system must match each parameter on the BEGIN command with a parameter on the procedure header. The system uses two methods of parameter matching, order-dependent and order-independent.

ORDER-DEPENDENT MODE

In order-dependent mode, the system compares in order each parameter on the procedure call with the parameter in that position on the procedure header. If any parameter entries do not conform to the restrictions in the parameter checklist or required parameters are omitted on the procedure call, the system prompts you for them. After all required parameters are entered, the system substitutes the selected keywords into the procedure body according to the checklists in the .PROC directive.

Example:

Procedure ITEM is on a global library file:

```
.PROC,ITEM*I,F"LOCAL FILE NAME"=(*N=LGO,*F),  
L"NAME OF LIST OUTPUT FILE"=(*N=OUTPUT,*F),  
BL"EACH FILE START ON NEW PAGE? YES OR NO"=(YES=$,$BL,NO=,*N=),  
NR"REWIND BEFORE & AFTER? YES OR NO"=(YES=,NO=$,$NR,*N=).  
ITEMIZE,F,#L=L_BL_NR.  
REVERT.
```

The procedure is called:

```
ITEM,LIST,,NO,NO.
```

The parameters are matched in order-dependent mode and the procedure body becomes:

```
ITEMIZE,LIST,L=OUTPUT,NR.  
REVERT.
```

In order-dependent mode, the system treats excess parameters on the BEGIN command as a nonfatal error.

ORDER-INDEPENDENT MODE

For each BEGIN command, parameter matching always starts in order-dependent mode. The system switches to order-independent mode if, in the comparison of a BEGIN command parameter and a procedure header parameter, one of the following occurs.

- A parameter on the procedure call is in the format keyword=value.
- A reverse slant (\) precedes the parameter on the .PROC directive or the BEGIN command.
- A slant (/) precedes the parameter on the .PROC directive.
- A slant precedes the parameter on the procedure call and a slant separates any two parameters on the .PROC directive.

However, if a slant is specified on the procedure call and not on the .PROC directive, the slant is not treated as a separator, but as part of the parameter value.

Parameter matching mode cannot switch back from order-independent to order-dependent mode.

Once in order-independent mode, the system matches each successive keyword on the procedure call or interactive entry to the identical keyword in the procedure header directive, regardless of the order of the procedure header parameters or the order of specification on the procedure call.

To show order-independent parameter matching, the preceding example is slightly modified. An *K entry has been added to the BL parameter checklist to make BL a valid parameter entry. A reverse slant is used as a separator before the NR parameter to ensure order-independent mode.

```
.PROC,ITEM1*I,F"LOCAL FILE NAME"=(*N=LGO,*F),  
L"NAME OF LIST OUTPUT FILE"=(*N=OUTPUT,*F),  
NR"REWIND BEFORE & AFTER? YES OR NO"=(YES=,NO=$,$NR,*N=),  
BL"EACH FILE START ON NEW PAGE? YES OR NO"=(YES=$,$BL,NO=,*N=,*K=$,$BL).  
ITEMIZE,F,#L=L _BL _NR.  
REVERT.
```

The procedure call (ITEM1 is on a global library file):

```
ITEM1,LIST,,BL,NR=NO.
```

starts parameter matching in order-dependent mode. The reverse slant in the procedure header switches parameter matching to order-independent mode. BL and NR are matched in order-independent mode. In order-independent mode you must specify all parameters in the form keyword=value, unless there is an *K entry in the parameter checklist. Then you can specify just the keyword as the parameter entry. Since *K is specified in the BL parameter checklist, the system accepts BL as a parameter entry. The NR parameter must be specified as NR=value or omitted.

After the substitution, the procedure body becomes

```
ITEMIZE,LIST,L=OUTPUT,BL,NR.  
REVERT.
```

Jobs entering the system consist of one or more logical records. The first logical record contains system directives (commands) which describe the processing that is to occur in the job file. In interactive jobs, you enter the commands directly at the terminal. This section describes command processing and how the commands affect other aspects of job processing.

The operating system recognizes four types of commands.

- Local File Commands These commands call programs or procedures on files that are assigned to the job. The name of the command is simply the name of the file. LGO is an example. It is the system default local file used for retaining object code generated by one of the language processors.

- Global Library File Commands These commands call programs or procedures on files that have been specified in a LIBRARY command.

- System Commands These commands are divided into 10 categories.
 - Execution control commands
 - Job control commands
 - Special commands for interactive jobs
 - File management commands
 - Permanent file commands
 - Load and dump central memory commands
 - Tape management commands
 - System utility commands
 - Library utility commands
 - Loader commands[†]

- Product Set Commands The product set commands call the various products available under NOS. Their formats are given in the applicable product reference manual and in the NOS 2 Applications Programmer's Instant.

Since your executing programs can access the command record of your job, it is possible that they might manipulate items like user names and passwords. Hence, your executing programs can affect system security.

[†]Refer to the CYBER Loader Reference Manual.

COMMAND FORMAT

All commands can consist of one to four fields. The first field is the command label field. If present (the field is optional), it begins with a numeric character and terminates with a separator character.

The second field, also optional, is a \$ or / prefix character which precedes the program name. If a \$ is present, it indicates that the specified program to be executed must be loaded from the system library. Therefore, even if a local file of the same name is present or a program or procedure of the same name resides on your global user library, the system program, not the local program or global library program, is executed. In all interactive subsystems except the batch subsystem, the system places a \$ in the second field of all commands. Even in the batch subsystem, the system places a \$ in front of the file editing commands ALTER, DELETE, DUP, LIST, MOVE, READ, WRITE, and WRITEN.

If you use a slant in the second field of local file or global library file commands, the system processes the parameters in operating system format. For global library file commands, the presence of a NPC= entry point in the specified library also forces the system to process the parameters in operating system format (refer to Volume 4, Program Interface). Otherwise, the system processes local file and global library file commands in product set format. The slant option is ignored for command calls to programs residing on the system library. For those types of calls, parameters are processed in the operating system format unless the SC directive of SYSEDIT has been entered. Refer to the SYSEDIT command in the NOS 2 System Maintenance Reference Manual for a description of the SC directive.

Example:

File EXTRACT is a local file command. If you enter

```
/EXTRACT,A,B,10.
```

the system processes the parameters in operating system format because of the leading slant.

The third field contains the name of the program to be executed.[†] The fourth field (optional) contains parameters which further define the operation to be performed. The parameter field is set off from the name field by a separator character. A valid terminator character must follow the fourth field (or the third field if no parameters are present).

The system allows continuation lines for the execution control commands and tape management commands (for details, refer to Command Syntax in section 6 and Command Rules in section 12).

The following is a comparison of the operating system and product set formats (refer to the NOS 2 Applications Programmer's Instant for commands using the product set format).

<u>Operating System Format</u>	<u>Product Set Format</u>
1. Valid separators are + - " / = , (and any other character with a display code value greater than 44g except *) \$. and blank.	1. Same as for the operating system format.

[†]The Job command is an exception (refer to section 7).

Operating System Format

2. Valid terminators are
 .)
3. Letters, numbers, and the * are the only characters allowed in the parameter field. The one exception to this rule is the use of literals (that is, character strings delimited by dollar signs). Characters other than letters, numbers, and the * can be included in literals. No characters within a literal have special meanings; the system merely checks the syntax of the literal. The called program must do its own processing of the literal.
4. All embedded blanks within a command except those appearing in literals are ignored.
5. Comments may appear on the command but they must follow the terminator. They may contain any character. Comments are not printed for some commands.
6. Parameters, separators, and terminators are stored in the user's field length beginning at RA+2. The characters , . and) are stored as binary zero. For all parameters and all valid separators except the comma, their display code equivalent is stored. Refer to section 10 of Volume 4, Program Interface, for more information.

Product Set Format

2. Same as for the operating system format.
3. Same as for the operating system format.
4. Same as for the operating system format.
5. Same as for the operating system format.
6. Parameters are stored in their display code equivalent beginning at RA+2. Separators and terminators are stored as follows:

<u>Character</u>	<u>Code (Octal)</u>
,	1
=	2
/	3
(4
+	5
-	6
;	10
) or .	17
Other valid separators	16

Refer to section 10 of Volume 4, Program Interface, for more information.

Operating System Format

7. File names are one to seven alphanumeric characters.
8. Not NOS/BE compatible.

Product Set Format

7. File names are one to seven alphanumeric characters. In some products, file names beginning with a numeric character are illegal.
8. NOS/BE compatible.

In general, no parameter can contain more than seven characters. The exceptions include the execution control commands and some tape management commands (refer to sections 6 and 12).

Depending on the program, the parameters can appear in either order-dependent or order-independent format. Order-dependent parameters are required when the parameters must be passed in a specific order. An example of order-dependent parameters is:

```
RESEQ,MYFILE,B,,20.
```

In this example, the system expects the resequencing increment to be passed as the fourth parameter; therefore, a separator must be present for the parameter not specified.

Order-independent parameters may be passed in any order. This is made possible by the use of keywords. A keyword is an identifier which has meaning either by itself or when used in conjunction with an option. Usually, keywords are passed with an option and a separator. The separator must not be a comma. When the list of parameters is passed to the called program, all separators except commas are also passed.

Some programs require specific separators (usually =), and others merely require that a separator be present. Examples of keyword notation are:

1. COBOL5,I=SFILE,B=BFILE.
2. COBOL5,B=BFILE,I=SFILE.
3. COBOL5,L=0,E,EL.
4. JOBX,T10,CM45000.

In examples 1 and 2, both parameters and separators are passed to the COBOL 5 compiler. Since these parameters are order independent, both commands produce the same result.

In example 3, two keywords are passed with no separator character. In example 4, the keyword is the first character of the first parameter and the first two characters of the second parameter.

The parameters and an image of the command being processed are written in the job communication area (refer to section 10 of volume 4). The job communication area is the first 110g words of your job's field length, from RA through RA+107g. Section 1 and appendix E in Volume 4, Program Interface, describe the first 100g words of this area.

The following commands produce the same image in the job communication area. Both commands are processed using operating system format.

```
PERMIT,FILEABC,USERAAA=R,USERBBB=W.
```

```
$PERMIT,FILEABC,USERAAA=R,USERBBB=W.
```


COMMAND PROCESSING FLOW

The system translates a command by:

1. Reading the command from the job's command record.
2. Verifying the format of the command as described in Command Format.
3. Comparing special command names with the name of the command being processed. If the command name is CTIME, HTIME, RTIME, or STIME, the system processes the command.
4. Searching the file name table for a file assigned to the job with a name identical to the name of the command. However, if a \$ precedes the program name, this step is skipped. If an identical name is found, the program is loaded into memory. The arguments are extracted from the command and stored in RA+2 through RA+n+1 (n is the number of parameters). The CPU is requested to begin execution unless special loader commands follow.
5. Searching the global library directory for a program name that matches the command name. If a \$ precedes the command name this step is skipped. If the program is found, the system proceeds as in step 4; otherwise, the system searches further.
6. Searching the central library directory for a program name that matches the command name. If the name is found, the system proceeds as in step 4; otherwise, the system searches further.
7. Searching the peripheral processor library directory for a program name that matches the command name if the command name is a three-character name with the first character alphabetic. If found, the name is placed, with a maximum of two arguments, as a peripheral processor request, and the system exits to the program.
8. If the command name is not found during any of the above searches, the command is declared invalid and the job is aborted.

Figure 5-1 illustrates the flow of command processing.

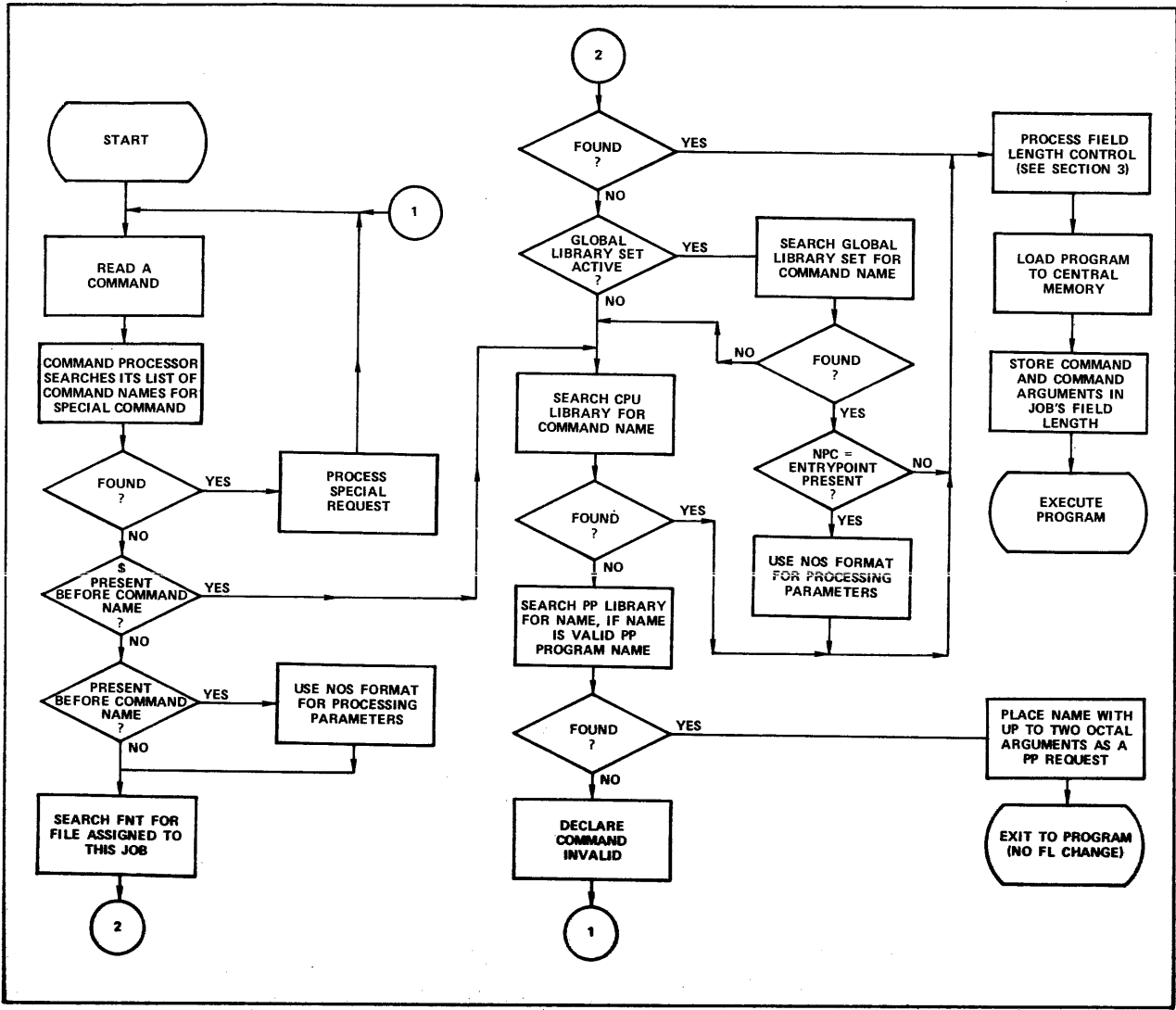


Figure 5-1. Command Processing Flow

EXIT PROCESSING

When an error condition occurs during noninteractive job processing, the system searches the command record for an EXIT command. If the record does not contain an EXIT command, the system terminates the job. If the system finds an EXIT command, it clears the error condition and processes the commands that follow the EXIT command. If the error was a time limit error, the limit is reset to the time used plus eight seconds. This gives you time for post-error cleanup operations. If the error was an SRU limit error, the limit is reset to the SRUs used plus eight SRUs.

If a NOEXIT command has been processed, normal error processing is not performed. That is, if the no exit flag has been set by the NOEXIT command prior to the error, the error flag is cleared, no search is made for an EXIT command, and processing continues with the next command. An ONEXIT command can be used to return to error processing mode; it clears the no exit flag.

The following sequence of commands illustrates this exit processing.

```
JOBCCC.  
USER,SMITH22,SMA1.  
CHARGE,*.  
NOEXIT.  
GET,A,B.  
ONEXIT.  
ATTACH,MASTER/M=W.  
SKIPEI,MASTER.  
COPYBF,A,MASTER.  
COPYBF,B,MASTER.  
PACK,MASTER.  
COPYSBF,MASTER,.  
EXIT.  
ENQUIRE,F.  
-EOR-  
-EOI-
```

This job attempts to make local copies of two indirect access permanent files and adds them to a direct access file. The NOEXIT command suspends error processing, and the job continues even if file A or B is not found. The ONEXIT command turns error processing back on. If any error occurs thereafter, processing skips to the EXIT command and continues with the ENQUIRE command. If no error occurs after the NOEXIT command, processing continues until reaching the EXIT command and then the job terminates (ENQUIRE command is not processed).

EXECUTION CONTROL COMMANDS

6

Execution control commands control the processing sequence of commands within the command record of a job. They can insert commands from a procedure file, conditionally or unconditionally skip commands, and control error processing. To determine the conditions for transfer of control, you can use error flags, file attributes, or other job attributes. The execution control commands and a brief description of each appear in the following list:

<u>Command</u>	<u>Description</u>
BEGIN	Initiates processing of a procedure.†
DISPLAY	Evaluates an expression and displays the result in the dayfile of the job.
ELSE	Terminates skipping initiated by a false expression within an IFE command or initiates skipping to a matching ENDIF command.
ENDIF	Terminates skipping initiated by a matching IFE, SKIP, or ELSE command.
ENDW	Establishes the end of the loop.
EXIT	Controls the command flow in the event of errors.
IFE	Skips to the first matching ELSE or ENDIF command if its conditional expression is false. Otherwise, the system processes the next command.
MODE	Specifies the type of errors the system recognizes for EXIT processing.
NOEXIT	Disables EXIT error processing.
ONEXIT	Enables EXIT error processing.
REVERT	Returns processing from a procedure to the command record of procedure that called it.†
SET	Assigns values to special symbolic names.
SKIP	Skips to the first matching ENDIF command.
WHILE	Establishes the beginning of a loop. If the associated expression is true, the loop is processed; if it is false, the loop is not processed.

†Section 4 contains a detailed description of this command.

NOTE

The control language KCL (described in appendix H) was the system control language available under NOS prior to these execution control commands. NOS will drop support of KCL at its next release. Upgrade your KCL-dependent procedures and jobs so that they are no longer KCL-dependent. Unpredictable results may occur if you mix KCL commands with the commands described in this section.

COMMAND SYNTAX

The syntax for these commands is similar to the syntax of all other commands as described in section 5. However, these are some significant differences.

- Functions, arithmetic expressions, relational expressions, and logical expressions can appear in parameters of certain commands.
- A right parenthesis ending an expression within a command cannot also serve as the command terminator. You must include an additional right parenthesis or period to terminate the command.
- Parentheses can nest expressions within expressions.
- A parameter can consist of more than seven characters.
- A command can be longer than 80 characters if its parameter specifications require such. It can extend over more than one line if each line to be continued contains no more than 80 characters and ends with a separator.

The following subsections describe the command syntax, including the operators, operands, and functions which make up valid expressions. Following that is a discussion of each command.

OPERATORS

Operators separate operands in an expression. There are three types of operators: arithmetic, relational, and logical. Operators are used in the expressions within the IFE, WHILE, DISPLAY, and SET commands and the FILE function.

ARITHMETIC OPERATORS

Integer arithmetic is used in each step of the evaluation of an expression. Division, multiplication, and exponentiation produce a zero result if the absolute value exceeds $2^{48}-1$. Computations are accurate to 10 decimal digits (20 octal digits) and overflow is ignored.

The following are the arithmetic operators.

<u>Operator</u>	<u>Operation</u>
+	Addition.
-	Subtraction.
*	Multiplication.
/	Division.
**	Exponentiation.
Leading -	Negation.
Leading +	Ignored.

RELATIONAL OPERATORS

A relational operator produces a value of 1 if the relationship is true, and 0 if it is false. The following are the relational operators (either form may be used).

<u>Operator</u>	<u>Operation</u>
= .EQ.	Equal to.
.NE.	Not equal to.
< .LT.	Less than.
> .GT.	Greater than.
.LE.	Less than or equal to.
.GE.	Greater than or equal to.

LOGICAL OPERATORS

When an expression contains a logical operator, the system evaluates the full 60 bits of each operand and produces a 60-bit result. If the result has any bits set, it is true (nonzero); if no bit is set, the result is false (zero). The following are the logical operators.

<u>Operator</u>	<u>Operation</u>
.EQV.	Equivalence.
.OR.	Inclusive OR.
.AND.	AND.
.XOR.	Exclusive OR.
.NOT.	Complement.

ORDER OF EVALUATION

The order in which operators in an expression are evaluated is:

1. Exponentiation.
2. Multiplication, division.
3. Addition, subtraction, negation.
4. Relations.
5. Complement.
6. AND.
7. Inclusive OR.
8. Exclusive OR, equivalence.

Operators of equal order are evaluated from left to right.

OPERANDS

One or more operands separated by operators make up an expression. Expressions are used within the IFE, WHILE, DISPLAY, and SET commands. An expression within an expression must begin with a left parenthesis and end with a right parenthesis. There is no limit on the length of an expression, except that a period or a right parenthesis (not acting as a command terminator) must appear within the first 50 operands. Expressions can contain operands of one or more types. There are three types of operands; constants, symbolic names, and functions.

CONSTANTS

A constant is a string of 1 to 10 characters that the system processes as an integer. The constant can be a numeric string or a literal.

Numeric Strings

Each character in the string must be a digit (0 through 9), except the final character. The final character can be a postradix of D or B to indicate the number base of the constant. A postradix of B denotes an octal integer. D or an omitted postradix denotes a decimal integer.

Literals

A literal is a \$-delimited string of characters or a string containing a \$-delimited string. The literal can be a null string (\$\$). The maximum length for a literal is 10 characters.

Special characters can appear in literals but they must appear in the \$-delimited portion of the string. To represent a dollar sign within a literal, you must use double dollar signs.

<u>Valid Literal</u>	<u>Invalid Literals</u>
\$LITERAL1\$	LITERAL1
\$5LITERAL\$	\$LITERAL2
\$*LITERAL\$	LITERAL3\$
\$\$\$LITERAL\$	*\$LITRAL\$
\$\$\$\$	\$\$*LITERAL\$
	\$\$LITERAL\$
	\$\$\$

When the literal appears as an operand, the system processes it as an integer. The system right-justifies the display code values of the string.

Example:

```
/display,$lit$
    49748   141124B
/display,$a$
    1     1B
/display,1
    1     1B
/display,$1$
    28    34B
/display,$$$
    43    53B
/display,literal
CCL157- UNKNOWN NAME - LITERAL
/
```

As noted in the previous example (DISPLAY,LITERAL), nonnumeric strings that are not \$-delimited cannot appear as operands. If they appear as such, the system treats them as erroneous symbolic names. You can, however, use such strings as parameter values for procedures.

SYMBOLIC NAMES

A symbolic name is a system-defined string of characters to which the system or you can assign a value. The symbolic names represent job or system attributes. You can ascertain and sometimes change these attributes by using these symbolic names (refer to the various commands and functions described later in this section).

Most symbolic names have an initial value of 0. The exceptions are OT (job origin type), SYS (host operating system), HID (machine identifier), VER (operating system version number), TIME (current time of day), and SS (subsystem for an interactive job).

The following lists contain the valid symbolic names and a brief description of each. The lists do not contain the symbolic names you can use with the FILE and DT functions. Those symbolic names appear with the descriptions of the functions.

- Symbolic names whose values are passed to, but not from, a procedure (refer to section 4). When a procedure reverts, they are restored to the values they held when the procedure was called.

<u>Name</u>	<u>Description</u>
DSC	Flag determining whether skipped commands are entered in the dayfile (refer to SET Command in this section).
EF	Previous error flag.
R1	Control register 1 contents.
R2	Control register 2 contents.
R3	Control register 3 contents.

- Symbolic names whose values you can set. All except EM are set by the SET command or the SETJCI macro (refer to Volume 4, Program Interface).

<u>Name</u>	<u>Description</u>
DSC	Flag determining whether skipped commands are entered in the dayfile.
EF	Previous error flag.
EFG	Global error flag.
EM	Current exit mode (refer to MODE Command, later in this section).
R1	Control register 1 contents.
R1G	Global control register 1 contents.

<u>Name</u>	<u>Description</u>
R2	Control register 2 contents.
R3	Control register 3 contents.
SS	Subsystem for an interactive job.

- Symbolic names whose values are set by the operating system.

<u>Name</u>	<u>Description</u>
CMN	Central memory RFL setting divided by 100g (refer to RFL Command in section 7).
DSC	Flag indicating that skipped commands are to be entered in the dayfile.
ECN	Extended memory RFL setting divided by 100g (refer to RFL Command in section 7).
EF	Previous error flag.
FL	Current CM field length.
HID	Two-character machine identifier (display code).
MFL	Maximum CM field length.
MFLL	Maximum extended memory field length.
OT	Job origin type.
SYS	Host operating system.
TIME	Current time of day (hhmm).
VER	Operating system version number (display code).

- Symbolic name whose value is set by the calling or termination of a procedure.

<u>Name</u>	<u>Description</u>
PNL	Procedure nesting level (0 when processing the original command record, 1 when processing a first level procedure, and so forth). Its maximum value is 50.

- Symbolic name whose value can be set by the termination of a procedure (refer to SET Command in this section).

<u>Name</u>	<u>Description</u>
EFG	Global error flag.

- Symbolic names corresponding to error code values. In an expression, you can check the error flag (EF) for a nonzero value; a nonzero value indicates an error, and a zero value indicates no error. For detailed error examination, you can compare EF with a particular symbolic name or its error code value. You are encouraged to use the symbolic name, because the numeric values can change in future releases of NOS. The following list contains the errors that allow exit processing.

<u>Name</u>	<u>Value (Octal)</u>	<u>Description</u>
TIE	1	User break 1.
TAE	2	User break 2.
ARE	3	Arithmetic error.
ITE	4	SCP invalid transfer address.
PSE	5	Program stop error.
PPE	6	PPU abort.
CPE	7	CPU abort.
PCE	10	PPU call error.
	11	Reserved for installation.
MLE	12	Message limit.
TLE	13	Time limit error.
FLE	14	File limit error.
TKE	15	Track limit error.
SRE	16	SRU limit error.
FSE	17	Forced error.
	20	Reserved for CDC.
ODE	21	Operator drop.
IDE	22	Idledown.
SPE	23	Beginning of special errors.
RRE	24	Operator rerun.
DRE	25	Deadstart rerun.
STE	26	Suspension timeout.
OKE	27	Operator kill.
TJE	30	Terminate job.
SVE	31	Security violation.

<u>Name</u>	<u>Value (Octal)</u>	<u>Description</u>
SSE	32	Subsystem aborted.
ECE	33	Extended memory parity error.
PEE	34	CPU parity error.
SYE	35	System abort.
RAE	36	Recovery abort.
ORE	37	Override error condition.
MXE	40	Maximum number of error flags.

- Symbolic names with fixed values that can be compared with the origin type (OT) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
BCO	1	Local batch origin.
EIO	2	Remote batch origin.
SYO	0	System origin.
TXO	3	Interactive origin.

- Symbolic name with a fixed value that can be compared with the host operating system (SYS) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
NOS	6	Network Operating System.
NOSB	4	Network Operating System/Batch Environment.

- Symbolic names with a fixed value that you can compare with the subsystem for an interactive job (SS).

<u>Name</u>	<u>Description</u>
ACCESS	The access subsystem.
BASIC	The BASIC subsystem.
BATCH	The batch subsystem.
EXECUTE	The execute subsystem.
FORTRAN	The FORTRAN 5 subsystem.
FTNTS	The FORTRAN Extended 4 subsystem.
NULL	The null subsystem.

- Symbolic names with true or false values. True is 1; false is 0.

<u>Name</u>	<u>Description</u>
F	Fixed value of 0 (false).
FALSE	Fixed value of 0 (false).
SWn	One of six sense switches (n can be from 1 to 6). Their values are set by the OFFSW, ONSW, and SWITCH commands (refer to section 7).
T	Fixed value of 1 (true).
TRUE	Fixed value of 1 (true).

FUNCTIONS

Functions are used as expressions or operands within expressions in commands. Functions are not commands. The functions are FILE, DT, and NUM.

FILE Function

The FILE function determines whether a file has a specified attribute. The system returns a value of 1 (true) or 0 (false) depending upon whether the file has or does not have the specified attribute(s). Only the equipment number (EQ) attribute can return values other than 1 or 0. The list of file attributes follows the description of the FILE function format.

The FILE function must be used as an expression or as a part of an expression in a command. A left parenthesis must appear before the file name, a comma must appear between the file name and the expression, and a right parenthesis must appear after the expression.

Format:

FILE(lfn,exp)

<u>Parameter</u>	<u>Description</u>
lfn	Name of the file for which attributes are being determined.
exp	Either a special FILE function attribute or an expression, consisting of logical operators and special FILE function attributes. The expression must be appropriate for the command in which the FILE function appears. If the FILE function is part of an IFE command, the expression should be one that can be evaluated as true or false. If the FILE function is part of the DISPLAY command, the expression could have a numeric value other than a true or false value.

The expression within a FILE function cannot include the NUM function, the symbolic name SS, or another FILE function; the DT function or the following symbolic names can be used within the expression. Any other symbolic name within the expression is treated either as an implicit DT function (refer to DT Function which follows) or as an unidentified variable.

<u>Parameter</u>	<u>Description</u>
AS	File is assigned or attached to your job.
BOI	File is positioned at BOI. This is effective only for a file on mass storage.
EOF	Last operation was a forward operation, which encountered an EOF and is now positioned at that EOF. This is effective only for a file on mass storage.
EOI	Last operation was a forward operation, which encountered an EOI and is now positioned at that EOI. This is effective only for a file on mass storage.
EQ	EST number of the equipment on which the file resides. If the file is not assigned to the job or is empty and local, it has an equipment number of zero.
EX	File has execute permission.
IN	File type is input.
LB	File is on a labeled tape.
LI	File type is library.
LO	File type is local.
MD	File has modify permission.
MS	File is on mass storage.
OP	File is opened.
PM	File is an attached direct access permanent file.
PT	File type is primary.
QF	File type is queued.
RA	File has read append permission.
RD	File has read permission.
RM	File has read modify permission.
TP	File is on magnetic tape.
TT	File is assigned to a terminal.
WR	File has write permission.

Example:

The following job segment shows the FILE function being used inside an IFE command. The FILE function determines if file ACCT is not at the beginning-of-information (BOI). If ACCT is not at BOI, the IFE command is true and the system rewinds ACCT before copying it onto ITEM. If ACCT is at BOI, the IFE command is false and the system skips to the ENDIF command and copies ACCT onto ITEM. In both cases, ACCT is copied to ITEM and is replaced.

```
IFE,FILE(ACCT,.NOT.BOI),LABEL1.  
REWIND,ACCT.  
ENDIF,LABEL1.  
COPY,ACCT,ITEM.  
REPLACE,ITEM.
```

DT Function

The DT function determines the device type on which a file resides. DT can be used only within a FILE function. The value of the DT function is true if the two-character mnemonic included in the function is equal to the two-character device type. The operating system defines the mnemonics.

Format:

FILE(lfn,DT(dt))

<u>Parameter</u>	<u>Description</u>																						
lfn	Name of the file for which device residence is being determined.																						
dt	A two-character mnemonic identifying the device, which may be any one of the following:																						
	<table><thead><tr><th><u>Type</u></th><th><u>Equipment</u></th></tr></thead><tbody><tr><td>DE</td><td>Extended memory.</td></tr><tr><td>DI</td><td>844-21 Disk Storage Subsystem (half-track).</td></tr><tr><td>DJ</td><td>844-41 or 844-44 Disk Storage Subsystem (half-track).</td></tr><tr><td>DK</td><td>844-21 Disk Storage Subsystem (full-track).</td></tr><tr><td>DL</td><td>844-41 or 844-44 Disk Storage Subsystem (full-track).</td></tr><tr><td>DM</td><td>885 Disk Storage Subsystem (half-track).</td></tr><tr><td>DP</td><td>Distributive data path to extended memory.</td></tr><tr><td>DQ</td><td>885 Disk Storage Subsystem (full-track).</td></tr><tr><td>DV †</td><td>819 Disk Storage Subsystem (single density).</td></tr><tr><td>DW †</td><td>819 Disk Storage Subsystem (double density).</td></tr></tbody></table>	<u>Type</u>	<u>Equipment</u>	DE	Extended memory.	DI	844-21 Disk Storage Subsystem (half-track).	DJ	844-41 or 844-44 Disk Storage Subsystem (half-track).	DK	844-21 Disk Storage Subsystem (full-track).	DL	844-41 or 844-44 Disk Storage Subsystem (full-track).	DM	885 Disk Storage Subsystem (half-track).	DP	Distributive data path to extended memory.	DQ	885 Disk Storage Subsystem (full-track).	DV †	819 Disk Storage Subsystem (single density).	DW †	819 Disk Storage Subsystem (double density).
<u>Type</u>	<u>Equipment</u>																						
DE	Extended memory.																						
DI	844-21 Disk Storage Subsystem (half-track).																						
DJ	844-41 or 844-44 Disk Storage Subsystem (half-track).																						
DK	844-21 Disk Storage Subsystem (full-track).																						
DL	844-41 or 844-44 Disk Storage Subsystem (full-track).																						
DM	885 Disk Storage Subsystem (half-track).																						
DP	Distributive data path to extended memory.																						
DQ	885 Disk Storage Subsystem (full-track).																						
DV †	819 Disk Storage Subsystem (single density).																						
DW †	819 Disk Storage Subsystem (double density).																						

† Model 176 only.

<u>Type</u>	<u>Equipment</u>
MT	Magnetic tape drive (seven-track).
NE	Null equipment.
NT	Magnetic tape drive (nine-track).
TT	Interactive terminal.

Example:

The following dayfile segment shows that TAXES is on a nine-track magnetic tape, so it is copied to output and then unloaded. If the DT function was false, TAXES would be unloaded without being copied.

```
14.00.45.IFE,FILE(TAXES,DT(NT)),LABL1.
14.00.46.COPY,TAXES,OUTPUT.
14.00.46.EOI ENCOUNTERED.
14.00.46.ENDIF,LABL1.
14.00.46.UNLOAD,TAXES.
```

NUM Function

The NUM function determines whether a character string is numeric. It evaluates the character string as true (1) if it is numeric or false (0) if it is not. NUM must be used as an expression or as part of an expression.

Format:

NUM(c)

<u>Parameter</u>	<u>Description</u>
c	A string of 1 to 40 characters. If the string contains one or more special characters, it must be delimited by dollar signs (for example, \$**\$). If delimited by dollar signs, the string is always evaluated as nonnumeric.

Example:

The following procedure uses the NUM function to ensure that the passed parameter, NUMBER, is numeric. If a nonnumeric value is passed, the procedure terminates with an appropriate message.

```
.PROC,PROC1*I,NUMBER.
IFE,NUM(NUMBER),QUIT.
WHILE,R1.LE.NUMBER,LOOP.
SET,R1=R1+1.
.
.
.
ENDW,LOOP.
REVERT. PROCESSING COMPLETED
ENDIF,QUIT.
REVERT,ABORT. NONNUMERIC PASSED
```

COMMAND DESCRIPTIONS

Individual descriptions of the commands follow in alphabetic order.

BEGIN COMMAND

Refer to Calling a Procedure in section 4.

DISPLAY COMMAND

The DISPLAY command evaluates an expression and sends the result to the job dayfile in both decimal and octal integer form. The largest decimal value which can be displayed is 10 digits. If the value is larger than 10 digits, GT followed by 9999999999 is displayed. If the value is negative and larger than 10 digits, LT followed by a minus and 9999999999 is displayed. In octal code, numbers as large as 20 digits can be displayed. For an expression larger than $2^{48}-1$, zeros are displayed.

Format:

DISPLAY(exp)

<u>Parameter</u>	<u>Description</u>
exp	An expression.

Example:

The following sample dayfile shows several display operations.

```
15.14.59.DISPLAY,TIME.
15.14.59.    1514    2752B
15.15.07.SET,R1=99.
15.15.21.SET,R2=901.
15.15.28.DISPLAY,R1.
15.15.28.    99     143B
15.15.38.DISPLAY,R1+R2.
15.15.38.    1000   1750B
15.15.47.DISPLAY,3/2.
15.15.47.    1     1B
15.16.04.DISPLAY,2**47.
15.16.04.GT  9999999999  400000000000000000B
15.16.15.DISPLAY,-2**47.
15.16.15.LT -9999999999  -400000000000000000B
15.16.27.DISPLAY,2**48.
15.16.28.    0     0B
15.16.41.DISPLAY,9999999999.
15.16.41.CCL156- STRING TOO LONG - 9999999999
```

The first DISPLAY command displays the value of the symbolic name TIME. The current time given is in the form hhmm. The next six lines demonstrate the use of the R1 and R2 symbolic names. The other DISPLAY commands specify numeric expressions. The integer constant in the final DISPLAY command has more than 10 digits, resulting in an error message.

ELSE COMMAND

The ELSE command performs one of the following functions.

- It terminates skipping initiated by a false IFE command whose label string matches that of the ELSE command. If the label string does not match, the ELSE command is skipped.
- It initiates skipping from the ELSE command to the ENDIF command whose label string matches that of the ELSE command. This happens for a true IFE command.

Neither a SKIP nor an ELSE command terminates skipping initiated by another SKIP or ELSE command.

Format:

ELSE,ls.

<u>Parameter</u>	<u>Description</u>
ls	Label string; 1 to 10 alphanumeric characters beginning with an alphabetic character.

Example:

The following commands use the FILE function to determine if a file named TEST1 is local to the job. If the file is local, it is copied to the OUTPUT file; if it is not, it is assumed to be an indirect access permanent file, and a local copy is obtained and copied to OUTPUT.

If the file is local, each succeeding command, up to the ELSE command, is processed, and the ELSE command initiates a skip to the ENDIF command. If the file is not local, control skips to the ELSE command, and each command succeeding the ELSE command is processed.

```
IFE,FILE(TEST1,LO),LABEL1.  
COPYSBF,TEST1,OUTPUT.  
ELSE,LABEL1.  
GET,TEST1.  
COPYSBF,TEST1,OUTPUT.  
ENDIF,LABEL1.
```

The following dayfile segment results when the preceding commands are processed and TEST1 is not initially a local file.

```
11.33.00.IFE,FILE(TEST1,LO),LABEL1.  
11.33.00.ELSE,LABEL1.  
11.33.00.GET,TEST1.  
11.33.00.COPYSBF,TEST1,OUTPUT.  
11.33.01.END OF INFORMATION ENCOUNTERED.  
11.33.01.ENDIF,LABEL1.
```

The following dayfile segment results when the preceding commands are processed and TEST1 is initially a local file.

```
15.40.19.IFE,FILE(TEST1,LO),LABEL1.  
15.40.19.COPYSBF,TEST1,OUTPUT.  
15.40.21.END OF INFORMATION ENCOUNTERED.  
15.40.21.ELSE,LABEL1.  
15.40.21.ENDIF,LABEL1.
```

ENDIF COMMAND

The ENDIF command terminates skipping initiated by a SKIP, IFE, or ELSE command. In all cases, the label string on the ENDIF command must match the label string on the command that initiates the skipping. If the system encounters an ENDIF command with a nonmatching label string, it ignores that command.

Format:

ENDIF,ls.

<u>Parameter</u>	<u>Description</u>
ls	Label string; 1 to 10 alphanumeric characters beginning with an alphabetic character.

Example:

When the SKIP command in the following sequence of commands is processed, control skips to ENDIF, and none of the commands between SKIP and ENDIF are processed.

```
SKIP,LABEL1.  
:  
commands  
:  
ENDIF,LABEL1.
```

ENDW COMMAND

The ENDW command identifies the end of the WHILE command loop. A command loop is a sequence of commands that may be repeatedly processed. The number of times the loop is processed depends on the evaluation of the expression specified in the WHILE command that begins the loop.

The ENDW command must have a label string that matches the label string specified in the WHILE command that begins the loop.

Format:

ENDW,ls.

<u>Parameter</u>	<u>Description</u>
ls	Label string; 1 to 10 characters beginning with an alphabetic character. The string cannot contain special characters.

Refer to WHILE Command in this section for an example of ENDW command use.

EXIT COMMAND

The EXIT command indicates the position in the command record where processing will resume if an error is encountered or where to terminate normal command processing if an error is not encountered.

Format:

EXIT.

For additional information, refer to the description of the NOEXIT and ONEXIT commands later in this section and to the description of exit processing in section 5.

IFE COMMAND

The IFE command conditionally initiates the skipping of succeeding commands. If the expression in the IFE command is true, the next command is processed. If the expression is false, CCL skips commands until it encounters a matching ELSE or ENDIF command. The commands match when their label strings are identical.

An IFE command must have a matching ELSE or ENDIF command. If the IFE command initiates skipping without a matching terminating command, the system aborts the job step. If the IFE command is in a procedure, the terminating command must also be in that procedure.

Format:

IFE,exp,ls.

<u>Parameter</u>	<u>Description</u>
exp	An expression. The separator following exp must be a comma.
ls	Label string; 1 to 10 alphanumeric characters beginning with an alphabetic character.

Example 1:

The following commands initiate the compilation and execution of a FORTRAN program and then test for any errors during execution. If an error was made, the error code is displayed.

```
FTN5,I=IFTEST.  
SET,EF=0. INITIALIZE ERROR FLAG  
NOEXIT.  
LGO.  
ONEXIT.  
IFE,EF.NE.0,LABL1.  
DISPLAY,EF.  
ENDIF,LABL1.
```

If the job step executes without error, the error flag (EF) is 0. In this case, control passes to the ENDIF command. If an error occurs, the error flag is not 0, the command is true, and control passes to the next command; the system then displays the error code in the error flag register. (The NOEXIT and ONEXIT commands are described later in this section.)

In the following sample dayfile segment resulting from processing of the preceding commands, the FORTRAN program attempts to call a subroutine BETA which does not exist (outside the field length of the job).

```
FTN5,I=FTNPROG.  
SET,EF=0. INITIALIZE ERROR FLAG  
NOEXIT.  
LGO.  
ONEXIT.  
IFE,EF.NE.0,LABL1.  
DISPLAY,EF.  
ENDIF,LABL1.
```

```
12.57.02.FTN5,I=FTNPROG.  
12.57.02. 61000 CM STORAGE USED.  
12.57.02. 0.009 CP SECONDS COMPILATION TIME.  
12.57.02.SET,EF=0. INITIALIZE ERROR FLAG  
12.57.02.NOEXIT.  
12.57.02.LGO.  
12.57.03. NON-FATAL LOADER ERRORS -  
12.57.03. UNSATISFIED EXTERNAL REF -- BETA  
12.57.03. MODE ERROR.  
12.57.03. JOB REPRIEVED.  
12.57.03. UNSATISFIED EXT IN FLAG NEAR LINE 2  
12.57.03. 7500 MAXIMUM EXECUTION FL.  
12.57.03. 0.000 CP SECONDS EXECUTION TIME.  
12.57.03. RPV - PREVIOUS ERROR CONDITIONS RESET.  
12.57.03. CPU ERROR EXIT AT 400121.  
12.57.03. CM OUT OF RANGE.  
12.57.03.ONEXIT.  
12.57.04.IFE,EF.NE.0,LABL1.  
12.57.04.DISPLAY,EF.  
12.57.04. 3 3B  
12.57.04.ENDIF,LABL1.
```

Example 2:

The following procedure file is an indirect access file called COLORPR. It uses the IFE command to determine if the color the BEGIN command substituted for COLOR is red or blue. Different processing is done for the colors red and blue. Any other color is ignored. The # character in the comment line inhibits substitution for the word (COLOR) it precedes (refer to Parameters and the Procedure Body in section 4).

```
.PROC,A*I,COLOR.  
IFE,$COLOR$.EQ.$RED$,L1.  
COMMENT. PROCESSING DONE FOR #COLOR OF COLOR  
REVERT.  
ENDIF,L1.  
IFE,$COLOR$.EQ.$BLUE$,L2.  
COMMENT. PROCESSING DONE FOR #COLOR OF COLOR  
REVERT.  
ENDIF,L2.  
COMMENT. NO PROCESSING FOR #COLOR OF COLOR
```

The following commands call procedure A.

```
BEGIN,A,COLORPR,BLUE.  
BEGIN,A,COLORPR,RED.  
BEGIN,A,COLORPR,PINK.
```

The following dayfile segment results when the preceding commands are processed. It shows the effect of the # character.

```

08.34.30.BEGIN,A,COLORPR,BLUE.
08.34.32.IFE,$BLUES.EQ.$REDS,L1.
08.34.32.ENDIF,L1.
08.34.32.IFE,$BLUES.EQ.$BLUES,L2.
08.34.32.COMMENT. PROCESSING DONE FOR COLOR OF BLUE
08.34.32.REVERT.
08.34.33.BEGIN,A,COLORPR,RED.
08.34.34.IFE,$REDS.EQ.$REDS,L1.
08.34.34.COMMENT. PROCESSING DONE FOR COLOR OF RED
08.34.34.REVERT.
08.34.34.BEGIN,A,COLORPR,PINK.
08.34.35.IFE,$PINK$.EQ.$REDS,L1.
08.34.35.ENDIF,L1.
08.34.35.IFE,$PINK$.EQ.$BLUES,L2.
08.34.35.ENDIF,L2.
08.34.36.COMMENT. NO PROCESSING FOR COLOR OF PINK
08.34.36.$REVERT.CCL

```

MODE COMMAND

The MODE command defines the error conditions that cause the system to exit from normal processing. When the error that you specified occurs, the system sets the appropriate error flag and exits from normal processing to perform any error processing required. If an error occurs for which you did not select the exit mode processing, the system ignores the error and continues normal processing.

Format:

MODE,m,n.

<u>Parameter</u>	<u>Description</u>
m	CPU program error exit mode ($0 < m < 17g$). Modes 10g through 17g are only valid for model 176. Selects the error condition(s) for which normal error processing does and does not occur. If no mode is selected, the default is m=7.
n	Included for compatibility with earlier versions of NOS. The system now ignores the value specified on the command.

The following values can be supplied for m.

<u>m</u>	<u>Normal Error Processing Occurs</u>	<u>Error Ignored and Job Continues</u>
0	None.†	Address out of range, operand out of range, indefinite operand, or underflow.††
1	Address out of range.	Indefinite operand, operand out of range, or underflow.††
2	Operand out of range.†	Address out of range, indefinite operand, or underflow.††
3	Address or operand out of range.	Indefinite operand or underflow.††
4	Indefinite operand.†	Address or operand out of range or underflow.††
5	Address out of range or indefinite operand.	Operand out of range or underflow.††
6	Indefinite operand or operand out of range.	Address out of range or underflow.
7	Indefinite operand, address out of range, or operand out of range (default value).	Underflow.††
10,11†††	Underflow mode or address out of range.	Operand out of range or indefinite operand.
12,13†††	Underflow mode, address out of range, or operand out of range.	Indefinite operand.
14,15	Underflow mode, address out of range, or indefinite operand.	Operand out of range.
16,17	Underflow mode, address out of range, operand out of range, or indefinite operand.	None.

Address out-of-range error is caused by an attempt to reference CM or extended memory outside of established limits, or by an attempt to reference the last 60-bit word (word 7) in the relative address FL of extended memory. Operand out-of-range error is caused by a floating-point arithmetic unit receiving an indefinite operand. An indefinite operand error is caused by a floating-point arithmetic unit receiving an indefinite operand. Dividing zero by zero results in an indefinite operand. Underflow mode error is caused by a floating-point arithmetic unit receiving a negative exponent value that is too small. For further information about the processing of error mode errors, refer to Error Control in section 3 and to the CYBER 170, CYBER 70, and 6000 Computer Systems reference manuals.

†On the model 176, address out of range (m=1) is always selected.

††Underflow is applicable only to the model 176.

†††These modes are valid only on the model 176. Since address out of range (m=1) is always selected, the two modes are equivalent.

NOEXIT COMMAND

The NOEXIT command suppresses EXIT command processing. If an error occurs, control is not transferred to the command following the next EXIT command. Instead, processing continues with the next command (unless the error causes the job to unconditionally terminate). Refer to the description of exit processing in section 5 for more information.

Format:

NOEXIT.

ONEXIT COMMAND

The ONEXIT command reverses the effect of a NOEXIT command. If an error occurs in processing the commands following ONEXIT, control transfers to the command following the next EXIT command. Refer to the description of exit processing in section 5 for further information.

Format:

ONEXIT.

REVERT COMMAND

Refer to Ending a Procedure in section 4.

SET COMMAND

The SET command assigns a value to a control register, an error flag, or the flag that determines whether skipped commands are entered in the dayfile. Using the SS symbolic name, it also can change the current interactive subsystem.

Format:

SET,sym=exp.

sym One of the following symbolic names (initially these names are set to 0, except for SS).

<u>Name</u>	<u>Description</u>
R1, R2, or R3	Local control registers. When a procedure is called, the current values of R1, R2, and R3 are passed to the procedure. The values of these registers may change within the procedure. However, when processing reverts, these registers are restored to the values they had when the procedure was called.
RIG	Global control register. When a procedure is called or reverts, RIG keeps its current value.
EF	Local error flag. When a procedure is called, the current value of the error flag is passed to the procedure. The value of the error flag may change within the procedure. However, when processing reverts, the error flag is restored to the value it had when the procedure was called.
EFG	Global error flag. When a procedure is called or reverts, EFG keeps its current value.
DSC	Dayfile-skipped-command flag. Initially, it is set to 0, so that commands that are skipped (not processed) are not entered in the dayfile. If DSC is 1, the system enters skipped commands in the dayfile with three leading periods.
SS	Interactive subsystem indicator. The default of SS is NULL.

exp Any valid expression. The value derived through evaluation of the expression is assigned to the symbolic name. Acceptable values for each symbolic name follow.

<u>sym</u>	<u>Suggested Value</u>
R1, R2, R3, or RIG	Any integer between -131071 and 131071. If the value is outside this range, it is truncated. The system does not issue a message as a result of the truncation.

<u>sym</u>	<u>Suggested Value</u>
EF or EFG	Any integer between 0 and 63. If the value is greater than 63, it is truncated. To assign the value defined by the system for an error condition, the user should set the error flag to one of the error condition symbolic names (refer to Symbolic Names at the beginning of this section). The system sets the EF flag to the appropriate error code when an error occurs. If EFG is 0 when a REVERT command is processed, the system sets EFG to the value in EF.
DSC	1 or 0. If the value of the expression is nonzero, DSC is set to 1. While DSC is 1, skipped commands are entered in the dayfile preceded by two periods. Some error processing routines set DSC to 1 so that skipped commands are written in the dayfile.
SS	Any subsystem name (ACCESS, BASIC, BATCH, EXECUTE, FORTRAN, FTNTS, or NULL).

Examples:

The first three examples use procedures from the following procedure file. It is an indirect access permanent file with the name SETFILE.

```
.PROC,P1*I.
DISPLAY,R1.
DISPLAY,R1G.
SET,R1=9.
SET,R1G=888.
-EOR-
.PROC,P2*I.
GET,ABC.
DISPLAY,EF.
DISPLAY,EFG.
-EOR-
.PROC,P3*I.
GET,BASIC1.
BASIC.
DISPLAY,EF.
DISPLAY,EFG.
-EOR-
-EOR-
```

Example 1 - Control Register Use:

The following commands (below on the left side) set and display registers R1 and R1G. A procedure, P1, is called which displays these registers, resets them, and then reverts to the command record where they are again displayed.

On the right is the dayfile segment resulting from processing of the commands.

SET,R1=1.	16.34.42.SET,R1=1.
SET,R1G=10.	16.34.42.SET,R1G=10.
DISPLAY,R1.	16.34.43.DISPLAY,R1.
DISPLAY,R1G.	16.34.43. 1 1B
BEGIN,P1,SETFILE.	16.34.43.DISPLAY,R1G.
DISPLAY,R1.	16.34.43. 10 12B
DISPLAY,R1G.	16.34.43.BEGIN,P1,SETFILE.
	16.34.44.DISPLAY,R1.
	16.34.44. 1 1B
	16.34.44.DISPLAY,R1G.
	16.34.44. 10 12B
	16.34.44.SET,R1=9.
	16.34.44,R1G=888.
	16.34.44.\$REVERT.CCL
	16.34.44.DISPLAY,R1.
	16.34.44. 1 1B
	16.34.45.DISPLAY,R1G.
	16.34.45. 888 1570B

The R1 and R1G registers retain their setting when the procedure is called. However, after new values are set in the procedure and control reverts to the command record, R1 returns to its previous value and R1G retains the value set in the procedure.

Example 2 - Error Flag Use (EFG Nonzero):

The following commands (below on left side) set values in the error flags EF and EFG and then call a procedure which attempts to access an indirect access permanent file. Control reverts to the command record where EF and EFG are displayed to see if any error code generated is returned via these flags. On the right side is the dayfile segment resulting from the processing of the commands.

NOEXIT.	16.43.35.NOEXIT.
SET,EF=10.	16.43.35.SET,EF=10.
SET,EFG=20.	16.43.35.SET,EFG=20.
DISPLAY,EF.	16.43.35.DISPLAY,EF.
DISPLAY,EFG.	16.43.35. 10 12B
BEGIN,P2,SETFILE.	16.43.35.DISPLAY,EFG.
DISPLAY,EF.	16.43.35 20 24P
DISPLAY,EFG.	16.43.35.BEGIN,P2,SETFILE.
	16.43.36.GET,ABC.
	16.43.36. ABC NOT FOUND.
	16.43.36.DISPLAY,FF.
	16.43.36. 3 3B
	16.43.36.DISPLAY,EFG.
	16.43.36. 20 24B
	16.43.36.\$REVERT.CCL
	16.43.37.DISPLAY,EF.
	16.43.37. 10 12B
	16.43.37.DISPLAY,EFG.
	16.43.37. 20 24B

The procedure attempts to get a permanent file which does not exist. This changes EF to error code 4. It does not affect EFG. Control reverts to the command record and displays EF and EFG. EF returns to its initial setting; EFG remains unchanged throughout.

Example 3 - Error Flag Use (EFG Zero):

To return the error code generated in a procedure to the command record, EFG must be 0 before there is an exit from the procedure. This is demonstrated by the following commands (below left side).

The dayfile segment (on the right) resulting from processing of the commands shows how the error code is returned.

NOEXIT.	09.42.52.NOEXIT.
SET,EF=10.	09.42.52.SET,EF=10.
BEGIN,P3,SETFILE.	09.42.52.BEGIN,P3,SETFILE.
DISPLAY,EF.	09.42.53.GET,BASIC1.
DISPLAY,EFG.	09.42.55.BASIC.
	09.42.56. INPUT FILE EMPTY OR MISPOSITIONED
	09.42.56.DISPLAY,EF.
	09.42.56. 5 5B
	09.42.57.DISPLAY,EFG.
	09.42.57. 0 0B
	09.42.57.\$REVERT.CCL
	09.42.58.DISPLAY,EF.
	09.42.58. 10 12B
	09.42.58.DISPLAY,EFG.
	09.42.58. 5 5B

The procedure attempts to compile a BASIC program that is not an INPUT record. This generates error code 5 in EF but does not affect EFG while control is still within the procedure. When control reverts to the command record, EF returns to its original setting and error code 5 is set in EFG.

Example 4 - DSC Flag Use:

The following commands (below on left side) demonstrate the effect of DSC=0 and DSC=1. On the right side is the dayfile segment resulting from processing of the preceding commands.

SET,DSC=0.	16.49.36.SET,DSC=0.
SKIP,LABL1.	16.49.36.SKIP,LABL1.
COMMENT. SINCE THE DAYFILE SKIP	16.49.36.ENDIF,LABL1.
COMMENT. CONTROL IS SET TO ZERO.	16.49.37.SET,DSC=1.
COMMENT. THESE STATEMENTS WILL NOT	16.49.37.SKIP,LABL2.
COMMENT. APPEAR IN THE DAYFILE.	16.49.37...COMMENT. SINCE THE DAYFILE SKIP
ENDIF,LABL1.	16.49.37...COMMENT. CONTROL IS NOW SET TO ONE,
SET,DSC=1.	16.49.37...COMMENT. THESE STATEMENTS WILL
SKIP,LABL2.	16.49.37...COMMENT. APPEAR IN THE DAYFILE
COMMENT. SINCE THE DAYFILE SKIP	16.49.37...COMMENT. AND EACH WILL BE FLAGGED
COMMENT. CONTROL IS NOW SET TO ONE,	16.49.37...COMMENT. WITH TWO INITIAL PERIODS.
COMMENT. THESE STATEMENTS WILL	16.49.37.ENDIF,LABL2.
COMMENT. APPEAR IN THE DAYFILE AND	
COMMENT. EACH WILL BE FLAGGED	
COMMENT. WITH TWO INITIAL PERIODS.	
ENDIF,LABL2.	

SKIP COMMAND

The SKIP command initiates unconditional skipping of succeeding commands. Skipping is terminated by an ENDIF command that has a label string matching the label string specified on the SKIP command. Only an ENDIF command, and not an ELSE command, terminates skipping initiated by a SKIP command.

Format:

SKIP,ls.

<u>Parameter</u>	<u>Description</u>
ls	Label string; 1 to 10 alphanumeric characters beginning with an alphabetic character.

An example of the use of the SKIP command is given after the description of the ENDIF command.

WHILE COMMAND

The iterative commands WHILE and ENDW bracket a group of commands into a loop that can be repeatedly processed. The beginning of the loop is identified by a WHILE command and the end by an ENDW command. The ENDW command must have a label string that matches the label string specified on the WHILE command. The loop is repeated as long as the expression in the WHILE command is true. If the expression is initially false, control immediately skips to the ENDW command; if no ENDW command is found, all the remaining commands in the command record are skipped.

Label strings of all WHILE commands within the command record of a job should be unique. Duplication of a label string within a command record or within a procedure can produce unpredictable results. The same label string can be used in a called procedure and in the calling command record or procedure.

Format:

WHILE,exp,ls.

<u>Parameter</u>	<u>Description</u>
exp	An expression. The separator following exp must be a comma.
ls	Label string; 1 to 10 alphanumeric characters, beginning with an alphabetic character.

Example:

The following commands initiate a loop which is repeated five times.

```
SET,R1=0.  
SET,R2=5.  
WHILE,R1.LT.R2,FINISH.  
SET,R1=R1+1.  
DISPLAY,R1.  
:  
:  
:  
ENDW,FINISH.
```

You can vary the number of repetitions by setting different values in R2.

JOB CONTROL COMMANDS

7

The job control commands alter information that controls jobs while in the system and retrieves information concerning the status of jobs. The commands included in this category are:

CFO	MACHINE	RTIME
CHARGE	MFL	SETASL
COMMENT	NORERUN	SETCORE
CTIME	NOTE	SETJOB
DAYFILE	OFFSW	SETJSL
DROP	ONSW	SETPR
ENQUIRE	PASSWOR	SETTL
ENTER	PAUSE	STIME
GO	PROTECT	SUBMIT
HTIME	QGET	SUMMARY
Job	RERUN	SWITCH
LDI	RESOURC	UPROC
LENGTH	RFL	USECPU
LIMITS	ROLLOUT	USER

You must have specific authorization to use LDI, PASSWOR, PROTECT, or SUBMIT. You can use the remaining commands regardless of your authorization. A listing of validation information can be obtained using the LIMITS command. Although you are allowed to change several control values for your job (with the Job, RFL, SETPR, and SETTL commands), you can never specify more than that for which you are validated.

The system uses the USER command and CHARGE command for checking your authorization and system accounting information. The RESOURC command is used by the system to prevent deadlocks from occurring when several tapes or packs are used concurrently.

CFO COMMAND

The CFO command sends data to one of your executing jobs. The command places data in locations RA + 70B through RA + 74B of the program's field length. The CPU program receiving the data must set the CFO flag (bit 14 of RA + 0) to indicate it will accept CFO data. The CFO command clears the CFO and the PAUSE flags in RA after it places data in the job's field length.

Format:

CFO,jsn.data

<u>Parameter</u>	<u>Description</u>
jsn	The job sequence name (JSN) of the job to receive the data.
data	The data for the job. The data must be 36 characters or less.

CHARGE COMMAND

The CHARGE command causes the system to record on the account dayfile all information regarding resources used in the previous account block, and designates a new charge and project number for subsequent activity. Its purpose is to control the accounting activity of the system for a customer or the installation. An account block is that portion of a job from one CHARGE command to the end of the job or to another CHARGE command.

Format:

CHARGE,chargenumber,projectnumber.

or

CHARGE,*.

or

CHARGE.

<u>Parameter</u>	<u>Description</u>
chargenumber	Specifies the charge number assigned to you by your site.
projectnumber	Specifies the project number assigned to you by your site.

If you use the third format of this command, the system reads the charge number and the project number from a record of file INPUT. This record must be a single line with the following format:

chargenumber,projectnumber

If your job is an interactive job, the system provides some security by overprinting an area on your terminal and then prompting you to enter your charge and project numbers in this area.

If you use the second format of this command, the system uses your default charge and project numbers (refer to the LIMITS command in this section). If you do not have default charge and project numbers and your site requires that you enter a CHARGE command, the system reads your charge and project numbers from file INPUT in the same fashion as for the third format of the command.

The CHARGE command is used in conjunction with user accounting control. An installation which implements this feature can impose limits on the SRUs your jobs may accumulate or can restrict your access to the system to a certain time-of-day interval.

If your site requires that you use a CHARGE command (refer to the LIMITS command in this section), you must enter a CHARGE command immediately after each USER command in your batch jobs. For an interactive job, the system automatically enters a CHARGE(*) for you during login. Thus, you do not need to enter a CHARGE command by name although you may have to enter your charge and project numbers during login.

The system processes your user prologue (refer to the UPROC command in this section) only after it has processed a CHARGE command. Thus, you may want to insert a CHARGE(*) command in your batch jobs (even if your site does not otherwise require it) to ensure that your user prologue is executed.

If you are assigned more than one charge and/or project number, you can include additional CHARGE commands in your job to record resources used under each charge number/project number combination. Whenever a new CHARGE command is issued, the SRU information for the previous charge number/project number is written to an account dayfile and then cleared. However, the other accumulators (central processor time, mass storage activity, and so on) are not cleared but continue to increment.

For a complete list of accounting messages issued to the job's dayfile, refer to Job Completion in section 3.

COMMENT COMMAND

The COMMENT command allows you to place a comment in the system dayfile and the dayfile of any of your jobs.

Format:

COMMENT,jsn.comment

or

COMMENT.comment

or

*comment

<u>Parameter</u>	<u>Description</u>
jsn	Specifies the job sequence name (JSN) of the job in whose dayfile you want the comment written. The default is the current job.
comment	Specifies the message you want written in the designated dayfiles.

If you use the third format of the command, the asterisk must be the first nonblank character of the command.

CTIME COMMAND

The CTIME command requests that the accumulated CPU time for the job be issued to the job's dayfile (in seconds).

Format:

CTIME.

DAYFILE COMMAND

The DAYFILE command causes the system to write the job's dayfile to the file specified.

Format:

DAYFILE,L=lfn,FR=string,OP=op,PD=pd,PL=pl,I=infile.

or

DAYFILE,lfn,string,op,pd,pl,infile.

<u>Parameter</u>	<u>Description</u>
L=lfn	File on which the dayfile is to be written. If L=lfn is omitted, OUTPUT is assumed. Pagination will occur if listing file name is OUTPUT or if the PD or PL parameters are specified.
FR=string	Specifies a character string for which a search is to be made in the dayfile. The op parameter specifies the field to be searched. The string specified must begin with the first character in the field to be searched. The time field begins with a blank. If the string contains characters other than letters and numbers (such as blanks), it must be enclosed by \$ delimiters. A \$ within the string must be entered twice (\$\$) so it is not interpreted as a delimiter. Interactive commands entered in the dayfile are preceded by a \$ unless you were under the batch subsystem when you issued the command. To search for a interactive command, the \$ preceding the command is replaced with two \$'s, and the command is enclosed by \$ delimiters (for example, \$\$SOLD\$). If the string is found, the portion of the dayfile following the last occurrence of the specified string is output. If the string is not found, an informative message and the entire dayfile is output for jobs where the output file is connected to a terminal; where the output file is not connected to the terminal, the job receives only the informative message.
OP=op	Selects search option (single character):

<u>op</u>	<u>Description</u>
T	Search time field for string specified by FR=string.
M	Search message field for string specified by FR=string.
I	Incremental dump (dayfile printed from point of last dayfile command).
F	Full dump.

If a literal string (string) is specified and op is omitted, OP=M is assumed. If both string and op are omitted, OP=F is assumed unless the output file is connected to a terminal. In this case, OP=I is assumed.

<u>Parameter</u>	<u>Description</u>
PD=pd	Print density (three, four, six, or eight lines per inch); if PD=pd is omitted, PD=6 is assumed.
PL=pl	Selects page size; if PL=pl is omitted, page size is determined from print density.

<u>pd</u>	<u>Assumed pl</u>
3	30
4	40
6	60
8	80

I=infile A copy of a dayfile is to be used for input. If I=infile is omitted, the DAYFILE command uses the active dayfile for input.

NOTE

A paginated dayfile listing cannot be used as the input file (I=infile). Refer to L=lfm parameter.

Examples:

DAYFILE,, \$ 11.21.\$,T.

The dayfile is dumped to OUTPUT starting at the last occurrence of 11.21. in the time field.

DAYFILE,I=DAY,FR=\$\$\$GET,STATS.\$

The dayfile is dumped to OUTPUT starting at the last occurrence of \$GET,STATS. in the message field of the previously written dayfile named DAY.

The system may place diagnostic messages in the output produced by DAYFILE. They are not part of the job's dayfile from which the DAYFILE output is produced. These messages begin with NOTICE*** and are described in appendix B.

DROP COMMAND

The DROP command drops executing jobs or queued files. You can drop only jobs or queued files belonging to you. You cannot drop the job from which you enter the DROP command.

A job or queued file belongs to you if it is your current job, was created by your current job, or was created by another job belonging to you. Thus, any job you submit from your current job belongs to you, even if the USER command in the submitted job has a user name different from the user name you specified when you logged in (or different from that of the first USER command if the current job is a batch job). In the case of remote batch jobs, you are the owner of the jobs and files generated at a remote batch terminal if your user name was used in the login for that session.

Format:

DROP,JSN=jsn,DC=q,UJN=ujn.

or

DROP,jsn,q,ujn.

The parameters are order-independent if you specify the parameters in the keyword=value form. Otherwise, they are order-dependent.

<u>Parameter</u>	<u>Description</u>
JSN=jsn	Specifies the job sequence name (JSN) of the job to be dropped.
UJN=ujn	Specifies the user job name (UJN) of the job to be dropped. You can specify the JSN, the UJN, or both.
DC=q	Specifies the disposition of the job to be dropped. You must specify q if neither the JSN nor the UJN is specified. In this case, the system drops all of your jobs with the specified disposition. If JSN or UJN is specified, the default for q is EX.
<u>q</u>	<u>Jobs</u>
WT	Jobs queued with a wait disposition.
PR	Jobs queued for printing.
PU	Jobs queued for punching.
PL	Jobs queued for plotting.
IN	Jobs queued for input.
EX	Executing jobs.
ALL	All jobs belonging to you.

Examples:

DROP,ABDA Drops executing job ABDA.

DROP,,PR. Drops all of your jobs queued for printing.

DROP. Invalid. You must specify a JSN, UJN, or the DC=q parameter.

ENQUIRE COMMAND

The ENQUIRE command gives you information about your jobs in the system.

Format:

ENQUIRE,OP= $p_1p_2\dots p_n$,FN= lfn_1 ,O= lfn_2 .

or

ENQUIRE, $p_1p_2\dots p_n$.

or

ENQUIRE,JSN= jsn ,O= lfn_2 .

or

ENQUIRE,UJN= ujn ,O= lfn_2 .

or

ENQUIRE.

Parameter

Description

OP= $p_1p_2\dots p_7$

Any combination of the following options. You can request a maximum of seven options on a command (for example, ENQUIRE,OP=BDFJLRS.).

P_i

Description

A

Gives listings of the B, D, R, U, J, L, and F options. If you specify no parameters or only the O= lfn_2 parameter in a batch job, the OP= $p_1p_2\dots p_n$ parameter defaults to OP=A.

B

Returns identification and priority information.

Example:

SYSTEM ACTIVITY.

USER NUMBER	SAH3333
USER INDEX HASH	AOUY
JOB SEQ. NO.	ACIE
FAMILY	CLS127
PACKNAME	*NONE*.
PRIMARY FILE	*NONE*.
SUB SYSTEM	NULL.
CPU PRIORITY	30
MAX FL (CM)	203700
MAX FL (EC)	0
LAST FL (CM)	0
LAST FL (EC)	0

Parameter

Description

Pi

Description

D Returns a listing of the resources you have demanded and those which have been assigned (refer to RESOURC Command in this section).

Example:

RESOURCE DEMAND INFORMATION.

RESOURCE	DEMAND	ASSIGNED
PE	1	0
HD	1	0

F Gives the status of files assigned to your job. An asterisk (*) after the file type indicates that the file is locked. (You cannot write on a locked file.) Refer to the FILE function in section 6 for the meaning of the file type mnemonics. The Status column indicates the current position of the file (I/C indicates mid-record) and whether the last operation performed was a write. The FS column indicates whether a file has the no-auto-drop status (refer to the SETFS command).

Example:

LOCAL FILE INFORMATION.

FILENAME	LENGTH/PRUS	TYPE	STATUS	FS
INPUT*	1	IN.*	EOR	NAD
SYSMOD	23	LO.	BOI	
DECK	2	PT.	BOI	
COMMON	1	LO.	BOI	NAD
OUTPUT		LO.		

TOTAL = 5

Parameter

Description

Pi

Description

J

Returns the contents of your control registers, error flag field, and succeeding commands.

If the J option is used within a procedure, only the next commands in the procedure are listed.

Example:

JOB CONTROL REGISTERS.

R1 = 1
R2 = 0
R3 = 0
EF = 0
EFG = 0
R1G = 0

CONTROL STATEMENT(S)

GET,ALPHA.
COPYSBF,ALPHA,.
EOR

L

Returns your loader information including the status of CYBER Interactive Debug utility (refer to the CYBER Loader Reference Manual).

Example:

LOADER INFORMATION.

MAP OPTIONS = DEFAULT
DEBUG = OFF
GLOBAL LIBRARY SET IS -
LIB3

R

Returns your amount of resources used. The resources listed include CPU time, mass storage activity, magnetic tape activity, permanent file activity, and adder activity. These statistics are factors used in calculating SRUs used.

Example:

RESOURCES USED.

CPU TIME	0.107 SECS.
MS ACTIVITY	0.914 KUNS.
MT ACTIVITY	0.000 KUNS.
PF ACTIVITY	0.034 KUNS.
ADDER	0.003 KUNS.
SRU	3.157 UNTS.

Parameter

Description

Pi

Description

S Returns your accumulated SRUs. The SRU represents your total usage of the system. This unit is derived from central processor time, I/O activity, adder activity, and memory usage.

Example:

SRU ACCUMULATOR.

SRU 3.162 UNTS.

T Returns accumulated CPU time.

Example:

CPU ACCUMULATOR.

CPU TIME 0.111 SECS.

U Returns the initial amount of resources available to you in seconds, job step SRU, account block SRU, and the remaining resources available for dayfile messages, commands, and mass storage.

These values apply to the current job and were set at the beginning of the job by default or by the SETJSL, SETASL, or SETTL commands. As the job progresses, the system increments the appropriate resource counters. Secondary USER commands do not affect these values. Entering a different CHARGE command affects your account block limit (refer to the CHARGE command).

Example:

RESOURCE USAGE ALLOWED.

SECONDS	64
JOB STEP SRU	31808
ACCOUNT BLK SRU	31808
DAYFILE MESSAGES	993
CONTROL STATMTS	NO LIMIT
MASS STORAGE	59008

Parameter

Description

UJN=ujn

Returns the job sequence name (JSN), service class (SC), user job name (UJN), and the current status of the job with the specified UJN. If the job is executing, the report includes the executing message.

Example:

enquire,ujn=ajla

JSN	SC	UJN	STATUS	EXECUTING MESSAGE
AAFA	T	AJLA	EXECUTING	
AACV	B	AJLA	WAIT QUEUE	

If you only specify the keyword UJN, the system returns the information described earlier for all jobs associated with your user name.

Example:

enquire,ujn

JSN	SC	UJN	STATUS	EXECUTING MESSAGE
AAFA	T	AJLA	EXECUTING	
AAFG	R	OUTPUT	PRINT QUEUE	
AAFF	B	OUTPUT	WAIT QUEUE	
AACV	B	AJLA	WAIT QUEUE	
AADC	B	EXAMPLE	WAIT QUEUE	

You may not specify the UJN=ujn parameter with any parameter other than O=1fn2.

JSN=jsn

Returns a detailed report of the job with the specified job sequence name (JSN). The report varies with the status of the job. If the job is executing, the report includes the JSN, service class, user job name (UJN), the current status of the job, job resource information, executing message a dayfile extract (maximum of 10 messages), and a list of the next commands to be executed (maximum of 10 commands).

Example:

enquire,jsn=abkp

```

ABKP.B.EXECUTING          UJN=EXAMPLE.
SRUS=      4.213.  SRU LIMIT=NO LIMIT.  CM FL=  7200.  ECS FL=   0.

```

DAYFILE=

```

12.48.39.EXAMPLE.
12.48.39.USER,AAVVV54,.
12.48.39.CHARGE,4532,564B123.
12.48.40.GET,RECOVER.
12.48.40.BASIC,I=RECOVER.
12.48.41.          25700B CM USED
12.48.41.          .008 CP SECONDS COMPILATION TIME
12.48.41.RECOVER --007200

```

```

NEXT CONTROL STATEMENTS=
ITEMIZE,RECOVER,L=ITEMZ.

```

Parameter

Description

If the job is a queued file, the report includes the job's service class, queue residence, UJN, length in PRUs, and output specifications.

Example:

enquire,jsn=aaep

AAEP.R.PRINT QUEUE UJN=SAMPLE .
PRU LENGTH= 7. DC=LP. FC= . ID= 0. EC= .

If you only specify the keyword JSN, the system returns the JSN, service class, and job status of all jobs associated with your user name (refer to the USER command).

Example:

enquire,jsn

```

      JSN  SC   STATUS                JSN  SC   STATUS
AAFA.T.EXECUTING                    AAFG.R.PRINT QUEUE
AAFF.B.WAIT QUEUE                   AACV.B.WAIT QUEUE
AADC.B.WAIT QUEUE
```

You may not specify the JSN=jsn parameter with any parameter other than O=lf_{n2}.

FN=lf_{n1}

Returns the status of local file lf_{n1} (refer to the OP=F parameter).

O=lf_{n2}

Writes the output on file lf_{n2}. The default is OUTPUT. If you specify only the O=lf_{n2} parameter in a batch job, the OP=p₁p₂...p_n parameter defaults to OP=A.

If you enter the ENQUIRE command with no parameters from an interactive job, the system gives a two-line report of the current job.

Example:

enquire

JSN: AAFA SYSTEM: NULL SRU: 2.055 FILE NAME: SAMPOUT
STATUS: IDLE

It always returns the status IDLE. To get a report on the current job while the system is executing another command, enter the cE or cS command (refer to Interactive Status Commands in section 8).

If you enter the ENQUIRE command with no parameters in a batch job, the command defaults to ENQUIRE,OP=A.

ENTER COMMAND

The ENTER command enters a series of commands on one line. This is especially useful for interactive users operating in the batch subsystem.

Format:

```
ENTER./command1/command2/.../commandn
```

/ Delimiting character used to separate the individual commands on one line. It can be any character not used within the commands and must immediately follow a period or right parenthesis.

command_i Any NOS command you are authorized to use. Interactive commands for which there are no batch counterparts are not acceptable.

The system supplies a terminator (period or right parenthesis) if it is missing from any command.

Example:

From a terminal, you enter the batch subsystem and type in an ENTER command on one line as follows:

```
batch
RFL,0.
/enter.#get,fprog#ftn5,i=fprog#map,part#lgo#exit#dmp#rewind,zzzdump#copy,zzzdump
```

This is the sequence of commands used in section 14 to illustrate the reading of CM dumps. Assuming that a FORTRAN 5 program is on the indirect access file FPROG, output like that shown in the figures in section 14 is produced when this command is processed.

GO COMMAND

The GO command clears the pause bit of one of your executing jobs. The system may have set the pause bit because of an error or a request for user input. You can set the pause bit with one of your programs or with the PAUSE command.

Format:

```
GO,jsn.
```

jsn Specifies the job sequence name (JSN) of the job whose pause bit you want to clear.

NOTE

You must check the status of the pause bit to effectively use this command. The program does not pause just because the pause bit is set. Similarly, the program does not proceed just because the pause bit has been cleared. You can write a program that checks the status of the pause bit and waits for that bit to be cleared (refer to the RECALL macro in Volume 4, Program Interface).

HTIME COMMAND

The HTIME command issues a dayfile message giving the model 176 accumulated clock cycle count for the job. A clock cycle on the model 176 is 27.5 nanoseconds. COMPASS instructions require a certain number of clock cycles to execute as described in the COMPASS Reference Manual. This command can be used for performance comparisons.

Format:

HTIME.

The resulting dayfile message has the following format. The cycle count is in kilocycle units.

HTIME nnnnnnnnnnnn.nnn KCYCLES.

An HTIME command processed on a machine other than the model 176 produces the following dayfile message.

HTIME NOT AVAILABLE.

JOB COMMAND

The Job command, also known as the job card, specifies the user job name (UJN) and may specify job processing parameters. Refer to Job Names in section 3 for the significance of the user job name. The first command of a batch job must be a Job command.

You can issue the Job command in order-independent or order-dependent format. In order-independent format, a separator character does not appear between the keyword and its value. If the order-dependent format is used and parameter values are omitted between separators, the parameter values are interpreted as zeros. A parameter value containing an 8 or 9 must not have a B suffix. If there is a syntax error in the Job command, the system issues an error message and terminates the job. All parameters are optional except the first parameter.

Format:

ujn,Pp,Tt,CMfl,ECfe.cm

or

ujn,p,t,fl,fe.cm

<u>Parameter</u>	<u>Description</u>
ujn	Alphanumeric user job name (UJN) of one to seven characters. It must begin with a letter. This name identifies the individual jobs being run under the same user name.
Pp or p	Priority level at which the job enters the system, ranging from 1 to 7777 ₈ . A value containing an 8 or 9 or the suffix D is interpreted as decimal. This parameter is not used by NOS (refer to Job Scheduling in section 3).
Tt or t	Central processor job step time limit in seconds. Values can range from 1 to 77777. Decimal is the default base. Octal values from 1 to 77777 ₈ (1 to 32767) can be entered if followed by a B suffix. The default limit is 64 (100 ₈). Decimal values from 32767 to 77777 set the time limit at its maximum. The time limit set by this parameter must be sufficient for completion of each of the steps in the job (refer to Time Limit Control in section 3).
CMfl or fl	Maximum octal field length for the job (refer to Field Length Control in section 3). The system rounds the value to the next highest multiple of 100 ₈ . The default field length is the maximum field length for which you are authorized (refer to the LIMITS command) or the maximum field length available for user jobs (dependent on machine size), whichever is smaller. The field length cannot exceed: 377700 ₈ on a 198 K or a 262 K machine 360000 ₈ on a 131 K machine 163000 ₈ on a 65 K machine A value containing an 8 or 9 or the suffix D is interpreted as decimal.

<u>Parameter</u>	<u>Description</u>
ECfe or fe	Maximum octal number of 1000 ₈ word extended memory blocks required by the job. This extended memory field length cannot exceed 3777 blocks or the amount of user extended memory allowed by the installation (refer to the LIMITS command). Your job must request the extended memory (refer to the RFL command in section 7) before it can be used. A value containing an 8 or 9 or the suffix D is interpreted as decimal.
cm	Conversion mode for card decks. The parameter cm must be entered in columns 79 and 80. A 26 indicates that coded cards are to be converted in O26 mode; 29 indicates cards are converted in O29 mode. This initial keypunch mode can be changed within the job by a conversion change card (refer to Coded Cards in appendix F) when reading cards or a ROUTE command when punching cards. If this parameter is omitted, the installation default keypunch mode is used. This conversion mode indicator is ineffective for remote batch jobs entered under mode 4 RBF; it is effective for remote batch jobs entered under HASP RBF.

The system issues error messages when parameter specifications exceed validation limits. It also issues an error message if an extended memory field length is specified when your CM validation limit is less than 10000₈ words. Consult installation personnel for further installation restrictions based on the machine configuration and subsystems used.

Example:

JOBAAA,,1,50000,20.

has the same effect as:

JOBAAA,CM50000,EC20,T1.

LDI COMMAND

The LDI routine queues one or more jobs for input and optionally enters the job sequence name (JSN) of each job in the dayfile of the job that entered the LDI command. LDI initiates a new job each time it encounters an EOF. LDI starts processing at the current position of the file and continues until an EOI, double EOF, or EOF followed by an empty record is encountered. The jobs submitted are batch origin type jobs. LDI does no reformatting of the job file. Therefore, SUBMIT directives (/JOB,/USER,/NOSEQ, and so forth) are not allowed.

Format:

LDI, FN=lfm, ID=id, OP=OP, DC=dc, UN=un, FM=fm.

or

LDI, lfm, id, OP, dc, un, fm.

The parameters in the first format are order-independent. Those in the second format are order-dependent.

<u>Parameter</u>	<u>Description</u>
FN=lfm	Specifies the name of the file containing the job(s) to be submitted. LOAD is the default.
ID=id	Specifies the identifier of the local device which will receive job output. The identifier id must be octal and in the range 0 through 67B. If you omit this parameter or specify only the keyword ID, implicit central site routing occurs (refer to Implicit Routing under the description of the ROUTE command, section 9). You may not specify ID=id parameter if you specify the UN=un or FM=fm parameter.
OP=OP	Directs the system to enter the job sequence name (JSN) of each job in the dayfile of the job that issued the LDI command.
DC=dc	Specifies the disposition of output for each job that is queued for input.

<u>dc</u>	<u>Description</u>
IN	Disposes of job output according to the default option for the job's origin type (refer to Job Origin Types and Service Classes in section 3).
NO	Discards all output.
TO	Queues the output with a wait disposition.

<u>Parameter</u>	<u>Description</u>
UN=un	<p>Specifies the user name of the remote batch user to whom the system routes each job's output. If you omit this parameter or specify only the keyword UN, implicit remote routing occurs (refer to Implicit Routing under the description of the ROUTE command, section 9).</p> <p>You may not specify this parameter if you specify the ID=id parameter.</p>
FM=fm	<p>Specifies the family name of the remote batch user to whom the system routes each job's output. If you omit this parameter or specify only the keyword FM, implicit remote routing occurs (refer to Implicit Routing under the description of the ROUTE command, section 9).</p> <p>You may not specify this parameter if you specify the ID=id parameter.</p>

The number of executing jobs and output files you have within the system cannot exceed your deferred batch job validation limit (refer to the LIMITS command in this section). If this limit is reached, no further jobs can be loaded until an existing job completes.

If any of the submitted jobs contain an invalid USER command, the job entering the LDI command is aborted with no exit processing. If you entered the command interactively, you are immediately logged off with no dayfile message. The security count for the user name that entered the LDI command is decremented accordingly.

LENGTH COMMAND

The LENGTH command gives you the current status of one of your local files.

Format:

LENGTH,lfn.

lfn Name of a file assigned to the job.

The information given for the local file includes its length in PRUs, type, and current status. The same information can be obtained with the ENQUIRE command.

LIMITS COMMAND

The LIMITS command directs the system to list your validation limits. The LIMITS command always lists the validation limits associated with the job's current user name. However, if a job contains more than one USER command, the actual limits associated with the job are a combination of the limits of the first user name and those of the current user name. You can use the ENQUIRE(OP=U) command to determine what your actual limits would be in such a case.

Format:

LIMITS,L=lfm.

lfm The file which receives the validation information. The default is OUTPUT.

The validation limits show the extent to which you can use various system resources. The LIMITS output falls into six categories:

- Resource Limits
- Other Characteristics
- User Permissions
- Network Applications
- Privileged Network Users
- Local Applications

RESOURCE LIMITS

The system indicates the maximum number of specific system resources you can use. The following list shows the resources that appear in this category:

- Magnetic tape units that may be assigned
- Removable auxiliary devices that may be assigned
- CPU seconds allowed for each job step
- Central memory words allowed
- Extended memory words allowed
- Jobs that can be detached
- Deferred batch files allowed
- Permanent files allowed in your catalog
- PRUs available for any one indirect file
- PRUs available for any one direct file
- PRUs available for all indirect files
- PRUs available per job
- Messages that may be issued to the dayfiles
- Batch commands that may be processed
- Cards that may be punched per file
- Lines that may be plotted per file
- Lines that may be printed per file
- SRUs allowed per job

The maximum number of deferred batch jobs does not apply to you if you have system privileges and debug mode is set on the system display console or if your jobs are of the system origin type.

OTHER CHARACTERISTICS

The system lists your user index, default charge and project numbers, the system and user prologues (procedure files that automatically execute whenever you initiate a new job), and the default subsystem and character set mode for interactive jobs. You can specify a user prologue with the UPROC command. Also, you can change the subsystem and character set mode for interactive jobs.

USER PERMISSIONS

In addition to specifying resource limits for certain resources, the system lists special privileges that you possess. The following list contains special privileges you might possess:

- Change your password
- Use privileged interactive commands
- Create direct access files
- Create indirect access files
- Have system origin privileges
- Access/create library files
- Assign magnetic tapes
- Access system without supplying charge/project number
- Access or create files on auxiliary devices
- Access transaction functions for library updates
- Have unlimited terminal connect time without timeout
- Use system control point facility
- Have special accounting privileges
- Have BATCHIO subsystem privileges
- Use the PROTECT command

NETWORK APPLICATIONS

The following are network applications you may be authorized to use:

- Interactive Facility (IAF)
- Remote Batch Facility (RBF)
- Transaction Facility (TAF)
- Message Control System (MCS)
- Terminal Validation Facility (TVF)

PRIVILEGED NETWORK USERS

The system lists the types of privileged network users you are authorized to become. The system considers a network operator (NOP), a local operator (LOP), or a network/local operator (NOPLOP) to be a privileged network user. If you are not authorized to become a privileged network user, this category does not appear in the output.

LOCAL APPLICATIONS

The system lists local applications you are authorized to use. If there are no such applications, this category does not appear in the output.

The following is an example of a user's validation limits:

Limits

MAXIMUM NUMBER OF -

MAGNETIC TAPE UNITS THAT MAY BE ASSIGNED.....	3
REMOVABLE AUXILIARY DEVICES THAT MAY BE ASSIGNED	2
CPU SECONDS ALLOWED FOR EACH JOB STEP/10B.....	UNLIMITED
CENTRAL MEMORY WORDS ALLOWED/100B.....	2037B
EXTENDED MEMORY WORDS ALLOWED/1000B.....	0B
JOBS THAT CAN BE DETACHED.....	3
DEFERRED BATCH FILES.....	8
PERMANENT FILES ALLOWED IN YOUR CATALOG.....	SYSTEM
PRUS AVAILABLE FOR ANY ONE INDIRECT FILE.....	192
PRUS AVAILABLE FOR ANY ONE DIRECT FILE.....	1024
PRUS AVAILABLE FOR ALL INDIRECT FILES.....	4096
PRUS AVAILABLE PER JOB.....	66048
MESSAGES THAT MAY BE ISSUED TO THE DAYFILES.....	1008
BATCH COMMANDS THAT MAY BE PROCESSED.....	UNLIMITED
CARDS THAT MAY BE PUNCHED PER FILE.....	UNLIMITED
LINES THAT MAY BE PLOTTED PER FILE.....	512
LINES THAT MAY BE PRINTED PER FILE.....	UNLIMITED
SRUS ALLOWED PER JOB.....	UNLIMITED

OTHER CHARACTERISTICS -

USER INDEX.....	36515B
DEFAULT CHARGE NUMBER ASSIGNED.....	6913
DEFAULT PROJECT NUMBER ASSIGNED.....	AB51296
USER PROLOGUE.....	STARTUP
INTERACTIVE SUBSYSTEM.....	NULL
INTERACTIVE CHARACTER SET MODE.....	NORMAL

THE FOLLOWING ARE VALID USER PERMISSIONS -

- CHANGE YOUR PASSWORD
- CREATE DIRECT ACCESS FILES
- CREATE INDIRECT ACCESS FILES
- ACCESS/CREATE LIBRARY FILES
- ASSIGN MAGNETIC TAPES
- ACCESS OR CREATE FILES ON AUXILIARY DEVICES

THE FOLLOWING ARE VALID NETWORK APPLICATIONS -

- INTERACTIVE FACILITY (IAF)
- REMOTE BATCH FACILITY (RBF)
- TRANSACTION FACILITY (TAF)
- TERMINAL VALIDATION FACILITY (TVF)

MACHINE COMMAND

Some programs that ran on mainframes other than models 825, 835, and 855 will not run on the models 825, 835, and 855 until changes are made in your job. Programs most likely to run incorrectly on models 825, 835, and 855 are COMPASS programs with tight instruction-modification loops. The MACHINE command will enable some of these programs to run correctly on models 825, 835, and 855.

Format:

MACHINE,EP=state.

state	Specifies whether extended stack purging is selected. EP=ON selects it and EP=OFF disables it.
-------	--

Instruction stack purging is extended by specifying ON and is made normal by specifying OFF and remains in effect until a subsequent MACHINE command or MODE macro alters it. The initial status of stack purging is site-selectable. For more information on stack purging, refer to the MODE macro in Volume 4, Program Interface.

To determine if a program will run correctly on models 825, 835, and 855, run the job with extended stack purging disabled (EP=OFF). If unexpected results occur, do another run with stack purging extended (EP=ON). When stack purging is extended, the execution time of the program may be increased. If unexpected results still occur, recode COMPASS programs or recompile high-level language programs on models 825, 835, and 855.

An attempt to change the stack purging status on any machine other than models 825, 835, and 855 aborts the job.

MFL COMMAND

The MFL command resets the maximum field length for each subsequent job step.

Format:

MFL,CM=nnnnnn,EC=mmmm.

or

MFL,nnnnnn,mmmm.

<u>Parameter</u>	<u>Description</u>
CM=nnnnnn	Specifies the maximum central memory field length (octal is assumed unless decimal is specified by a D suffix or use of the digits 8 or 9).
EC=mmmm	Specifies the maximum extended memory field length. The value of mmmm is the actual extended core field length divided by 1000g.

The first format is the order-dependent form and the second format is the order-independent form.

The parameter nnnnnn sets an upper boundary for the field length of subsequent job steps. The value cannot exceed the maximum field length for the job nor can it be less than 1500, the field length required for the utility (CONTROL) that processes MFL. If extended memory is assigned for the job, the CM field length cannot be less than 10000. Likewise, the parameter mmmm sets an upper boundary for the extended memory field length of subsequent job steps and cannot exceed the maximum field length for the job. If the value 0 (zero) is entered for CM or extended memory field length, the MFL is set to either the maximum field length for which you are authorized or the field length specified on the Job command, whichever is smaller.

The MFL command clears any initial running field length previously established with the RFL command or the SETRFL macro and allows the system to determine the field length for each succeeding job step. The system continues to determine field lengths until another RFL command or SETRFL macro is encountered.

NORERUN COMMAND

The NORERUN command clears the job rerun status.

Format:

NORERUN.

If the NORERUN command has been issued, the job may not be rerun. This may be desirable to prevent updating of a data base when the job would otherwise be rerun by the operator.

This command is ignored from an interactive origin job.

NOTE COMMAND

The NOTE command creates a file containing lines of data entered as a character string on the same line as the command.

Format:

NOTE, lfn, NR./line₁/line₂/.../line_n

<u>Parameter</u>	<u>Description</u>
lfn	Name of the file which contains the specified lines. Default is OUTPUT.
NR	Inhibits rewind of lfn. Default is to rewind the file at the beginning and end of NOTE command execution.
/	Delimiting character used to separate the individual entries that become lines in the file. It can be any character. It must immediately follow a period or right parenthesis. Multiple delimiters generate blank lines.
line _i	A character string which constitutes one line of data in lfn.

If a file contains more lines than can be entered with a single NOTE command, a series of NOTE commands, each with an NR, can be used. This series should be followed with a PACK command since each NOTE command writes an EOR.

Example:

The following sequence of commands creates a procedure file (PFILE) that can insert an input record after any record in an existing master file (LISTFIL).

```
NOTE,PFILE,NR.*PROC,INSERT,N.*GET,LISTFIL.*COPYBR,LISTFIL,NEWLIST,N.  
NOTE,PFILE,NR.*COPYBR,INPUT,NEWLIST.*COPYEI,LISTFIL,NEWLIST.  
NOTE,PFILE,NR.*REPLACE,NEWLIST=LISTFIL.  
PACK,PFILE.  
SAVE,PFILE.
```

To insert an input record after the second record in LISTFIL, the user includes the following command in the job that contains the new input record.

```
BEGIN,INSERT,PFILE,2.
```

OFFSW COMMAND

The OFFSW command clears the sense switches for reference by one of your jobs.

Format:

```
OFFSW,jsn,switch1,switch2,...,switchn.
```

<u>Parameter</u>	<u>Description</u>
jsn	Specifies the job sequence name (JSN) of the job whose sense switch(es) will be changed. The default is the current job. The parameter jsn can appear anywhere in the parameter list.
switch _i	Specifies the switch(es) to be cleared. You must denote each switch switch _i by an integer in the range of 1 through 6. If you specify 0 for this parameter, the system clears all sense switches.

Refer to the description of the ONSW command for further information on sense switch settings.

ONSW COMMAND

The ONSW command sets the sense switches for reference by one of your jobs. The system stores the sense switch settings in your control point area and copies them to RA at the beginning of each job step for use by the central program. The sense switch field in the control point area and the one in RA are updated separately.

Format:

```
ONSW,jsn,switch1,switch2,...,switchn.
```

<u>Parameter</u>	<u>Description</u>
jsn	Specifies the job sequence name (JSN) of the job whose sense switches will be changed. The default is the current job. The parameter jsn can appear anywhere in the parameter list.
switch _i	Specifies the switch(es) to be set. You must denote each switch switch _i by an integer in the range of 1 through 6. If you specify 0 for this parameter, the system sets all sense switches.

You can also use the SWITCH command to set the sense switches.

PASSWOR COMMAND

The PASSWOR command is used to change your password.

Format:

PASSWOR,oldpassword,newpassword.

oldpassword Specifies the old password.

newpassword Specifies the new password.

The new password must be the minimum length required by the installation. The default minimum is four characters; its maximum length is seven characters. Only alphanumeric characters can be used in the password.

For added security, you may issue the PASSWOR command without parameters. In this case, the system reads the parameters from a record in file INPUT. This record must be a single line with the following format:

oldpassword,newpassword

In interactive jobs, the system overprints an area on your terminal and then prompts you to enter the parameters in that area.

You must be authorized to change your password (refer to the LIMITS command in this section).

PAUSE COMMAND

The PAUSE command sets the pause bit of one of your executing jobs.

Format:

PAUSE,jsn.

jsn Specifies the job sequence name (JSN) of the job whose pause bit you want to set.

NOTE

You must check the status of the pause bit to effectively use this command. The program does not pause just because the pause bit is set. Similarly, the program does not proceed just because the pause bit has been cleared. You can write a program that checks the status of the pause bit and waits for that bit to be cleared (refer to the RECALL macro in Volume 4, Program Interface).

PROTECT COMMAND

The PROTECT command is used to activate or deactivate preservation of the extended memory assigned to your job field between job steps.

Format:

PROTECT,EC=state.

or

PROTECT,state. Activates or deactivates extended memory presentation. The two values are ON and OFF. OFF is the default.

Ordinarily, the extended memory assigned to your job is zeroed at the completion of a job step. With EC=ON, it is preserved between job steps.

You must be authorized to use the PROTECT command (refer to the LIMITS command in this section).

QGET COMMAND

The QGET command assigns a queued file to the correct job. You must specify the job sequence name (JSN) or the user job name (UJN) associated with the output. Unless explicitly routed elsewhere, the file remains with your current job.

Format:

QGET,JSN=jsn,DC=q,UJN=ujn,FN=lfn.

or

QGET,jsn,q,ujn,lfn.

<u>Parameter</u>	<u>Description</u>										
JSN=jsn	Specifies the four-character, system-generated JSN associated with the queued file.										
DC=q	Specifies the disposition of the queued file. WT is the default.										
	<table><thead><tr><th><u>q</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>PR</td><td>Print</td></tr><tr><td>PU</td><td>Punch</td></tr><tr><td>PL</td><td>Plot</td></tr><tr><td>WT</td><td>Wait</td></tr></tbody></table>	<u>q</u>	<u>Description</u>	PR	Print	PU	Punch	PL	Plot	WT	Wait
<u>q</u>	<u>Description</u>										
PR	Print										
PU	Punch										
PL	Plot										
WT	Wait										
UJN=ujn	Specifies the UJN associated with the queued file.										
FN=lfn	Specifies the local file name for the queued file. If specified, the JSN is the default local file name. Otherwise, the default is the UJN.										

If you specify neither a JSN nor a UJN or specify a UJN associated with two queued files with the same disposition, a fatal error occurs:

RERUN COMMAND

The RERUN command sets job rerun status.

Format:

RERUN.

If the RERUN command has been issued, the operator can rerun the job if necessary. This command is ignored from a interactive origin job.

RESOURC COMMAND

The RESOURC command is required in any job that uses more than one tape or pack concurrently; it prevents deadlocks with other jobs which may need the same resources.

Format:

RESOURC,rt₁=u₁,rt₂=u₂,...,rt_n=u_n.

<u>rt_i</u>	<u>Resource type:</u>
LO	200-bpi, seven-track magnetic tape unit
HI	556-bpi, seven-track magnetic tape unit
HY	800-bpi, seven-track magnetic tape unit
HD	800-cpi, nine-track magnetic tape unit
PE	1600-cpi, nine-track magnetic tape unit
GE	6250-cpi, nine-track magnetic tape unit
MT†	Seven-track magnetic tape unit
NT†	Nine-track magnetic tape unit (800/1600-cpi)
DIi	844-21 Disk Storage Subsystem (1<i<8)
DJi	844-41/44 Disk Storage Subsystem (1<i<8)
DKi	844-21 Disk Storage Subsystem (full-track)(1<i<8)
DLi	844-41/44 Disk Storage Subsystem (full-track)(1<i<8)
DMi	885 Disk Storage Subsystem (half-track)(1<i<3)
DQi	885 Disk Storage Subsystem (full-track)(1<i<3)
DV	819 Disk Storage Subsystem (single-density)††
DW	819 Disk Storage Subsystem (double-density)††

u_i Maximum number of units of resource type rt_i this job will use concurrently; any rt_i=u_i entry can be changed on subsequent RESOURC commands. (Refer to Altering Resource Requirements.) An rt=0 entry can be entered to clear a resource type that is no longer required.

†Retained for compatibility with NOS 1.2. Refer to restrictions described under Tape Units in this section.

††Model 176 only.

DEADLOCK PREVENTION

The system manages the use of tape units and disk packs so as to prevent deadlocks from occurring. A deadlock means that the system, by assigning a tape unit or pack to one job, prevents another job with currently assigned resources from completing. For example, an installation with two tape units is processing jobs A and B. Each job needs both units during some phase of processing. Job A is assigned unit 1. If job B were assigned unit 2, neither A nor B could complete until the other job relinquishes its assigned unit.

To prevent deadlocks from occurring, the system requires that a RESOURC command be included in any job that uses more than one tape or disk pack concurrently. Thus, in the previous example, a RESOURC command is required in both jobs. The information supplied by the commands enables the system to anticipate the deadlock situation and roll out job B until job A no longer needs both units. When the RESOURC command is executed, the system first checks if the specified number of units exceeds the number of units you are authorized to use or the number of units available at the installation.[†] If either of these situations occurs, the system issues an error message to your dayfile and aborts the job. (Refer to figure 7-1.)

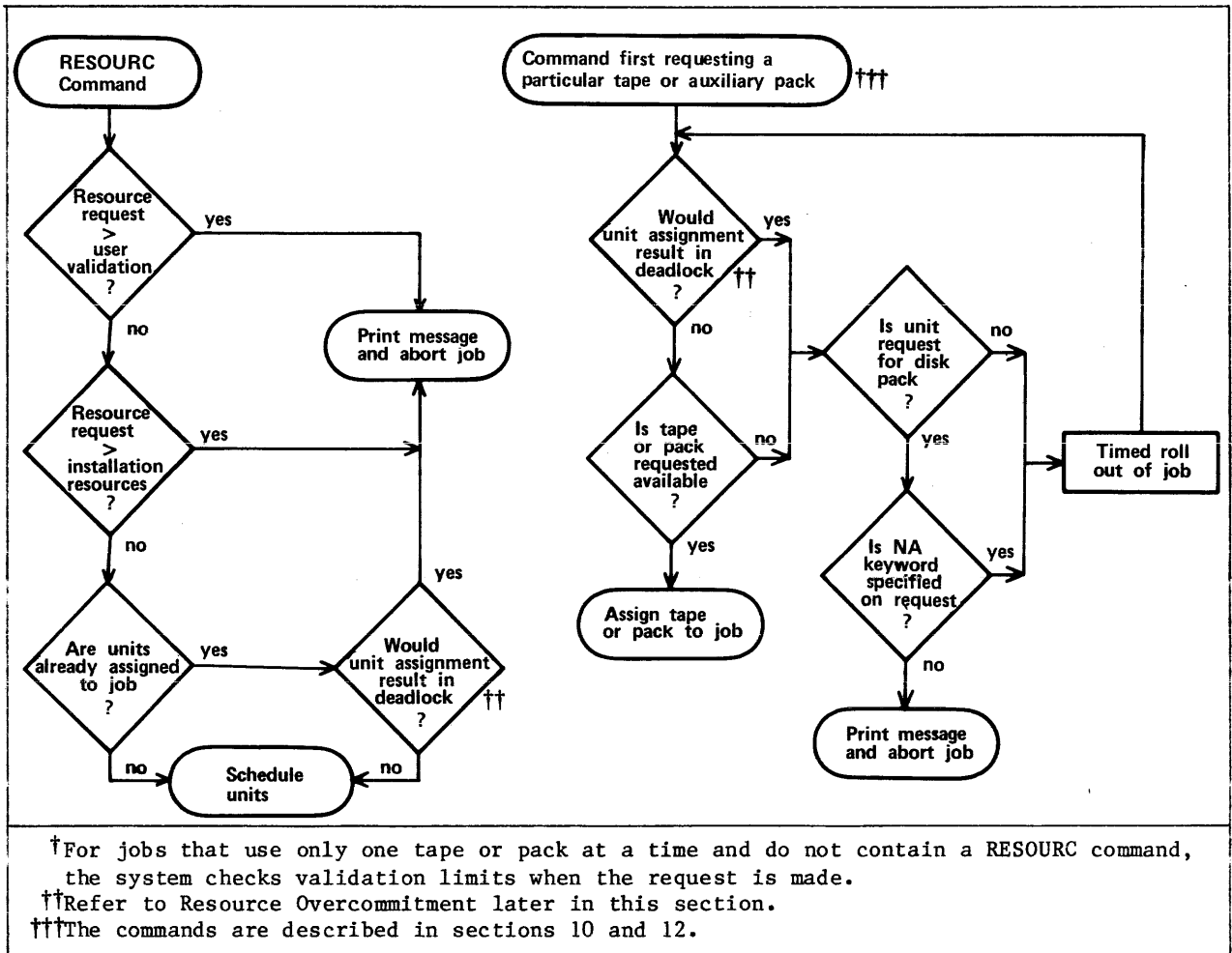


Figure 7-1. Resource Commitment Processing (Simplified)

When the job requests a tape or pack,† the system compares the number of units that jobs being processed have scheduled via RESOURC commands with the number of units actually assigned. If it determines that the assignment would cause a deadlock (refer to Resource Overcommitment), it rolls out the job until a deadlock would not occur. If the assignment would not cause a deadlock, the system searches for the requested tape or pack. If found, it is assigned to the requesting job. If the pack is not found and the NA keyword was included in the request or if the tape is not found, the requesting job is rolled out until the operator makes the pack or tape available.

SINGLE RESOURCE USE

A job that uses only one tape or disk pack concurrently does not need to specify resource demand with a RESOURC command. However, before assigning the same or a different type of resource, the current single resource (tape or disk pack) must be returned with either the RETURN or UNLOAD command. To allow more flexible resource handling, both the RETURN and UNLOAD functions decrement the default resource demand count from one to zero for jobs requiring only one tape or disk pack concurrently. For those jobs requiring more than one tape or disk pack concurrently (as specified by the RESOURC command), UNLOAD does not decrement the resource demand count; RETURN decrements the resource demand count only when all concurrent resource demands have been satisfied.

TAPE UNITS

Density resource identifiers (HD, PE, GE) should be used to indicate nine-track magnetic tape unit demand. The system supports nine-track drives with alternate densities and needs this information to prevent deadlocks and overcommitments. The 679-2/3/4 tape units are capable of processing both 800-cpi and 1600-cpi nine-track tapes; the 679-5/6/7 tape units handle both 1600-cpi and 6250-cpi nine-track tapes. An 800-cpi nine-track tape cannot be processed on a 1600/6250-cpi unit, and 6250-cpi nine-track tape cannot be processed on an 800/1600-cpi unit. The NT resource identifier, retained for compatibility, can be used only to allocate 800/1600-cpi nine-track units†† and cannot be specified concurrently in the same job with HD, PE, and GE resource demands. Default nine-track resource allocation is by density.

Examples:

Assume that an installation has the following tape drive resources:

- Two 679-4 nine-track tape drives (800/1600-cpi densities).
- Two 679-7 nine-track tape drives (1600/6250-cpi densities).

† Refer to Permanent File Commands in section 9 for a description of disk pack requests and to Tape Management Commands in section 12 for a description of tape requests.

†† NT resource demand cannot exceed the number of 800/1600-cpi nine-track drives at the installation. However, at tape assignment time, a 1600-cpi tape mounted on a 1600/6250-cpi unit is accepted for NT resource demand if it does not cause overcommitment (potential deadlock).

1. If a job makes a tape unit resource request with

RESOURC,NT=3.

the job is aborted with the message

INSUFFICIENT RESOURCES ON SYSTEM.

because only two units (the 679-4s) meet the NT specification.

2. If a job makes a tape unit resource request with

RESOURC,NT=1,PE=1.

the job is aborted with the message

CONFLICTING RESOURCE TYPES.

because the NT specification cannot be used with a density specification (HD/PE/GE).

3. If a job contains the following commands

LABEL,TAPE,NT,D=PE,VSN=TAPE1.

RESOURC,NT=2.

the job is aborted with the message

CONFLICTING RESOURCE TYPES.

because the LABEL command requested a tape unit by density (the default); therefore, later commands cannot schedule tape units using the NT specification.

Density identifiers are provided for seven-track tape units even though these units do not have alternate densities. This is done for consistency of format. The LO, HI, HY, and MT resource identifiers are all equivalent, and the last specification of any one of these is the seven-track tape unit demand for the job. For example, the resource request RESOURC(HI=1,HY=2) results in two seven-track tape resources being allocated for the job.

RESOURCE OVERCOMMITMENT

Under certain conditions, the system overcommits resources, provided all jobs with currently assigned resources can complete. For example, an installation with three tape units is processing jobs A and B. Included in each job is a RESOURC command scheduling two units. Job A requests its first tape. It is assigned the tape (unit 1) because there are enough units available for job A to complete. Job B requests its first tape. It is assigned the tape (unit 2) because either A or B can complete if assigned the last unit, and when the job that is assigned the last unit completes, the other can then use that unit and also complete. Job B then requests and is assigned its second tape (unit 3). It completes its operations (that is, terminates or returns the files on the tape) and makes the unit available for job A to complete.

NOTE

In a multimainframe environment, only the configuration of the machine on which the job is processed is considered in the overcommitment algorithm.

ALTERING RESOURCE REQUIREMENTS

The system manages its resources by keeping totals of the number of units of each device type scheduled and assigned to jobs. The number of units scheduled and the number of units assigned to a job can vary during job processing.

To change the number of units of a device type scheduled for this job, you can issue another RESOURC command. When decreasing the number of units scheduled for the job via a RESOURC command, the total resulting scheduled units must not be less than the number of units currently assigned to the job. If the resulting total would be less than the number currently assigned, the system aborts the job with an error message.

If the job has tape and/or removable pack units assigned to it when it attempts to increase its resource demands, the system determines if the request would cause a deadlock. If it would, it aborts the job with an error message.

NOTE

Always return all units assigned to your job before issuing another RESOURC command to increase resource demands. This action prevents a possible job abort that might lead to a deadlock condition.

The scheduled units can also be decreased by a RETURN command if the job, at the time of the return, is using its maximum scheduled units (refer to the description of the RETURN command in section 9).

Example:

The first RESOURC command schedules two 800-cpi, nine-track tape units. The two LABEL commands assign the tape units to the job. Because the maximum scheduled units were used concurrently, the RETURN command decreases the scheduled tape units to zero. The second RESOURC command schedules two 844-21 disk units and one 800-cpi, nine-track tape unit.

```
SAMSJOB,CM50000,T40.  
USER,SJGREEN,WGT.  
CHARGE,*.  
RESOURC,HD=2.  
LABEL,X,D=HD,VSN=TAPE1.  
LABEL,Y,D=HD,VSN=TAPE2.  
RETURN,X,Y.  
RESOURC,DI1=2,HD=1.  
.  
.  
.  
-EOI-
```


UNIT ASSIGNMENT

The method of assigning units depends on the resource type. For example, all tapes and all private disk packs not accessible by alternate users can only be assigned to one job at a time. All public packs and those private packs accessible by alternate users are shareable, and therefore, can be assigned to several jobs at the same time.

On indirect access file requests, the pack is charged to the job in fulfilling its resource demand only if the request causes the pack to be mounted. For direct access file requests, the pack is charged to the job when the first ATTACH of a direct access file is made.

A unit is assigned to a job until the job terminates or all direct access files residing on the unit that are assigned to the job are returned. At this point, a tape or a nonshareable pack can be dismounted. A shareable pack, however, can be dismounted only when there are no files residing on the unit that are assigned to any of the jobs sharing the pack.

NOTE

In GET requests for indirect access files, a pack is assigned to a job only as long as the pack is actually being used (that is, until the system retrieves the local copy of the file). Therefore, during a series of GET requests, the operator may determine that the pack is not being used and dismount it. If you have a direct access file on the pack, you can avoid this situation by attaching the direct access file before issuing the GET requests.

A single job cannot have more than 36 removable pack devices attached to the job concurrently.

RFL COMMAND

The RFL command sets the initial running field length for each subsequent job step when neither the routine for processing that step nor a loader table specifies a field length (refer to Field Length Control in section 3).

Format:

RFL,CM=nnnnnn,EC=mmmm.

or

RFL,nnnnnn,mmmm.

<u>Parameter</u>	<u>Description</u>
CM=nnnnnn	Specifies the central memory field length (octal is assumed unless decimal is specified by a D suffix or use of the digits 8 or 9). The value is rounded up to the nearest multiple of 100 ₈ . Specifying nnnnnn as 0 removes the effect of the previous RFL command and returns the setting of the field length to system control.
EC=mmmm	Specifies the extended memory field length in octal. The value of mmmm is the actual extended memory divided by 1000 ₈ .

The parameters may be specified positionally, by keyword, or intermixed positionally and by keyword. If intermixed, the positional parameters are evaluated according to their position among all the parameters.

The values of nnnnnn or mmmmm cannot exceed the values specified on the last MFL command or the maximum allowed for the job.

Prior to the appearance of the RFL command (or SETRFL macro), the system determines the field length for each job step, provided no field length is specified by a system routine or loader table (refer to Field Length Control in section 3).

ROLLOUT COMMAND

The ROLLOUT command suspends job execution and places the job in the rollout state. This releases the control point, central memory, and extended memory assigned to the job. You can specify a time period that must elapse before the job is returned. Otherwise, the job scheduler usually returns the job to execution when its priority is the highest of the jobs rolled out (refer to Rollout Control in section 3).

Format:

ROLLOUT,t.

t Optional time delay measured in job scheduler delay intervals. Values for t range from 0 to 262080 (777700g) intervals. Although the default base is decimal, octal values can be specified by a B suffix.

RTIME COMMAND

The RTIME command requests that the time be read from the real-time clock and issued to the dayfile (in seconds). This is the accumulated time since the last system deadstart.

Format:

RTIME.

The dayfile message format is:

RTIME nnnnnn.nnn SECS.

SETASL COMMAND

The SETASL command sets the system resource unit (SRU) limit for an account block. An account block is the job step sequence whose execution is charged to an account (refer to SRU Limit Control in section 3). The account is specified by the charge and project numbers on a CHARGE command, or if no CHARGE command is required, by the user name on the USER command. Each user name and each account has an SRU validation limit (refer to the LIMITS and ENQUIRE commands). Except for interactive jobs, the default account block SRU limit is the smaller of the user name and the account validation limits. For interactive jobs, the default limit is 64 SRUs.

Format:

SETASL,s.

s Maximum number of SRUs allowed for account block execution. Although the default base is decimal, octal values can be specified by a B suffix on the value.

s must be greater than or equal to the current job step SRU limit and less than or equal to the SRU limit†associated with your user name. Exceptions to this rule are the asterisk (*) and a value of 32760 (77770B) or greater. These exceptions set the account block SRU limit to the validation limit.

If the account block SRU limit is reached during account block execution, the system issues an error message and terminates the job (refer to Exit Processing in section 5).

SETCORE COMMAND

The SETCORE command presets each word within the field length.

Format:

SETCORE,p.

or

SETCORE,-p.

p Any of the following: (If a minus sign precedes the parameter p, the complement of p is set in core.)

<u>p</u>	<u>Fill Characters</u>
0	0
ZERO	Zeros (0)
INDEF	Indefinite (1777 0000 0000 0000 0000)
INF	Infinite (3777 0000 0000 0000 0000)

Each word within the field length (except for the first three words) is set to p. If p is omitted, the system assumes p=0.

To preset memory within a load sequence, use a LDSET,PRESET command as described in the CYBER Loader Reference Manual.

†The job step SRU limit must be lowered in the job before the account block SRU limit is lowered. Refer to the SETJSL command in this section.

SETJOB COMMAND

The SETJOB command changes some of the current job's attributes. You can change the user job name (UJN), can redirect the disposition of output if the job is not interactive or becomes a detached interactive job, and can change the job processing option for job termination or job suspension.

The SETJOB command does not take effect until the job terminates or is suspended. If you specify more than one SETJOB command, the last one takes precedence.

Format:

SETJOB,UJN=ujn,DC=dc,OP=op.

or

SETJOB,ujn,dc,op.

<u>Parameter</u>	<u>Description</u>								
UJN=ujn	Changes the job's default UJN to ujn. A UJN must be seven or less alphanumeric characters. For interactive jobs, the default is the user index hash (refer to the ENQUIRE command). Otherwise, the default is the UJN specified on the Job command.								
DC=dc	Specifies the disposition for output upon job termination. The default for all jobs is DF. You cannot change the output disposition for an interactive job unless the job terminates as a detached interactive job.								
	<table><thead><tr><th><u>dc</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>TO</td><td>Queues output with a wait disposition.</td></tr><tr><td>NO</td><td>Discards output.</td></tr><tr><td>DF</td><td>Specifies default output processing. The output disposition depends on the job's origin type (refer to File Types in section 2). No output is queued for interactive jobs.</td></tr></tbody></table>	<u>dc</u>	<u>Description</u>	TO	Queues output with a wait disposition.	NO	Discards output.	DF	Specifies default output processing. The output disposition depends on the job's origin type (refer to File Types in section 2). No output is queued for interactive jobs.
<u>dc</u>	<u>Description</u>								
TO	Queues output with a wait disposition.								
NO	Discards output.								
DF	Specifies default output processing. The output disposition depends on the job's origin type (refer to File Types in section 2). No output is queued for interactive jobs.								
OP=op	Specifies the job processing option the system exercises upon detached interactive job when it is no longer executing. Not applicable to noninteractive jobs. The default is SU.								
	<table><thead><tr><th><u>op</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>SU</td><td>The job remains in a suspended state until recovered or timed out (refer to the RECOVER command).</td></tr><tr><td>TJ</td><td>The system terminates the job.</td></tr></tbody></table>	<u>op</u>	<u>Description</u>	SU	The job remains in a suspended state until recovered or timed out (refer to the RECOVER command).	TJ	The system terminates the job.		
<u>op</u>	<u>Description</u>								
SU	The job remains in a suspended state until recovered or timed out (refer to the RECOVER command).								
TJ	The system terminates the job.								

Job attributes associated with omitted parameters remain unchanged.

SETJSL COMMAND

The SETJSL command sets the system resource unit (SRU) limit for each subsequent job step (refer to SRU Limit Control in section 3). Except for interactive jobs, the default job step SRU limit is the smaller of the user name and the account validation limits (refer to the LIMITS and ENQUIRE commands). For interactive jobs, the default job step limit is 64 SRUs. From an interactive job, you can increment your job step SRU limit to complete job step execution (refer to section 8).

Format:

SETJSL,s.

s Maximum number of SRUs allowed for job step execution. Although the default base is decimal, octal values can be specified by a B suffix on the value.

s must be greater than 0 and less than or equal to the current account block SRU limit, and the SRU limit associated with your user name. Exceptions to this rule are the asterisk (*) and values greater than 32760 (77770B). These values set the job step SRU limit at the validation limit if the account block SRU limit is set at the validation limit.

The system issues an error message when the job step SRU limit is reached. A job step within a batch job is then terminated (refer to Exit Processing in section 5). In interactive jobs, you can increment the SRU limit after receiving the SRU limit message or terminate the job. 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. You can enter one of the following:

<u>Entry</u>	<u>Description</u>
S	Increases the SRU limit by 320 units. Job execution continues.
S,nnnnn	Increases the SRU limit by nnnnn decimal units. Job execution continues. You can enter octal units by specifying a B after the octal number.
S,*	Increases the SRU limit to your maximum. Job execution continues.
Ⓒ	Terminates the job step. Subsequent commands, 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 interactive 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 interactive session.

The account block SRU limit must be raised before the job step SRU limit can be raised. Refer to the SETASL command in this section.

SETPR COMMAND

The SETPR command decreases the CPU priority of a job.

SETPR(p)

p Specifies the priority, $1 \leq p \leq 708$; if p exceeds 708 or the maximum priority defined for the origin type of the job, it is reduced to that value.

Upon job initiation, a job is assigned the maximum priority allowed for its origin type. (The installation defines these priority values.) If a job's CPU priority is lower than that of other jobs, the job is assigned control of the CPU only when jobs of a higher priority do not need it.

SETTL COMMAND

The SETTL command sets the CPU time limit for each subsequent job step. Each user name is validated for a maximum job step time limit (refer to the LIMITS and ENQUIRE commands). For batch jobs, when you do not specify a time limit on the Job command, the system sets the limit at your maximum validation. For interactive jobs, the default time limit is 64 CPU seconds. Interactive jobs can increment their job step time limit to complete job step execution (refer to Section 8).

Format:

SETTL,t.

t Maximum number of CPU seconds allowed for job step execution. Although the default base is decimal, octal values can be specified by a B suffix on the value.

t must be greater than 0 and less than or equal to your validated time limit. Exceptions to this rule are the asterisk (*) and values greater than 32760 (77770B). These values set the job step time limit at your validated time limit.

To set a time limit for job step execution on one CPU of a dual-processor machine, you must include a USECPU command in the job. Otherwise, the time limit is set for the cumulative job step execution time on both CPUs.

The system issues an error message when the job step time limit is reached. A job step within a batch job is then terminated (refer to Exit Processing in section 5). In interactive jobs, you can increment the time limit after receiving the time limit message or terminate job.

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

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

is entered and you can enter one of the following:

<u>Entry</u>	<u>Description</u>
T	Increases the central processor time limit by 64 CPU seconds. Job execution continues.

<u>Entry</u>	<u>Description</u>
T,nnnn	Increases the central processor time limit by nnnn decimal seconds. Job execution continues. You can enter octal seconds by specifying a B after the number.
T,*	Increases the central processor time limit to your maximum. Job execution continues. Causes the job to go through normal abort procedures (for example, EXIT processing which can be useful when using procedure files).
Ⓞ	Terminates the job step. Subsequent commands, 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.

In the following example, a program exceeds its allocated time limit. The user enters T to increase the time limit and the program then runs to completion.

```

List

00100 PROGRAM T
00110 DO 6 I=1,2500
00120 DO 6 J=1,4000
00130 A=1
00140 6 CONTINUE
00150 END

READY.
settl,10

READY.
run

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

SRU      66.835 UNTS.

RUN COMPLETE.

```

STIME COMMAND

The STIME command requests that the accumulated SRU value for the job be issued to the job's dayfile.

Format:

```
STIME.
```

Dayfile message format:

```
STIME nnnnnn.nnn UNTS.
```

SUBMIT COMMAND

The SUBMIT command places a user-supplied job file into the input queue as a separate job. SUBMIT can reformat the file according to directives within the file. Refer to Job Names in section 3 for a description of the names associated with the submitted job.

Format:

SUBMIT, lfn, q, NR, c

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the file to be submitted to the system for processing as a batch job. The default is the job's primary file.
q	Specifies disposition of job output (OUTPUT, PUNCH, PUNCHB, and P8) as follows: BC or B Use the default processing for local batch jobs to dispose of job output at the central site. NO or N Discard job output at job termination unless the files have been explicitly routed (default value for interactive origin jobs). RB=un or E=un Route job output to the remote batch or interactive user specified by the user name un. The default for un is the user name associated with the submitting job. TO Job output is queued with a wait disposition.
NR	No rewind option; the submit file and the file specified on a cREAD reformatting directive are not rewound before or after processing. If NR is omitted, the files are rewound before, but not after processing.
c	Escape character used to identify reformatting directives in the file to be submitted (lfn). If omitted, the system assumes c is /.

The number of deferred batch (LDI, SUBMIT, and ROUTE) jobs that you can have in the system concurrently depends on your validation (refer to the LIMITS command in this section). If this limit is exceeded, an error message is issued to the dayfile, and the SUBMIT command is not processed.

For SUBMIT to process reformatting directives, the first line of the submit file must be a cJOB directive. Each line preceded by an escape character is recognized as a reformatting directive. The escape character is specified on the SUBMIT command (/ by default). Throughout this description, the letter c, preceding a directive, denotes the escape character. Reformatting directives may 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 and requires you to observe special rules when inserting subsequent directives into the file (refer to description of cTRANS and cNOTRANS directives).

The system does not process reformatting directives unless the first line of the submit file contains the cJOB directive. In addition, the first two commands following the cJOB directive (second and third commands of the submit file) must be a Job command and a USER command (or a cUSER directive), respectively. Thus, the first three lines of a submit file to be reformatted should be:

```
ln1          cJOB
ln2          ujn,parameters.
ln3          USER,username,password.
```

or

```
ln1          cJOB
ln2          ujn,parameters.
ln3          cUSER
```

ln₁, ln₂, and ln₃ are optional line numbers.

You can include line numbers on the submit file and specify which line numbers are to be removed during reformatting with the SEQ and NOSEQ directives. This is especially useful if the submit file contains a BASIC program where line numbers are a requirement of the language. If line numbers are included in a submit file, the file must begin with a cJOB directive.

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. If line numbers are included in a submit file, the file must begin with a cJOB directive.
cUSER	Inserts a USER command identical to that of the submitting job during reformatting.
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 the following lines are preceded by line numbers and requests that they be removed (default value).

<u>Directive</u>	<u>Description</u>
cNOSEQ	Reverses the effect of the cSEQ directive. No attempt is made to remove leading line numbers from subsequent lines. This is especially useful when line numbers are required (such as in a BASIC program).
cPACK	Requests that all succeeding end-of-record and end-of-file marks be removed (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 end-of-record and end-of-file marks that currently exist.
cTRANS	Requests transparent submit mode. In transparent submit mode, SUBMIT ignores reformatting directives until an EOR or EOF mark is encountered. The EOR or EOF mark cannot be a mark to be created by a cEOR or cEOF directive. SUBMIT performs the following procedure for transparent submit mode processing.

1. Read cTRANS directive.
2. Check if the next line is a reformatting directive. If it is not, skip steps 3 and 4.
3. Process reformatting directive. If it is a cNOTRANS directive, end transparent submit mode processing.
4. Return to step 2.
5. Select transparent submit mode and read lines until an internal EOR or EOF mark is encountered.
6. If the cPACK directive is in effect, remove the EOR or EOF mark.
7. Return to step 2.

The cTRANS directive is typically used in conjunction with the cREAD directive. The cREAD directive copies 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 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. If not, transparent submit mode is selected before the cREAD directive is encountered and the cREAD is ignored.

The cSEQ or cNOSEQ directive in effect before transparent submit mode was 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 was selected remains in effect after it is selected.

<u>Directive</u>	<u>Description</u>
cNOTRANS	<p>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 and remains in effect until a cTRANS directive is encountered in the submit file.</p> <p>Be careful in placing 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 or an internal EOR or EOF mark.</p>
cREAD, lfn	<p>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 local to the job, the system automatically attempts a GET and then an ATTACH on the file. If lfn is not specified in the directive, TAPE1 is assumed. If the file specified cannot be found or the read file is found to be busy (direct access files only), 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 on the SUBMIT command. In this case, the rewind directive must precede the cREAD directive in the submit file if it is desired to rewind file lfn before the read operation begins. The system returns all files specified in cREAD directives before completion of the job.</p> <p>If the cPACK directive is in effect when the file lfn is 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.</p> <p>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 directive remains in effect for the file being read. Moreover, only those directives that immediately follow an internal EOR in the file being read are processed.</p> <p>If the file to be read is a binary file, it is recommended that the cTRANS directive be used to ensure that binary data is not mistaken for a reformatting directive. The cTRANS directive should immediately precede the cREAD directive in the submit file, if used.</p>
cREWIND, lfn	<p>Requests that the system rewind file lfn to the beginning-of-information (BOI). If lfn is not supplied, TAPE1 is assumed. This directive is required only if the NR parameter is included in the SUBMIT command. Otherwise, file lfn is automatically rewound.</p>

Directive

Description

This directive is used in conjunction with the cREAD directive. Thus, if it is desired to rewind a file before the read operation begins, this directive must precede the cREAD directive in the submit file.

c₁EC=c₂

Indicates that the escape code character is to be changed from c₁ (current escape code) to c₂ (new escape code). The new escape code is used to recognize all subsequent reformatting directives until further change.

Input lines must not exceed 150 6-bit characters. SUBMIT processes the first 80 characters as the command. The remaining 70 characters are discarded and may contain a sequence number or comments. If a line exceeds 150 characters, the results are unpredictable.

If the submitted job contains an invalid USER command, the job entering the SUBMIT command is aborted with no exit processing. The security count for the user name that entered the SUBMIT command is decremented. Terminal users are immediately logged off. For interactive jobs, the system then begins the login sequence if the security count is greater than zero. For further information concerning use of the SUBMIT command from an interactive terminal, refer to Volume 2, Guide to System Usage.

Consult your job's dayfile to determine the cause of any errors that occurred during job processing. The dayfile for the submitted job is disposed according to the disposition parameter on the SUBMIT command.

When you submit a batch job from an interactive job, all output is dropped (unless requested otherwise by the disposition parameter). This includes the dayfile output. Therefore, make provisions within your job to save the contents of the dayfile if a processing error occurs. You can do this by including the following commands at the end of the command record.

EXIT.
DAYFILE, lfn.
REPLACE, lfn.

SUMMARY COMMAND

The SUMMARY command gives you information about your jobs in the system. In general, it performs the same function as the ENQUIRE command, which is described earlier in this section.

Format:

SUMMARY, OP=p₁p₂...p_n, FN=lf_{n1}, O=lf_{n2}.

or

SUMMARY, p₁p₂...p_n.

or

SUMMARY.

You cannot specify the JSN=jsn or UJN=ujn parameters on the SUMMARY command. Otherwise, the parameters of this command function identically to those of ENQUIRE, except that the third format of SUMMARY (no parameters specified) defaults to SUMMARY,OP=R.

NOS will drop support of the SUMMARY command at a future release. The system issues the following informative message after it processes the command:

DROPPING SUPPORT OF *SUMMARY*, USE *ENQUIRE*.

SWITCH COMMAND

The SWITCH command sets the sense switches for reference by one of your programs.

Format:

SWITCH,jsn,switch₁,switch₂,...,switch_n.

<u>Parameter</u>	<u>Description</u>
jsn	Specifies the job sequence name (JSN) of the job whose sense switches will be changed. The default is the current job. The parameter jsn can appear anywhere in the list.
switch _i	Specifies the switch(es) to be set. You must denote each switch switch _i by an integer in the range of 1 through 6. If you specify 0 for this parameter, the system sets all sense switches.

UPROC COMMAND

The UPROC command specifies a user prologue, a procedure file the system executes each time you start a job.

Format:

UPROC,FN=pfile.

pfile	One of your permanent files that contains the desired prologue. If pfile contains more than one procedure, the system executes only the first one.
-------	--

You must specify the parameter in the keyword=value form. If you omit the FN=pfile parameter, the system will no longer execute a user prologue.

The system executes the system and user prologues only after it processes your user validation and charge information but before it initiates job recovery processing. If your site does not require a CHARGE command, you still must include one in some form in batch jobs if you want your user prologue executed. Depending on the length and nature of your user prologue, you may want to initiate recovery processing before your user prologue ends. To do this, place the RECOVER command within your user prologue (refer to section 8 for the RECOVER command and section 4 for the writing of procedures).

USECPU COMMAND

The USECPU command specifies which central processor is to be used when more than one is available for processing.

Format:

USECPU,n.

n Specifies which central processor to use.

<u>n</u>	<u>Description</u>
0	Use either central processor.
1	Use CPU 0.
2	Use CPU 1.

The USECPU command may be used only when the system is running on a model 73-2x, 74-2x, 174, 720-2, 730-2, 6500 or 6700. On a model 74-2x or 6700, CPU 0 is the parallel processor, and CPU 1 is the serial processor. On the other systems, both CPUs are serial processors. This command is ignored on single CPU machines.

USER COMMAND

The system uses the parameters on the USER command to determine if you are a legal user, which resources you are validated to use, and the extent (limits) to which you may use those resources.

No commands are allowed between the Job and USER commands in noninteractive jobs. If this is attempted, the first command is interpreted as an invalid USER command, and the submitting job is aborted with appropriate messages to the dayfile. The submitted job is dropped.

Format:

USER,username,password,familyname.

<u>Parameter</u>	<u>Description</u>
username	A one- to seven-character alphanumeric user name.
password	Alphanumeric password. Its maximum length is seven characters; its minimum length is defined by the installation.
familyname	Optional parameter identifying the family [†] of permanent file devices on which your permanent files reside. Specify a family name when the system can access more than one permanent file device family.

[†]Refer to section 2 for a description of permanent file device families.

This command also defines controls and validation limits for the job and defines your permanent file base. An installation may operate with secondary USER commands either enabled or disabled. If enabled, you may specify a different permanent file catalog or family during job processing by issuing another USER command. The system only associates permanent file validation limits with this new user name. The remaining validation limits still reflect those of the user name specified at job initiation (refer to the LIMITS command earlier in this section). If secondary USER commands are disabled (default mode) and a secondary USER command is issued, the job is aborted (no exit processing). The security count for the current user name is decremented accordingly.

The job is aborted and the security count is decremented if a USER command containing an invalid user name is detected at any time, regardless of whether secondary USER commands are enabled or disabled. At a terminal, you are immediately logged off with no dayfile message issued to the terminal. If your security count is exhausted, the user name is denied all access to the system until the security count has been reset by the installation personnel.

The password is deleted from the USER command before this command is issued to the dayfile.

Normally, the familyname parameter need not be included on the USER command. However, if you make a practice of specifying your family name each time you submit a job, you can be sure that your job will be processed even if your normal system is not available and your permanent file family is moved to a backup system. If, after the first USER command, you do not specify a familyname on the USER command, your permanent file family remains the same. If you specify the 0 (zero) familyname, your permanent file family becomes the system default family.

Example:

An installation has two systems, A and B. System B provides backup service for system A. The system default family name for system A is AFAM, and the system default family name for system B is BFAM.

During normal operations, system A user CWJONES with password JPWD could enter either of the following USER commands.

```
USER,CWJONES,JPWD.
```

```
USER,CWJONES,JPWD,AFAM.
```

System B user JDSMITH with password SPWD could enter either of the following commands.

```
USER,JDSMITH,SPWD.
```

```
USER,JDSMITH,SPWD,BFAM.
```

If system A failed, user CWJONES would be required to enter

```
USER,CWJONES,JPWD,AFAM.
```

to identify his family of permanent file devices. User JDSMITH could enter either of the USER commands as before because the default family name would still be valid.

There are many system commands that you can use only in interactive jobs. This section describes these commands under the following headings:

- Terminal Control Commands

- ASCII
- AUTO
- BRIEF
- CSET
- NORMAL
- TIMEOUT
- TRMDEF

- Subsystem Selection Commands

- ACCESS
- BASIC
- BATCH
- EXECUTE
- FORTRAN
- FTNTS
- NULL

- Interactive Status Commands

- cD
- cE
- cS

- Job Processing Commands

- BYE
- DIAL
- GOODBYE
- HELLO
- HELP
- LIB
- LIST
- LOGIN
- LOGOUT
- NOSORT
- RECOVER
- RUN
- TEXT
- WHATJSN
- X

- Primary File Editing Commands

- ALTER
- DELETE
- DUP
- LIST
- MOVE
- READ
- RESEQ
- WRITE
- WRITEN

TERMINAL CONTROL COMMANDS

The terminal control commands change the characteristics of the terminal and vary the source and format of information given to and received from the system. You can enter these commands at any time after you successfully log in.

NOTE

Except where specified, these commands are not valid from procedures.

ASCII COMMAND

If you are at an ASCII terminal, this command allows you to use the ASCII 128-character set and to place the terminal in ASCII mode. On non-ASCII terminals, this command allows the use of up to 128 of the characters defined for the terminal. The ASCII command allows you to enter lowercase letters that are interpreted by the system rather than translated to uppercase as is done in normal mode.

Format:

ASCII

In ASCII mode, characters entered from the terminal are translated into 6/12 display code. On an ASCII terminal, 6/12 display code consists of 95 graphic characters and 33 control characters (the 128-character set). The standard ASCII graphic 64-character set contains only the first 64 (or 63 depending on an installation option) of these 95 characters. In normal mode, the user is restricted to these 64 (or 63) characters. Characters of the ASCII graphic 64-character set are processed internally as 6-bit display code characters. The additional characters that make up the ASCII 128-character set are processed internally as 12-bit display code characters. Refer to appendix A for further information on display codes and character sets.

IAF receives all ASCII characters except line feed, carriage return, NUL, EOT, DEL, and the logical backspace character. Control characters (line feed, cancel line, and backspace) are not received unless in special editing mode. Commands can be entered in either uppercase or lowercase regardless of ASCII mode.

NOTE

If a correspondence code terminal is being used, you must be in ASCII mode in order to use the full correspondence code set.

AUTO COMMAND

The AUTO command automatically generates five-digit line numbers.

Format:

AUTO,nnnn,iiii

nnnn	Specifies the beginning line number; default value is 00100.
iiii	Specifies the increment value added for each succeeding line number; default value is 10.

To exit from auto mode, you press the cancel line character followed by $\text{\textcircled{CR}}$, thus deleting the current line. You can also exit from auto mode by entering the termination sequence. You can then enter a new command on the next line.

You can alter the line numbering sequence by deleting the line with the cancel line character and then entering a new beginning line number. You 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 deletes a line number to change the numbering sequence and then deletes another line number in order to place a character in column 6.

```
READY.
auto
00100 program header
00110 write 1000
00120                                     User enters cancel line character.
*DEL*
00230 1000 format(15x,'headings')       User enters entire line.
00240 write 2000
00250 2000 format(15x,'author',9x,'title',15x,
00260                                     User enters cancel line character.
*DEL*
00260+      'publisher',6x,'cost')       User completes the line.
00270 stop
00280 end
00290                                     User enters cancel line character.
*DEL*
list

00100 PROGRAM HEADER
00110 WRITE 1000
00230 1000 FORMAT(15X,'HEADINGS')
00240 WRITE 2000
00250 2000 FORMAT(15X,'AUTHOR',9X,'TITLE',15X,
00260+      'PUBLISHER',6X,'COST')
00270 STOP
00280 END
```

READY.

NOTE

Do not attempt to enter input until the line number prompt appears.

BRIEF COMMAND

The BRIEF command suppresses all full and partial headers produced by certain commands (for example, LIBEDIT). It also prevents the system from echoing editing changes you make to the primary file when using the primary file editing commands described later in this section.

Format:

BRIEF

To clear the effect of the BRIEF command, use the NORMAL command.

CSET COMMAND

The CSET command sets the character set mode of the terminal.

Format:

CSET,c

c	Specifies terminal character set mode.
ASCII	Set ASCII mode (ASCII 128-character set).
NORMAL	Set normal mode (ASCII graphic 64-character set or 63-character set).

The CSET command may also appear in a procedure file.

NOTE

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

NORMAL COMMAND

The NORMAL command reverses the effect of the ASCII, AUTO, BRIEF, CSET, and NOSORT commands on both input and output. The system initially assumes that this command has been entered. Normal mode uses the ASCII graphic 64-character set or 63-character set. All lowercase letters are converted to uppercase (refer to ASCII command) and all command headers are printed (refer to BRIEF command).

Format:

NORMAL

TIMEOUT COMMAND

The TIMEOUT command can change your status from the no-timeout status to the standard timeout status. In standard timeout status, you are automatically logged out after 10 minutes of inactivity. If you have a no-timeout status (refer to the LIMITS command), the terminal remains connected until you log out. The TIMEOUT command changes your status for the session in progress.

Format:

TIMEOUT

TRMDEF COMMAND

The TRMDEF command changes one or more of the characteristics of your terminal (for example, the termination character or page width). You can also use the terminal definition commands for this purpose. Appendix K describes these commands.

Format:

TRMDEF,L=lfm,tc₁=v₁,...,tc_n=v_n

L=lfm Specifies an optional file on which the terminal redefinition information is written. If lfm is omitted, file OUTPUT is assumed. If another file is specified, the changes are implemented when the file is listed or copied to the terminal.

tc_i=v_i Specifies the terminal characteristic to be changed. tc_i is specified by a two-character mnemonic listed in table K-1 in appendix K. The NUL and DEL characters can be specified on the TRMDEF command although they cannot be used with the terminal definition commands (refer to table K-1). Almost all of the parameters available for terminal definition commands can be used in the TRMDEF command. (Exceptions are DL and some IN parameters as noted in appendix K.) The new option, v_i, is any value in the range given for that terminal characteristic in table K-1. It can be a decimal value, a coded value with a special meaning (such as PR for printer), or a single character. If the new option is a single character, it can be given in any of the following formats.

<u>v_i</u>	<u>Description</u>
v	Any alphanumeric character (a character with a display code in the range from 0 through 44 ₈).
\$v\$	Any character, including special characters, delimited by dollar signs (for example, \$*\$). If the character is a dollar sign (\$), it must be specified using two dollar signs (for example, \$\$\$\$).
vvvB	Octal value of an ASCII character (for example, 52B, which would be equivalent to an entry of \$*\$).
vvD	Decimal value of an ASCII character (for example, 42D, which would be equivalent to an entry of \$*\$).
Xvv	Hexadecimal value of an ASCII character (for example, X2A, which would be equivalent to an entry of \$*\$).

NOTE

If a numeric value for a single ASCII character is specified without a pre- or post-radix, the value is assumed to be octal unless it contains an 8 or 9. In this case, it is translated as decimal.

If any errors are found in the command, a message is issued, and the terminal characteristics in effect before entry of the command remain in effect.

Example:

TRMDEF,IN=PT	This command changes the input device to a paper tape reader, thus allowing you to read data and commands from a tape. It has the same effect as the corresponding terminal definition command (that is, entry of the control character followed by IN=PT).
--------------	---

The TRMDEF command can appear in a procedure.

SUBSYSTEM SELECTION COMMANDS

A subsystem creates an environment that facilitates the execution of certain classes of jobs. For instance, the FORTRAN subsystem provides a framework in which you can conveniently enter, compile, and execute FORTRAN 5 programs.

The access, execute, BASIC, FORTRAN, FTN5, and null subsystems greatly restrict your job capabilities. Under them, you cannot enter many of the commands described in this manual unless you use the X,ccc command format (described later in this section). To have the largest set of system commands at your disposal, use the batch subsystem. If you want to use the RUN command, however, you must select either the BASIC, execute, FORTRAN, or FTN5 subsystem.

The following commands select a specific subsystem. You 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 (refer to the LIMITS command).

A specific subsystem can be associated with an indirect access file so that whenever you specify that file as the primary file, the associated subsystem is selected automatically. A subsystem flag can be set when you save the file. If you include the SS=subsystem parameter on the SAVE command, you can specify any of the valid subsystems. If you enter SAVE without SS=subsystem and the file is the primary file, the subsystem flag for the current subsystem with the file is set. To save the primary file with no subsystem association, you can either enter the null subsystem before saving the file, or 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, it must be saved with the SS=subsystem parameter specified.

ACCESS COMMAND

The ACCESS command selects the access subsystem. You cannot enter a RUN command under the access subsystem. While in the access subsystem, you can communicate with another interactive terminal using the DIAL and WHATJSN commands. You must be authorized to use the access subsystem (refer to LIMITS command).

Format:

```
ACCESS
```

The DIAL and WHATJSN commands can be entered only when the access subsystem is active.

BASIC COMMAND

The BASIC command selects the BASIC subsystem. A RUN command under the BASIC subsystem uses the BASIC compiler.

Format:

```
BASIC,ccc
```

ccc Specifies a command to be executed after the system executes 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 a copy of permanent file PRIME the primary file. The file name (PRIME) is a valid parameter with the OLD command.

BATCH COMMAND

The BATCH command selects the batch subsystem. You cannot use the RUN command under the batch subsystem.

Format:

BATCH,f1

or

BATCH

f1 Specifies the initial running field length for subsequent job steps. Entering this value is equivalent to entering the RFL,f1 command (refer to section 7). If f1 is omitted, 0 is assumed. If you implicitly selected the batch subsystem by retrieving an indirect access permanent file with the OLD command, the field length is either 0 or that entered with the last RFL command.

Unless you have a reason to do otherwise, choose the batch subsystem for normal job processing. The batch subsystem allows you to enter essentially all commands allowed in batch jobs. Some file management commands and nearly all execution control processing commands are available in the other subsystems only through the X,ccc command.

Although you can enter any system command from an interactive job, the output produced by several of these commands is formatted for transmission to a line printer (137 characters per line). Through use of the L072 command, you can format the output for transmission to a terminal (72 to 160 characters per line) but some data may be lost. Refer to the description of the L072 command in section 9.

EXECUTE COMMAND

The EXECUTE command selects the execute subsystem. A RUN command under this subsystem executes a previously compiled (object code) program. The RUN command must be entered to initiate execution.

Format:

EXECUTE,ccc

or

EXECUTE

ccc Specifies one command to be executed after the system executes the EXECUTE command. Any valid command is permitted, as well as all valid parameters for that command.

EXECUTE,OLD,OBJFILE

In this example, you select the execute subsystem and make a copy of permanent file OBJFILE the primary file. The file name (OBJFILE) is a valid parameter with the OLD command.

NOTE

Source language programs cannot be executed under the execute subsystem.

Use the execute subsystem whenever possible to conserve system resources. Efficiency can be achieved when certain programs are used frequently. For example, a source code program created under the FORTRAN subsystem must first be compiled before it can be executed (this process is automatic; you need only enter the RUN command). The program automatically goes through a compilation phase, which produces an executable object code program, and the object code program is then executed. Thus, greater efficiency can be 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 (under the FORTRAN subsystem):

```
RUN,B=lfm
```

Causes the object code program to be placed on temporary file lfm.

```
SAVE,lfm/SS=EXECUTE
```

Retains file lfm as a permanent file and sets the execute subsystem flag.

If this is done, subsequent requests for the file using the OLD command causes the execute subsystem to be selected. Entering the RUN command initiates execution of the object code program.

FORTRAN COMMAND

The FORTRAN command selects the FORTRAN subsystem. A RUN command under this subsystem uses the FORTRAN Version 5 compiler.

Format:

```
FORTRAN,ccc
```

ccc	Specifies a command in addition to the FORTRAN 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.

```
FORTRAN,OLD,TAX
```

In this example, the user selects the FORTRAN subsystem and makes a copy of permanent file TAX the primary file. The file name (TAX) is a valid parameter with the NEW command.

FTNTS COMMAND

The FTNTS command selects the FTNTS subsystem. A RUN command under this subsystem uses the FORTRAN Extended Version 4 compiler.

Format:

FTNTS,ccc

ccc Specifies a command in addition to the FTNTS 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.

FTNTS,OLD,DATAGEN

In this example, the user selects the FTNTS subsystem and makes a copy of permanent file DATAGEN the primary file. The file name (DATAGEN) is a valid parameter with the OLD command.

NULL COMMAND

The NULL command selects the NULL subsystem. You cannot use the RUN command under this subsystem.

Format:

NULL

This command is entered before saving the primary file if you do not want a specific subsystem flag to be associated with the file. Normally, when you save 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 alternative way of saving the primary file with no subsystem association is to specify the SS=NULL parameter on the SAVE command. No automatic subsystem association occurs when saving temporary files other than the primary file.

If you do not specify a subsystem at login time, the system sets the subsystem specified in the validation file during login (refer to the LIMITS Command in section 7). Null is the default.

INTERACTIVE STATUS COMMANDS

The system executes interactive status commands immediately. You may enter them while the system is executing another command. The system queues other commands that are typed prior to prompts (refer to section 17).

Format:

cx

c c is the control character defined for the terminal (refer to appendix K).

x x can be the letter D, E, or S.

The system generates only a line feed if you enter an interactive status command during one of the following states:

- Text mode
- Auto mode
- Input mode (a program is waiting for input)
- During login
- User break processing

DETACH COMMAND (d)

The Detach command detaches a terminal job from the terminal. You can detach a job at any time during a terminal session. To detach the job during output, first enter the user break 1 and then enter the Detach command. Otherwise, cD alone is sufficient.

When you detach a job, the following message appears at the terminal:

```
JOB DETACHED, JSN=jsn
```

Subsequent to the detachment, the system gives the terminal session a new job sequence name (JSN). Any commands you entered before or after the detachment that the system has not yet processed become part of the new terminal job. However, if your job is executing a procedure when you detach it, the procedure remains a part of the detached job. The system initiates the new job with recovery dialogue processing (refer to the RECOVER command in this section).

The detached job executes until one of the following occurs:

- The system executes the last command in its command record. Detaching a job while it is in the IDLE status causes the system to reach the end of the command record.
- A fatal error aborts the job.
- The job requests input from the terminal or queues output to the terminal.
- The job exceeds its time limit or SRU limit.
- Certain system requests are encountered.

At this point, the system suspends the detached job (unless you had specified the OP=TJ parameter on the most recent SETJOB command). The job remains suspended until you recover it or until it exceeds its timeout period. You can recover an executing or suspended job with the RECOVER command, which is described later in this section.

The system limits the number of jobs you can detach (refer to the LIMITS command in section 7).

IMMEDIATE JOB STATUS COMMAND (cE)

This command requests detailed job status response. You can enter this command while another command is active, and the system processes it immediately. The following output is an example of the system's response to a cE command:

```
JSN: AADF  SYSTEM:  BATCH      SRU:      2.095  FILE NAME:  SAMPLE2
STATUS:  IDLE
```

ABBREVIATED JOB STATUS COMMAND (cS)

This command causes the system to give an abbreviated job status response. The system issues one of the following responses.

<u>Response</u>	<u>Description</u>
EXECUTE	System is currently processing your command.
IDLE	System is waiting for you to enter a command.
WAIT	System is waiting for system resources to become available.

JOB PROCESSING COMMANDS

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

BYE COMMAND

The BYE command ends your session with IAF and either logs you out of the network and disconnects the terminal or places you under the control of another application.

Format:

BYE,application

or

BYE

application

Optional parameter specifying an application to which you wish to be connected.

<u>application</u>	<u>Description</u>
MCS	Message Control System.
RBF	Remote Batch Facility.
TAF	Transaction Facility.

If the application is available and you are validated for that application, you are automatically released from IAF control and placed under the control of the named application. If this parameter is not specified, you are logged out of the system and the terminal is disconnected. In this case, IAF responds by printing:

```
UN=username LOG OFF      hh.mm.ss.  
JSN=jsn      SRU-S       s.sss UNTS.  
IAF      CONNECT TIME  ho.mi.se.
```

The network then prints:

```
LOGGED OUT.
```

The logout information printed is:

```
username      User name.  
hh.mm.ss.    Time of day this command was entered.  
jsn          Job sequence name.  
s.sss       Measure of system resources used during IAF  
            connection. The SRU is a unit of measurement  
            which includes all CPU time, memory usage, and  
            I/O activity since login to IAF.  
ho.mi.se.    Length of time you were connected to IAF.
```

NOTE

Ensure that all temporary files to be retained are made permanent before issuing this command. All files which have not been saved are released when the BYE command is processed.

DIAL COMMAND

The DIAL command sends a one-line message to another terminal user.

Format:

DIAL,jsn,sss

jsn Specifies the job sequence name (JSN) of the terminal to receive the message.

sss Specifies the one-line message.

The DIAL command can be entered only if the access subsystem is active. The terminal to receive the message can have any subsystem active; it need not be under the access subsystem or have access validation.

You can determine the appropriate jsn by using the WHATJSN command. If the terminal specified is currently receiving output or has an input request outstanding, the message is lost. No further attempt to transmit the message is made.

GOODBYE COMMAND

Same as BYE command.

Format:

GOODBYE,application

or

GOODBYE

HELLO COMMAND

The HELLO command logs you out of IAF and switches you to another application or reinitiates the login sequence.

Format:

HELLO,application

or

HELLO

If application is specified, this command is the same as the BYE,application command.

If application is not specified, this command logs you out of IAF, as in the BYE command, but reinitiates the login sequence. Any temporary files that are to be retained must be made permanent before the HELLO command is entered. This form of the command allows a new user to log in without first having to establish communications with the network by dialing the terminal telephone number.

HELP COMMAND

The HELP command assists you in using any command you can enter interactively. If you are 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:

Format:

HELP

FOR MORE HELP TYPE YES, OTHERWISE ENTER COMMAND DESIRED.

?

You either type YES in reply to the request or enter the command you want to have described. You should enter enough characters to uniquely identify the command. If the entry is not unique, the system describes the first command that contains the supplied characters. You must type END after a question mark to terminate the HELP command.

You can also use the HELP command in a procedure file.

LIB COMMAND

The LIB command retrieves a copy of an indirect access permanent file from the catalog of special user name LIBRARY and makes it the primary file.

Format:

LIB, lfn=pfm/PW=password, PN=packname, R=r, NA, ND, WB

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the local file name for permanent file pfn. The default is pfn.
pfm	Specifies the permanent file to be retrieved.
PW=password	Specifies the password needed to retrieve file pfn if one exists.
PN=packname	Specifies the auxiliary device on which pfn resides if pfn resides on one.
R=r	Specifies the device type of the auxiliary device on which pfn resides, only if different from the installation-defined default.
NA	Specifies the no-abort option. LIB processing errors do not terminate the job. The job waits for the mounting of disk packs.
ND	Specifies the no-drop option. Local files are not released.
WB	Specifies the wait-if-busy option. The job waits for the mounting of disk packs.

Section 10 contains a detailed description of parameters used on permanent file management commands.

Direct access files can also reside in the catalog of LIBRARY although another command method must be used to access these files (described in the following paragraphs). 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 you enter the LIB command, the system retrieves a copy of the specified indirect access permanent file and makes it become the new primary file. For example:

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

Unless you enter the ND parameter, the system releases all files that do not have the no-auto-drop status. Certain system scratch files (ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD, for example) have this status. You can give a local file this status with the SETFS command.

In this example, you retrieve a copy of permanent file ABC75 and it becomes the new primary file; all local files remain associated with the job. The previous primary file becomes a nonprimary temporary file. The LIB command accesses only indirect access permanent files.

You can enter the following commands to access permanent files in the catalog of LIBRARY (differences in command function are described).

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

This command must be entered to use any direct access permanent files in the catalog of LIBRARY. The system does not create a temporary file since all I/O operations directly involve the permanent file itself. However, if you wish to reference the file by a name other than its permanent file name, you can assign a local file name (lfn) in the command. The primary file name remains unchanged.

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

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

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

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

You can access from the catalog of LIBRARY only those files that allow alternate user access. The system considers all users, except user name LIBRARY, alternate users. A summary of the rules for alternate user access 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 name (LIBRARY).

The following forms of CATLIST generate the list of accessible files in the catalog of LIBRARY.

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

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

LIST COMMAND

The LIST command lists the contents of a local file. If you want to selectively list lines of the primary file, use the alternate form of the command described later in this section under Primary File Editing Commands.

Format:

LIST,F=lfm

or

LIST

lfm Specifies the local file to be listed. The default is the primary file.

LOGIN COMMAND

Same as HELLO command.

Format:

LOGIN,application

or

LOGIN

LOGOUT COMMAND

Same as BYE command.

Format:

LOGOUT,application

or

LOGOUT

NOSORT COMMAND

The NOSORT command prevents the system from automatically sorting the primary file by clearing an internal indicator called the sort flag. Normally, the system sorts the primary file, if needed, whenever the current command causes it to read the file.

Format:

NOSORT

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.

RECOVER COMMAND

The RECOVER command recovers detached jobs or interrupted terminal sessions. The RECOVER command can also appear in a procedure.

Format:

RECOVER,JSN=jsn,OP=T

or

RECOVER,jsn,T

or

RECOVER

The parameters are order-independent if you specify the parameters in the keyword=value form. Otherwise, the parameters are order-dependent.

<u>Parameter</u>	<u>Description</u>
JSN=jsn	Specifies the job sequence name (JSN) of the detached job.
OP=T	Directs the system to abort recovery processing if you have no recoverable jobs. Otherwise, the system engages you in a recovery dialogue.

If you specify no parameters and have recoverable jobs, the system displays those jobs and their status and prompts you for further recovery directives.

During job processing, recovery may be necessary in the following cases:

- Your terminal is disconnected from the system without being logged out.
- A system malfunction occurs which requires a restart.
- You want to continue a job you detached.

The length of time before the system times out the job depends on your validation limits (refer to the LIMITS command) and the site-determined default time-out period.

If you have recoverable jobs in the system, the system automatically displays them after login. The system lists four attributes of each recoverable job: its job sequence name (JSN), user job name (UJN), current job status, and the time remaining before the system forces termination of the job.

Example:

81/12/18. 14.14.38. terminalname
CDC NETWORK OPERATING SYSTEM
FAMILY: ,username,password
JSN: ABNA, NAMIAF

RECOVERABLE JOB(S)

JSN	UJN	STATUS	TIMEOUT
ABMK	AJLA	SCH ROLLED	
ABMQ	AJLA	SUSPENDED	14 MIN.

ENTER GO TO CONTINUE CURRENT JOB,
RELIST TO LIST RECOVERABLE JOBS,
OR DESIRED JSN: go
READY.

There are eight statuses that can appear in the STATUS column:

<u>Status</u>	<u>Description</u>
EXECUTING	The job is executing.
IDLE	The system has reached the end of the job's command record.
INITIAL	The job awaits its initial rollin.
I/O ROLLED	The system is processing interactive input or output from the job.
SCP ROLLIN	The job awaits a rollin for an SCP request.
SUSPENDED	The system is waiting for user input to complete the jobstep.
T/E ROLLED	The rolled out job awaits a time or an event.
SCH ROLLED	The scheduler has rolled out the job.

If you choose the GO option, the system ends the recovery dialogue. If you choose the RELIST option, the system relists the recoverable jobs. If you specify one of the JSNs, you get the following display:

```
JSN: jsn  SYSTEM:subsystem SRU: nnnnnn.nnn  FILE NAME: primaryfile
STATUS: message
CHARACTER SET: charset  MODES: modes
instruction
```

jsn	Job sequence name.
subsystem	Interactive subsystem.
nnnnnn.nnn	The SRUs used by the job.
primaryfile	The job's primary file. The system suppresses this field if there is no primary file.
message	Message from executing job, if present. Otherwise, the field is suppressed.
charset	The interactive character set mode (ASCII or NORMAL) of the job.
modes	The prompting mode (BRIEF or PROMPT ON).
instruction	The system's instruction on job continuation. There are five possibilities: TEXT MODE. ENTER GO TO CONTINUE. OUTPUT AVAILABLE. ENTER GO TO CONTINUE. INPUT REQUESTED. ENTER GO TO CONTINUE. IDLE. ENTER COMMAND. JOB IN SYSTEM. ENTER GO TO CONTINUE.

If you choose the GO option, the recovered job resumes. If you enter something other than the word GO, the system terminates the job. If you type GO and the job has output available, the job is started and output continues at the terminal. A job waiting for input will not start until you enter the input.

In general, recovery is designed to provide minimum inconvenience to you with maximum security. However, under certain circumstances, recovery of your information cannot be perfect. Here are three examples of such circumstances:

- In some cases, a few lines of output may be lost when a phone line is disconnected because data queued in the network cannot be recaptured by IAF.
- If a phone line is disconnected when you are entering lines, you can normally expect to lose the last few lines you entered.
- When you are disconnected, terminal characteristics for that terminal session are set to the default. If a system prologue or user prologue sets terminal characteristics, these characteristics are restored during job recovery. If the recovered job had altered any of the terminal characteristics, they will not be restored automatically. For example, you set the page width to 90 before being disconnected. Upon recovery, the page width is set to the default for the terminal class, not to 90.

The following is another example of job recovery.

READY.
list

```
00100 LET A=1
00110 INPUT B
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT I," FACTORIAL IS ";A
00150 NEXT I
00160 END
```

READY.
run

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

```
82/01/20. 17.51.54. terminalname
CDC NETWORK OPERATING SYSTEM NOS 2
FAMILY: ,username,password
JSN: ACPW, NAMIAF
```

RECOVERABLE JOB(S)

JSN	UJN	STATUS	TIMEOUT
ACPJ	AJLA	SUSPENDED	8 MIN.

ENTER GO TO CONTINUE CURRENT JOB,
RELIST TO LIST RECOVERABLE JOBS,
OR DESIRED JSN: acpj

```
JSN: ACPJ SYSTEM: BASIC SRU: 0.047 FILE NAME: RECOVER
STATUS:
CHARACTER SET: NORMAL MODES: PROMPT ON
INPUT REQUESTED. ENTER GO TO CONTINUE. go
```

? 10

1	FACTORIAL IS	1
2	FACTORIAL IS	2
3	FACTORIAL IS	6
4	FACTORIAL IS	24
5	FACTORIAL IS	120
6	FACTORIAL IS	720
7	FACTORIAL IS	5040
8	FACTORIAL IS	40320
9	FACTORIAL IS	362880
10	FACTORIAL IS	3.62880E+6

READY.

RUN COMMAND

The RUN command compiles and/or initiates execution of a local file. The compiler used (if any) depends on the current subsystem. This command is valid only in the BASIC, FORTRAN, FTNFS, and execute subsystems.

Formats:

RUN

RUN,I=lf_{n1}

RUN,B=lf_{n2} or RUN,C=lf_{n2}

RUN,I=lf_{n1},B=lf_{n2} or RUN,I=lf_{n1},C=lf_{n2}

RUN,T,q₁,q₂,...,q_n

<u>Parameter</u>	<u>Description</u>
lf _{n1}	Specifies the local file that contains the source program to be compiled and executed.
lf _{n2}	Specifies the compiler/assembler output file. File lf _{n2} can be later executed in the execute subsystem by entering just RUN.
q _i	Specifies local files to replace files specified on the PROGRAM statement in a FORTRAN 5 program.

The first format of the command compiles and initiates execution of the primary file.

The second format of the command compiles and initiates execution of the local file lf_{n1} if the subsystem and the source program are compatible.

The third format of the command compiles the primary file and generates a binary file lf_{n2}, containing the resulting object code. The object code is not executed. File lf_{n2} can later be executed under the execute subsystem by entering the RUN command.

The fifth format of the command applies only when running a previously compiled (object code) FORTRAN program in the execute subsystem. It allows you to rename local files specified in the PROGRAM statement without recompiling the program. Parameters q₁ 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 command.

```
PROGRAM TEST (INPUT,OUTPUT,AAA,BBB,TAPE1=AAA,TAPE2=BBB)
```

Normally, to execute a program containing this command, you must define local file AAA. Local file BBB results from the execution of the program. However, you 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:

```
RUN,T,INPUT,OUTPUT,CCC,DDD
```

TEXT COMMAND

The TEXT command selects text mode. Text mode allows direct entry of information (program commands, data, or text) into the primary file without specifying line numbers. The system interprets commands entered in text mode as text and does not process them directly.

Unless changed during installation, the input line can consist of a maximum of 160 characters. The system appends data to the end of the file.

To terminate text mode, you should terminate the current input line with a carriage return, then enter the interruption sequence or the termination sequence defined for the terminal in use or the ETX character followed by a carriage return (refer to section 17).

The system automatically packs the text file to ensure that the data is in one logical record.

NOTE

Do not enter a user break to terminate text mode until you have received the ENTER TEXT MODE message at your terminal. If entered before this message, your user break clears any unprocessed input, and no data goes into the primary file.

WHATJSN COMMAND

The WHATJSN command returns the job sequence name (JSN) of the terminal you specify by a user name. When several terminals are logged in under the specified user name, a list of JSNs is returned. The WHATJSN command can be entered only from the access subsystem. It is used in conjunction with the DIAL command to send a message to a user at an interactive terminal.

Format:

```
WHATJSN,username
```

username Specifies the user name of user whose JSN you seek.

X COMMAND

The X command forces the system to interpret a command as a batch command. The X command does not explicitly change the current subsystem for your job, however.

Format:

X,ccc

ccc Any valid batch command.

This command cannot exceed 80 characters. You must type the entire command but can omit the terminator. However, you must enter a terminator after a command if you want to append a comment to the command.

This command is used to enter a batch command that either is invalid in your current subsystem or has a different function in that subsystem.

For example, the command

X,BASIC

calls the BASIC compiler instead of changing your subsystem to BASIC.

PRIMARY FILE EDITING COMMANDS

To allow ease of editing primary files, the system provides a line editing capability that can be used under any subsystem. The line editing capability allows you to:

- List lines
- Correct lines
- Delete lines
- Insert lines
- Copy lines

The following commands perform the previously indicated editing operations upon the primary file (hereafter referred to as the edit file):

ALTER - changes character strings within lines of the edit file.

DELETE - deletes lines from the edit file.

DUP - duplicates lines and moves the duplicates to a specific location in the edit file.

LIST - selectively lists lines of the edit file.

MOVE - moves lines from one location to another in the edit file.

READ - inserts another file at a specified location in the edit file.

RESEQ - resequences or adds line numbers to the edit file.

WRITE - copies lines from the edit file to the edit specified sequenced file.

WRITEN - copies lines from the edit file to an unsequenced file.

For more detailed editing, Text Editor or XEDIT should be used (refer to the Text Editor Reference Manual or the XEDIT Reference Manual).

PARAMETER FORMAT

The following subsections describe the specific format of each command. Each command uses one or more of the following parameters:

<u>Parameter Name</u>	<u>Format</u>	<u>Function</u>
Line number	n	Specifies edit file line number.
Line range	q..r	Specifies the edit set of sequential lines in the edit file. q is the beginning line number of the range. q may be specified by an asterisk to signify that the line range starts at the beginning of information (BOI). r represents the ending line number of the range. r may be specified by an asterisk to signify that the line range ends at the end of information (EOI). To specify a range from BOI to EOI you may enter a single asterisk. When using line numbers, q must be less than or equal to r.

<u>Parameter Name</u>	<u>Format</u>	<u>Function</u>
Lines	c_1, c_2, \dots, c_n	<p>Specifies a nonsequential set of lines and line ranges separated by commas. c_i represents a line or line range. For example,</p> <p style="text-align: center;">60..90,120,10..40,350</p> <p>is a valid entry. Regardless of the order the lines or line ranges appear in the parameter, the lines are processed in the order they appear in the file.</p>
Line number increment	z	<p>Generates sequential line numbers when inserting text into the file or when resequencing is required due to overlapping line numbers (refer to Line Number Overlap section). z must be in the range of one to 4095 and, when omitted, assumes a value of one.</p>
String	<code>/string/</code>	<p>Specifies a sequence of characters, including spaces, delimited by a string delimiter. The string delimiter may not consist of the following:</p> <p style="text-align: center;">digit , * : letter \$) . @ space</p> <p>The delimiter chosen may not appear in the string. In this subsection, a slant (/) is used as the string delimiter.</p>
String pair	<code>/string1/string2/</code>	<p>Specifies two different character sequences separated by a string delimiter. String1 represents a string as it appears in the file. String2 represents how string1 will appear after command processing. String1 and string2 may be null strings. A null string contains no characters. For example, if string1 were null, the string pair would appear as <code>//string2/</code>. Refer to the string parameter description for invalid string delimiters.</p>
File name	filename	<p>A file name consisting of 1 to 7 characters where each character is a letter or digit.</p>

SUPPRESSING EDITING RESPONSES

When changing lines within a file, the changed lines are displayed immediately after the edit command has been processed. If you do not want the changed lines listed, enter:

BRIEF

This command will prevent the listing of changed lines. To resume listing changed lines, enter:

NORMAL

The **BRIEF** and **NORMAL** commands are described earlier in this section.

LINE NUMBER OVERLAP

Some editing commands require the system to resequence lines of your file. New line numbers may overlap existing line numbers. If this occurs, you are asked if the command should be executed.

Example: INSERTED LINES OVERLAP EXISTING LINES, AT 130.
ENTER Y TO CONTINUE OR N TO STOP
?

If your response is **YES**, the lines will be inserted and all lines that are overlapped will have new line numbers generated in increments of the specified line number increment. If no increment is specified, new line numbers are generated in increments of one from the last inserted line. If your response is **NO**, the command is not executed.

If your edit file is a **BASIC** or **FORTRAN** program, select the corresponding subsystem before editing. If you are not under the correct subsystem, the system may not make the necessary changes when resequencing lines. Refer to the **RESEQ** command in section 9 for more information on how the system resequences **FORTRAN** and **BASIC** programs.

INSERTING LINES

You can add a line to the edit file by prefacing it with a unique line number.

The following example demonstrates how to insert a single line into a sequenced file.

Example:

```
List                                     File is listed.
00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END

READY.
00115 Let c=30                             Line is entered inserting
                                           line 115 between 110 and 120

List                                     File is listed to show
00100 LET A=1                             corrected version.
00110 LET B=10
00115 LET C=30
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

ALTER COMMAND

The ALTER command enables you to change character strings within specified lines in the edit file.

Format:

```
ALTER,c1,c2,...,cn,/string1/string2/
```

where the lines parameter is optional. If the lines parameter is omitted, the changes specified by string1 and string2 apply to the entire file.

When you enter the lines and string pair, only the occurrences of string1 appearing in the lines specified by the lines parameter are changed to string2.

When you enter the lines parameter, string1, and a null string2 (/string1//), all occurrences of string1 appearing in the lines specified in the lines parameter are deleted.

When you enter the lines parameter, a null string1, and string2 (//string2/), string2 is appended to the lines specified in the lines parameter.

To correct a single line in error, you can simply respecify the line number and write the line correctly.

The following example demonstrates how to change a line by entering the corrected version of the line:

Example:

```
List
00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
READY.
00110 let b=20
```

File is listed.

Line is entered changing B's value from 10 to 20. READY message is not given after entry of line.

```
List
00100 LET A=1
00110 LET B=20
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
0200 END
```

File is listed to show corrected version.

When you enter the lines parameter, and string1 and string2 are null (///), the lines specified in the lines parameter are listed.

Example:

```
List
00100 LET A=1
00110 LET B=20
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END

READY.
alter,100..130,/b/c/
00110 LET C=20
00120 FOR I=1 to C

READY.
List
00100 LET A=1
00110 LET C=20
00120 FOR I=1 TO C
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

File is listed.

ALTER command is entered, changing B to C in lines 100 through 130. Changed lines are displayed automatically.

File is listed to show corrected version.

DELETE COMMAND

You can delete specified lines from a sequenced file using the DELETE command.

Format:

```
DELETE,c1,c2,...,cn,/string/
```

where either the lines parameter or string parameter must be specified.

When you enter the lines parameter and do not enter the string parameter, all lines specified by the lines parameter are deleted.

When you enter the string parameter and do not enter the lines parameter, all lines in the file containing the specified string are deleted.

When you enter both the lines parameter and the string parameter, only those lines specified by the lines parameter containing the string specified in the string parameter are deleted.

Example:

```
list
00100 LET A=1
00105 LET B=10
00110 LET C=20
00120 FOR I=1 TO C
00130 A=A*I
00140 PRINT "FACTORIAL I;I,"IS ";A
00150 NEXT I
00200 END
```

File is listed.

```
READY.
delete,100..130,/b=10/
00105 LET B=10
```

DELETE command is entered,
deleting all lines from 100
through 130 which contain
the string B=10. The deleted
line is automatically listed.

```
READY.
list
00100 LET A=1
00110 LET C=20
00120 FOR I=1 TO C
00130 A=A*I
00140 PRINT "FACTORIAL I;I,"IS ";A
00150 NEXT I
00200 END
```

File is listed to show
corrected version.

You can also delete a line by specifying its line number and entering an empty line for it.

The following example demonstrates how to delete a single line from a sequenced file by entering the number of the line to be deleted.

Example:

```
List                                     File is listed.
00100 LET A=1
00105 LET B=10
00110 LET C=20
00120 FOR I=1 to C
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

```
READY.
00105                                     00105 followed by carriage
                                         return is entered, deleting
                                         line 00105. The READY message
                                         is not given after entry
                                         of line.
```

```
List                                     File is listed to show
00100 LET A=1                             corrected version.
00110 LET C=20
00120 FOR I=1 TO C
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

DUP COMMAND

You can duplicate lines and insert them in a specified location in the edit file using the DUP command.

Format:

```
DUP,q..r,n,z
```

where q..r is a line range and may be specified as a single line number.

n specifies the line number after which the duplicated the lines are to be inserted. If n is not specified, the duplicated lines are inserted at the end of the file. If n is less than the first line number in the file, the duplicated lines are inserted at the beginning of the file. If n is between two existing line numbers, the duplicated lines are inserted between the adjacent lines and incremented by z. If the new line numbers overlap existing edit file line numbers, line resequencing occurs as described under Line Number Overlap.

When BASIC statements containing line numbers and their referenced lines are duplicated, the lines and their duplicates are kept separate. The duplicate statements containing line references reference duplicate lines. Duplicate statements do not reference original statements.

Example:

```
* list
00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

File is listed.

```
READY.
dup,130..150,150,10
00160 A=A*I
00170 PRINT "FACTORIAL ";I,"IS ";A
00180 NEXT I
```

DUP command is entered, duplicating lines 130 through 150, and inserting the duplicates after line 150, incrementing subsequent lines by 10. Affected lines are automatically listed.

```
READY.
list
00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00160 A=A*I
00170 PRINT "FACTORIAL ";I,"IS ";A
00180 NEXT I
0200 END
```

File is listed to show corrected version.

LIST COMMAND

You can selectively list the lines of the edit file using the LIST command.

Format:

```
LIST,lines,/string/
```

If you specify no parameters, the system lists all lines in the file. To list all of the lines in a local file other than the edit file, you can use the alternate format of the LIST command, described earlier in this section under Job Processing Commands.

The following example illustrates how to use the command.

Example:

READY. list	List the entire file.
00100 LET A=1 00105 LET B=10 00110 LET C=20 00120 FOR I=1 TO C 00130 A=A+1 00140 PRINT "FACTORIAL ";I, "IS ";A 00150 NE T I 00200 END	
READY. list,100..130	List a range of lines.
00100 LET A=1 00105 LET B=10 00110 LET C=20 00120 FOR I=1 TO C 00130 A=A+1	
READY. list,120,100,140..200	List a nonsequential set of lines.
00100 LET A=1 00120 FOR I=1 TO C 00140 PRINT "FACTORIAL ";I, "IS ";A 00150 NEXT I 00200 END	
READY. list,/a/	List all lines that contain the character A.
00100 LET A=1 00130 A=A+1 00140 PRINT "FACTORIAL ";I, "IS ";A	
READY. list,130..200,/a/	List only those lines in a specified range that contain the character A.
00130 A=A+1 00140 PRINT "FACTORIAL ";I, "IS ";A	
READY.	

MOVE COMMAND

You may also manipulate lines of the edit file with the use of the MOVE command. The MOVE command moves lines from one location in the edit file to another location.

Format:

```
MOVE,q..r,n,z
```

where q..r is the line range parameter and may be specified by a single line number. The line number (n) and line number increment (z) parameters are optional.

The line range parameter specifies the lines to be moved.

If the n parameter is omitted, the lines are moved to the end of the file and successive line numbers are generated in increments of z, if specified, from the last line of the edit file.

If the new line numbers overlap existing line numbers in the edit file, line resequencing occurs as described in the Line Number Overlap subsection.

Example:

```
list                                     File is listed.
00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

```
READY.
move,100,110,10                          MOVE command is entered,
INSERT LINES OVERLAP EXISTING LINES, AT 120. moving line 100 and
ENTER Y TO CONTINUE OR N TO STOP          inserting it immediately
? y                                        following line 110,
00120 LET A=1                              incrementing subsequent lines
00130 FOR I=1 TO B                          by 10. Affected lines are
00140 A=A*I                                  automatically listed.
00150 PRINT "FACTORIAL ";I,"IS ";A
00160 NEXT I
```

```
READY.
list                                     File is listed to show
00110 LET B=10                              corrected version.
00120 LET A=1
00130 FOR I=1 TO B
00140 A=A*I
00150 PRINT "FACTORIAL ";I,"IS ";A
00160 NEXT I
00200 END
```

READ COMMAND

The READ command inserts the lines of a file at a specific location in the edit file.

Format:

```
READ,filename,n,z
```

where the line number (n) and line number increment (z) parameters are optional. The parameter filename is the name of the file to be inserted (read file).

When the line number parameter is entered, the contents of the file to be inserted are inserted immediately following line n. If the parameter n is a line number that is not in the edit file, the read file is inserted after the largest line number that is less than n. The line numbers of read files are incremented by 1, unless an increment is specified by the z parameter.

If the line number parameter is not entered, the contents of the read file are added to the end of the edit file and incremented by z from the last line number of the edit file. For sequenced read files, line numbers are added in increments of one, unless otherwise specified by the z parameter. A sequenced read file will be resequenced in increments of z from the last line number in the edit file.

Unsequenced read files will have a five-digit line number added and followed by a blank (in the FORTRAN and FTNLS subsystems, the line number is added but, not the blank). If a line consists of only a line number or a line number followed by zero or one blank, the line will not be inserted into the edit file.

If the new line numbers overlap existing lines in the edit file, line resequencing occurs as described in the Line Number Overlap subsection.

When operating in the BASIC subsystem, if the read file is sequenced and no line overlap occurs, statements containing line number references and the referenced lines within the read file remain local to the lines' read file.

Example:

list	
00100 LET A=1	Edit file is listed.
00110 LET B=10	
00120 FOR I=1 TO B	
00130 A=A*I	
00140 PRINT "FACTORIAL ";I,"IS ";A	
00150 NEXT I	
00200 END	
READY.	
get,skid	Read file is made local.
READY.	
read,skid,150,10	READ command is entered in-
INSERT LINES OVERLAP EXISTING LINES,	serting the read file after
AT 200.	line 150, incremented by 10
ENTER Y TO CONTINUE OR N TO STOP	
? y	Y is entered to continue
00160 LET A=1	command processing.
00170 LET B=50	
00180 FOR I=1 TO B	
00190 A=A*I	
00200 PRINT "FACTORIAL";I,A	Inserted lines are
00210 NEXT I	automatically listed.
00220 END	
READY.	
list	Edit file is listed to
00100 LET A=1	show corrected version.
00110 LET B=10	
00120 FOR I=1 TO B	
00130 A=A*I	
00140 PRINT "FACTORIAL ";I,"IS ";A	
00150 NEXT I	
00160 LET A=1	
00170 LET B=50	
00180 FOR I=1 TO B	
00190 A=A*I	
00200 PRINT "FACTORIAL";I,A	
00210 NEXT I	
00220 END	

RESEQ COMMAND

Refer to the RESEQ command in section 9.

WRITE COMMAND

You can also append lines and line numbers of the edit file to another specified file (destination file) by using the WRITE command.

Format:

```
WRITE,filename,c1,c2,...cn,/string/
```

where the required parameter filename is the name of a local file. The lines and string parameters are optional.

If the lines parameter is entered and the string parameter is omitted, all lines specified by the lines parameter are copied to the destination file. If both the lines parameter and string parameter are entered, the lines specified by the lines parameter which contain the string specified by the string parameter are copied to the destination file.

If the string parameter is entered and the lines parameter is omitted, all lines in the edit file containing the specified string are copied to the destination file. If both the lines and string parameters are omitted, all lines of the edit file are copied to the destination file.

Example:

```
list
00100 LET A=1
00110 LET B=10
00115 LET C=20
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

Edit file is listed.

```
READY.
write,file2,100..110,120..*
```

WRITE command is entered,
writing lines 100 through
110 and lines 120 through
200 to destination file
FILE2.

```
READY.
list,f=file2
```

FILE2 is listed.

```
00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

WRITEN COMMAND

To copy lines from a sequenced edit file to an unsequenced destination file, enter the WRITEN command. The WRITEN command copies specified lines of the edit file to the destination file while removing the line numbers; resulting in an unsequenced destination file.

Format:

```
WRITEN,filename,lines,/string/
```

Example:

```
List
00100 LET A=1
00110 LET B=10
00115 LET C=20
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT "FACTORIAL ";I,"IS ";A
00150 NEXT I
00200 END
```

Edit file is listed.

```
READY.
writen,file3,100..110,120..*
```

WRITEN command is entered,
writing lines 100 through
110 and lines 120 through
200 to destination file
FILE3.

```
READY.
List,f=file3
```

FILE3 is listed.

```
LET A=1
LET B=10
FOR I=1 TO B
A=A*I
PRINT "FACTORIAL ";I,"IS ";A
NEXT I
END
```

Parameter combinations and functions are the same for the WRITEN and WRITE commands.

FILE MANAGEMENT COMMANDS

9

The file management commands manipulate files assigned to your job. The commands included in this category are:

ASSIGN	COPYX	RENAME	SKIPFB
BKSP	DOCUMENT	REQUEST	SKIPR
CLEAR	EVICT	RESEQ	SORT
CONVERT	FCOPY	RETURN	TCOPY
COPY	LIST80	REWIND	TDUMP
COPYBF	LOCK	ROUTE	UNLOAD
COPYBR	LO72	SCOPY	UNLOCK
COPYCF	NEW	SETFS	VERIFY
COPYCR	OUT	SKIPEI	WRITEF
COPYEI	PACK	SKIPF	WRITER
COPYSBF	PRIMARY		

The commands in this section position files, copy data from one file to another, specify the method and format of input/output, sort files, and add corrections. You can assign your files to a specific device type; change the file type, identification code, and write interlock status; and release them from your job. You can also receive information about records in a file or documentation in a file.

If you use these commands on files structured by CYBER Record Manager (CRM), the output may not reflect the CRM logical file structure. In the case of S- or U-type records, the CRM file structure is preserved. Otherwise, the resulting NOS records will not correspond to the CRM records (refer to File Structure in section 2).

If a file is not specifically assigned through the use of an ASSIGN, LABEL, or REQUEST command, the system assigns the file to available mass storage.

ASSIGN COMMAND

The ASSIGN command directs the system to assign a file to the specified device or device type. The following descriptions refer to devices other than magnetic tape. For use of the ASSIGN command with magnetic tape, refer to section 12.

Format:

ASSIGN,nn,lfn,checkpoint.

nn Device or device type to which the specified file is to be assigned; nn may be either the EST ordinal† of a peripheral device or the device type as defined as follows:

<u>Type</u>	<u>Equipment</u>
DE	Extended memory
DI	844-21 Disk Storage Subsystem (half-track)
DJ	844-41 or 844-44 Disk Storage Subsystem (half-track)
DK	844-21 Disk Storage Subsystem (full-track)
DL	844-41 or 844-44 Disk Storage Subsystem (full-track)
DM	885 Disk Storage Subsystem (half-track)
DP	Distributive data path to extended memory
DQ	885 Disk Storage Subsystem (full-track)
DV††	819 Disk Storage Subsystem (single-density)
DW††	819 Disk Storage Subsystem (double-density)
MS	Mass storage device
NE	Null equipment
TT†††	Interactive terminals

lfn Name of the file to be assigned to the specified equipment.

checkpoint Specifies that lfn is to be used as a checkpoint file (refer to section 13).

<u>checkpoint</u>	<u>Meaning</u>
CK	Each dump is written at the previous EOI of lfn.
CB	Each dump is written at the BOI of lfn.

†Contact installation personnel for a list of EST ordinals. You must have system origin privileges (refer to the LIMITS command) to specify an EST ordinal.

††Available on model 176 only.

†††This device type applies only to interactive origin jobs.

Example 1:

ASSIGN,MS,OUTPUT.

This command assigns file OUTPUT to mass storage. With this assignment in an interactive job, you can cause output normally printed at your terminal to be written on a mass storage file instead. Here, output means information generated by a program during execution. Dayfile messages are still printed at the terminal. Once this assignment is made, output is written on the mass storage file OUTPUT until the file is returned or reassigned.

Example 2:

ASSIGN,TT,XYZ.

This command assigns file XYZ to your interactive terminal. The assignment means that input that the system would have read from file XYZ is instead solicited by a prompt at the terminal and that output that the system would have written on file XYZ is instead displayed at the terminal.

If you have multiple files assigned to your terminal, you may get some unpredictable results.

Example 3:

ASSIGN,DI,ABC.

This command assigns file ABC to an 844-21 disk drive, if one is available.

The ASSIGN command can also be used to create or access existing seven- or nine-track unlabeled tapes. For a description of the command as it applies to magnetic tape assignment, refer to section 12, Tape Management.

BKSP COMMAND

The BKSP command directs the system to bypass a specified number of logical records in the reverse direction.

Format:

BKSP,lfm,n,m.

<u>Parameter</u>	<u>Description</u>
lfm	Name of the file to be backspaced.
n	Number of logical records (decimal) to backspace; if this parameter is omitted, the system assumes n=1.
m	File mode: C for coded, B for binary. If m is omitted, the system assumes the file is in binary mode.

The BKSP request can be issued at any point in a logical record. If, for example, FILE1 were positioned within the third record, a

BKSP,FILE1.

request would reposition FILE1 to the beginning of the third record. The system does not backspace past the beginning-of-information (BOI). However, EOF indicators are considered separate records and are included in the record count.

The BKSP command has no effect on a primary file since that file is rewound before every operation.

CLEAR COMMAND

The CLEAR command releases all files currently assigned to the job. You can also specify files that are not to be released.

Format:

CLEAR.

or

CLEAR,* ,lfn₁,lfn₂,...,lfn_n.

The first format releases all files. The second format releases all files except those named. If no files are named, all files assigned to the job are released.

There are several exceptions. The CLEAR command does not release any files that have the no-auto-drop status. Certain system scratch files (for example, ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD) have this status. You can give any local file this status by using the SETFS command.

Refer to RETURN command in this section for the operations performed on each file type.

CONVERT COMMAND

The CONVERT command converts records from one character set to another.

Format:

CONVERT,P₁,P₂,...,P_i.

<u>P_i</u>	<u>Description</u>
P=lfn ₁	Input on file lfn ₁ ; if omitted, file OLD is assumed.
N=lfn ₂	Output on file lfn ₂ ; if omitted, file NEW is assumed.
RS=n ₁	Maximum record size in characters (decimal); 1<n<500. If omitted, 300 is the assumed maximum record size. (Each character is 6 bits.)

<u>Pi</u>	<u>Description</u>														
64	Convert from 63- to 64-character set; if omitted, no conversion takes place. The TS=t parameter must be specified if 64 is not.														
TS=t	Convert from old interactive 61-character set to new interactive 63-character set; t may be one of the following terminal types.														
	<table border="0" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: left;"><u>t</u></th> <th style="text-align: left;"><u>Terminal Type</u></th> </tr> </thead> <tbody> <tr> <td>TTY</td> <td>ASCII code terminal with standard print.</td> </tr> <tr> <td>COR</td> <td>Correspondence code terminal with standard print.</td> </tr> <tr> <td>CORAPL</td> <td>Correspondence code terminal with APL print.</td> </tr> <tr> <td>MEMAPL</td> <td>Memorex 1240 (ASCII code) terminal with APL print.</td> </tr> <tr> <td>BLKEDT</td> <td>Block transmission (ASCII code) terminal with full display screen editing capability and standard print.</td> </tr> <tr> <td>NAMIAF</td> <td>Virtual network terminal. Same as TTY.</td> </tr> </tbody> </table>	<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 full display screen editing capability and standard print.	NAMIAF	Virtual network terminal. Same as TTY.
<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 full display screen editing capability and standard print.														
NAMIAF	Virtual network terminal. Same as TTY.														
	If t is omitted, it is assumed to be TTY. If TS=t is omitted, no interactive conversion takes place. The 64 parameter must be specified if TS is not.														
R	Rewind input and output files before, but not after, processing. If omitted, the files are not rewound before or after processing.														
RC=n ₂	Convert n ₂ decimal records. If n ₂ is omitted, convert until an EOF is encountered. If RC=n ₂ is omitted, one record is assumed.														
NM	Used in conjunction with TS=t parameter and specifies that conversion is to normal mode; if omitted, conversion is to ASCII mode. Note the effect of conversion on the following characters.														
	<table border="0"> <tbody> <tr> <td style="padding-right: 20px;">^ (circumflex)</td> <td>If TS is specified, display code 70 (circumflex character) is converted to 76. If NM is omitted, conversion is to 7402 (ASCII mode).</td> </tr> <tr> <td style="padding-right: 20px;">: (colon)</td> <td>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).</td> </tr> </tbody> </table>	^ (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).										
^ (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 lists legal conversion using the appropriate CONVERT parameter.

<u>Type of Record</u>	<u>Legal Conversion Parameters</u>
63-character set, noninteractive record	64
Old interactive record	TS or 64 and TS
New NORMAL interactive record (equivalent to BATCH character set)	64
New ASCII interactive record	None

COPY COMMAND

The COPY command copies data from one file to another if the files are within the range of permissible formats listed in table 9-1.

NOTE

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Table 9-1. Range of Permissible Formats for the COPY Command

		Output (O=lf _{n2})						
		Mass Storage or Terminal	Tape Formats					
			I	SI	S	L	F	
Mass Storage or Terminal		Yes	Yes	Yes	Yes	Yes	No	
Input (I=lf _{n1})	Tape Formats	I	Yes	Yes	Yes	Yes	Yes	No
		SI	Yes	Yes	Yes	Yes	Yes	No
		S	Yes	Yes	Yes	Yes	Yes	No
		L	Yes	Yes	Yes	No	Yes	No
		F	Yes	Yes	Yes	No	No	Yes

Order-dependent format:

COPY, lfn₁, lfn₂, x, c, tc, copycnt, bsize, charcnt, erlimit, p₁p₂...p_n, lfn₃

Order-independent format:

COPY, I=lfn₁, O=lfn₂, V=x, M=c, TC=tc, N=copycnt, BS=bsize, CC=charcnt, EL=erlimit,
 PO=p₁p₂...p_n, L=lfn₃.

The parameters can appear in order-dependent format, order-independent format, or a combination of both. If order-dependent and order-independent parameters are mixed in one COPY command, the order-dependent parameters must appear in their proper position. All parameters are optional. However, the specification of certain parameters precludes the application of others. A nonapplicable parameter may be ignored or it may be invalid. This is stated in the individual descriptions of the parameters.

<u>Parameter</u>	<u>Description</u>	<u>Default</u>												
I=lfn ₁	Specifies the name of the file to be copied.	INPUT												
O=lfn ₂	Specifies the name of the file receiving the copy.	OUTPUT												
V=x	If the x parameter (one to seven alphanumeric characters) is present, both files are rewound, copied, rewound, verified, and rewound. The x parameter must not be zero.	No verify												
M=c	Specifies whether input or output is in coded mode.													
	<table border="1"> <thead> <tr> <th><u>c</u></th> <th><u>Description</u></th> <th></th> </tr> </thead> <tbody> <tr> <td>C1</td> <td>Coded mode is set on input only.</td> <td>Binary</td> </tr> <tr> <td>C2</td> <td>Coded mode is set on output only.</td> <td></td> </tr> <tr> <td>x</td> <td>Coded mode is set on both input and output. The value x is any alphanumeric, one-to-seven character string.</td> <td></td> </tr> </tbody> </table>	<u>c</u>	<u>Description</u>		C1	Coded mode is set on input only.	Binary	C2	Coded mode is set on output only.		x	Coded mode is set on both input and output. The value x is any alphanumeric, one-to-seven character string.		
<u>c</u>	<u>Description</u>													
C1	Coded mode is set on input only.	Binary												
C2	Coded mode is set on output only.													
x	Coded mode is set on both input and output. The value x is any alphanumeric, one-to-seven character string.													
	This parameter applies only to S and L format tapes. If coded mode is set on an SI tape, the system aborts the job. For other formats, the system ignores the mode setting.													
TC=tc	Specifies the termination condition for copying used in conjunction with N=copycnt. The termination condition can be specified as follows:	Copy to double EOF (TC=D or TC=EOD)												
	<table border="1"> <thead> <tr> <th><u>tc</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>F or EOF</td> <td>The N keyword specifies the number of files to copy.</td> </tr> <tr> <td>I or EOI</td> <td>Copy to the end of information. The N keyword is ignored.</td> </tr> </tbody> </table>	<u>tc</u>	<u>Description</u>	F or EOF	The N keyword specifies the number of files to copy.	I or EOI	Copy to the end of information. The N keyword is ignored.							
<u>tc</u>	<u>Description</u>													
F or EOF	The N keyword specifies the number of files to copy.													
I or EOI	Copy to the end of information. The N keyword is ignored.													

<u>Parameter</u>	<u>Description</u>	<u>Default</u>
	<u>tc</u>	<u>Description</u>
	D or EOD	The N keyword is the number of double EOFs to copy to. If N > 1 is specified together with this TC value, and verify is also selected, the files are verified only to the first empty file (COPY calls VERIFY with N=0 parameter).
N=copycnt	Specifies the copy count used with the copy termination condition specified by the parameter TC=tc.	1
BS=bsize	Specifies the maximum block size (in central memory words) which specifies S or L tape PRU size. This applies only when copying to or from S and L tapes. It cannot be specified with the CC=charcnt parameter.	If CC=charcnt is not specified, 1000 _g for S tape copy and 2000 _g for L tape copy.
CC=charcnt	Specifies the maximum number of characters in an S or L tape block. This parameter can be specified only when copying to or from S and L tapes. The PRU size and unused bit count are calculated from the character count. However, the unused bit count is used only when writing a full block to an S or L output tape during a copy from mass storage, I, or SI format tape. The charcnt value should be a multiple of 10. If it is not, the characters that exceed the charcnt value in the last word of the record are discarded when writing an S or L format tape. This parameter cannot be specified with the BS=bsize parameter.	Not used (the PRU size is specified by the BS=bsize parameter)
EL=erlimit	Specifies the number of nonfatal errors allowed before abort. This includes both parity errors and block-too large errors which are returned by the tape subsystem after completing recovery procedures. If EL=U is specified, unlimited error processing is allowed. Error recovery is supported on mass storage and on all tape formats but is not supported on a terminal or on unit record equipment. In the latter cases, any error aborts the job.	0 (zero)

<u>Parameter</u>	<u>Description</u>	<u>Default</u>
PO=p ₁ p ₂ ...p _n	One or more of the following processing options:	
	E Input blocks with parity errors or block-too-large errors are processed (copied).	Error blocks are skipped.
	D Any noise blocks generated by a copy from mass storage, I format tape, or SI format tape to an S or L format tape are deleted. This parameter cannot be specified on any other type of copy.	For S or L binary tapes, noise blocks are padded to noise size with binary zeros; for coded mode, they are padded with blanks.
	R Allows record splitting during a copy from mass storage, I format, or SI format to S or L format tape. This parameter cannot be specified on any other type of copy.	Record splitting is not allowed.
	M Copy files according to the copy termination condition specified by the keyword TC, eliminating each EOF on output. This option is primarily for use with labeled S and L output tapes since it eliminates the conflict of the double meaning of a tape mark on these formats (the tape mark on these formats serves as both an EOF and label group delimiter).	Copy files according to specification of the copy termination (TC), writing an EOF after each file on output.
L=lf _{n3}	Specifies the name of an alternate file to receive parity error messages when extended error processing is in effect (nonzero EL specified), in which case, file name lf _{n3} must not be the same as lf _{n1} or lf _{n2} .	OUTPUT

Example:

The following COPY command combines order-dependent and order-independent parameters.

```
COPY,FILE1,FILE2,VERIFY,CODED,EOF,6,L=MYOUT,PO=E,EL=10.
```

FILE1 is the input file, and FILE2 the output file. Six coded files are copied and verified. Up to 10 nonfatal errors are allowed, and the bad data is copied with informative error messages written to the file MYOUT.

The COPY command begins a copying operation at the current position of both files unless the verify option is specified. If verification is specified, both files are rewound before the copying begins and rewound, verified, and rewound again after the copying is completed. This verification may not be meaningful if the logical structure of the two files is incompatible.

COPY TERMINATION

Copying continues until the termination condition for copying is met or EOI is encountered. The termination condition can be a file count, a double EOF count, or EOI. If the copying is terminated by a double EOF (for TC=EOD), the second EOF is detected on lfn₁, but is not transferred to lfn₂. If lfn₁=lfn₂ the named file is read until the termination condition is satisfied or EOI is encountered.

If the copying command specifies a file count, TC=EOF, and EOI is encountered on the input file before the file count is satisfied, an additional EOF is written on the output file only if data or records have been transferred since the previous EOF was written (or since the beginning of the copying if no EOFs have been encountered).

BLOCK SIZES

Both L and F tapes may require additional field length to accommodate their maximum block size. The maximum block size for an L tape copy is specified either by the BS=bsize parameter (or its default), or it is calculated from the CC=charcnt. The maximum block size for an F tape is determined by the maximum frame or character count specified when the file was assigned. The more accurate the selection of these values which determine block size, the less are the requirements for field length, CPU time, and I/O time.

PROCESSING OPTIONS

The PO=D option specifies noise block processing, and the PO=R option specifies record splitting for copies from mass storage, I format, or SI format to S or L format tapes. Due to the incompatibilities between the logical structure of the input and output files, records may be encountered on the input file that are too small or too large to be copied directly to the S or L output tape. If the output file block size is less than noise block size, it is deleted if PO=D is specified. If PO=D is not specified, the block size is rounded to the word multiple of noise size with binary zero fill for a binary S or L tape or with blank fill for a coded S or L tape. Empty records on the input file are skipped since they cannot exist on an S or L tape. If PO=R is specified and an input file record length exceeds the S or L tape maximum block size (the PRU size as specified by BS=bsize or its default, or by CC=charcnt), it is split into multiple blocks.

The PO=M option makes it possible to copy a multifile file to a labeled S or L format tape without writing the EOF tape marks. This avoids the conflict of a tape mark serving the double purpose of defining an EOF and delimiting a label group on S and L format tapes. This is in keeping with the tendency in the computer industry to define a tape mark only as a label delimiter.

The EL=erlimit and PO=E parameters provide extended error processing.

COPYBF COMMAND

The COPYBF command copies a specified number of files from one multifile file to another.

NOTE

The COPYBF command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For S, L, or F format tape copying operations, issue a COPY command with the appropriate parameter specifications.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYBF, lfn₁, lfn₂, n, c.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to copy. If this parameter is omitted, file OUTPUT is assumed.
n	Specifies number of files (decimal) on lfn ₁ to copy. If this parameter is omitted, 1 is assumed.
c	Indicates that the copying to or from an S or L format tape should be performed in coded rather than binary mode. C is a one- to seven-character string. If coded mode is set on an SI tape, the system aborts the job. The system ignores this parameter for mass storage files and I and F format tape files.

The copying begins at the current position of lfn₁. If lfn₁=lfn₂, the file is read until the file count is satisfied or EOI is encountered.

If EOI is encountered on lfn₁ before the file count is satisfied, an additional EOF is generated on lfn₂ only if data or records have been transferred since the previous EOF was written (or since the beginning of copy if no EOFs have been encountered).

COPYBR COMMAND

The COPYBR command copies a specified number of records from one file to another.

NOTE

The COPYBR command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYBR, lfn₁, lfn₂, n, c.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to receive the copy. If this parameter is omitted, file OUTPUT is assumed.
n	Specifies the number of records (decimal) to copy; if this parameter is omitted, 1 is assumed.
c	Indicates that the copying to or from an S or L format tape should be performed in coded rather than binary mode. The value C is an alpha-numeric one- to seven-character string. If coded mode is set on an SI tape, the system aborts the job. The system ignores the mode setting for other formats.

The copying begins at the current position of lfn₁. EOF indicators are considered separate records and are included in the record count. If lfn₁=lfn₂, the file is read until the record count is satisfied or EOI is encountered.

If EOI is encountered on lfn₁ before the record count is satisfied, an additional EOR is written on lfn₂ only if data has been transferred since the previous EOR or EOF was written (or since the beginning of the copy if no EORs or EOFs have been encountered).

COPYCF COMMAND

The COPYCF command copies a specified number of coded files from one file to another. A coded file is defined as a file containing lines of 160 characters or less, each terminated by a zero byte (12 zero bits in the lowest byte of a word).

NOTE

The COPYCF command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

The COPYCF command cannot copy SI format tapes. If coded mode is set for an SI tape, the system terminates the job. The TCOPY command converts SI coded tape files.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYCF, lfn₁, lfn₂, n, fchar, lchar, na.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to receive the copy. If this parameter is omitted, file OUTPUT is assumed.
n	Specifies the number of files (decimal) to copy. If this parameter is omitted, 1 is assumed.
fchar [†]	Specifies the first character position of each line to copy for 6-bit characters. If this parameter is omitted, the copying begins at character position 1.
lchar [†]	Specifies the last character position of each line to copy for 6-bit characters. lchar must be greater than or equal to fchar. If this parameter is omitted, the copying ends at character position 136.
na	Specifies that the job step should not abort when a line terminator does not appear before an EOR. na can be any alphanumeric, one-to-seven-character string.

[†]Since many characters in 6/12 display code require 12 bits rather than 6, this parameter may produce unforeseen results when copying a file containing 6/12 display code data. Refer to appendix A for a description of character set codes.

The copying begins at the current position of lfn₁. If lfn₁=lfn₂, the file is read until the file count is satisfied or EOI is encountered. If EOI is encountered before the file count is satisfied, an EOF is written on lfn₂, and the operation terminates. If a line is encountered that has more than lchar characters, the excess characters are truncated.

The COPYCF command writes lines with an even number of characters. If an input line has an odd character count and the last character is a blank not immediately preceded by a colon, the last character is removed. If an input line has an odd character count and the last character is not a blank or is a blank immediately preceded by a colon, an additional trailing blank is appended.

If the COPYCF command attempts to copy a line longer than 160 6-bit characters, the line is truncated, and an informative message is issued to the dayfile after the copying completes.

If the last line of a record does not have an end-of-line terminator, the system issues a dayfile message. If the na parameter is not specified, the job step then aborts.

COPYCR COMMAND

The COPYCR command copies a specified number of coded records from one file to another. A coded record contains lines of 160 characters or less, each terminated by a zero byte (12 zero bits in the lowest byte of a word).

NOTE

The COPYCR command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

The COPYCR command cannot copy SI format tapes. If coded mode is set for an SI tape, the system terminates the job. The TCOPY command converts SI coded tape files.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYCR,lfn₁,lfn₂,n,fchar,lchar,na.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to receive the copy. If this parameter is omitted, file OUTPUT is assumed.
n	Specifies the number of records (decimal) to copy. If this parameter is omitted, 1 is assumed.
fchar	Specifies the first character position of each line to copy for 6-bit characters. If this parameter is omitted, the copying begins at character position 1.†
lchar	Last 6-bit character position of each line to copy for 6-bit characters. lchar must be greater than or equal to fchar. If this parameter is omitted, the copying ends at character position 136.†
na	Specifies that the job step should not abort if a line terminator does not appear before an EOR. The parameter na can be any alphanumeric, one- to seven-character string.

†Since many characters in the 6/12 display code require 12 bits rather than 6, this parameter may produce unforeseen results when copying a file containing 6/12 display code data. Refer to appendix A for a description of character set codes.

The copying begins at the current position of lfn₁. If lfn₁=lfn₂, the file is read until the record count is satisfied or EOI is encountered. EOF indicators are considered separate records and are included in the record count. If the EOI is encountered before the record count is satisfied, an EOR is written on lfn₂, and the operation terminates. The COPYCR command is processed in exactly the same manner as the COPYCF command except that n specifies the number of records rather than the number of files.

If the COPYCR command attempts to copy a line longer than 160 6-bit characters, the line is truncated, and an informative message is issued to the dayfile after the copying completes.

If the last line of a record does not have an end-of-line terminator, the system issues a dayfile message. If the na parameter is not specified, the job step then aborts.

COPYEI COMMAND

The COPYEI command copies one file to another. The copying begins at the current position of the file and continues until the EOI is encountered. The EOI is not defined for certain tape formats (refer to table 2-1).

NOTE

The COPYEI command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYEI,lfn₁,lfn₂,x,c.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to receive the copy. If this parameter is omitted, file OUTPUT is assumed.
x	Specifies that both files are rewound before the copying, and rewound, verified, and rewound again after the copying is complete. x can be any alphanumeric, one- to seven-character string.
c	Indicates that the copying to or from an S or L format tape should be performed in coded rather than binary mode. If coded mode is set on an SI tape, the system aborts the job. For other formats, the system ignores the mode setting. c can be any alphanumeric, one- to seven-character string.

If lfn₁=lfn₂, the file is read until EOI is encountered.

COPYSBF COMMAND

The COPYSBF command copies a file where the first character of each line is not a printer control character and is to be printed.

NOTE

The COPYSBF command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYSBF, lfn₁, lfn₂, n, na.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to receive the copy. If this parameter is omitted, file OUTPUT is assumed.
n	Specifies the number of files (decimal) to copy. If this parameter is omitted, 1 is assumed.
na	Specifies that the job step should not abort if a line terminator does not appear before an EOR. na can be any alphanumeric, one- to seven-character string.

The COPYSBF command copies n files beginning at the current position of lfn₁ to file lfn₂, shifting each line image one character to the right and adding a leading space. Each line image may contain up to 160 (6-bit) characters. Any characters beyond 160 are lost. A page eject character is inserted at the beginning of each logical record (refer to appendix I for a list of carriage control characters). If lfn₁=lfn₂, n files are skipped but no data transfer occurs. If the EOI is encountered before the file count is satisfied, an EOF is written to lfn₂, and the operation terminates.

If the COPYSBF command attempts to copy a line longer than 160 6-bit characters, the line is truncated, and an informative message is issued to the dayfile after the copying completes.

If the last line of a record does not have an end-of-line terminator, the system issues a dayfile message. If the na parameter is not specified, the job step then aborts.

COPYX COMMAND

The COPYX command copies logical records of a file until a user-specified condition is met.

NOTE

The COPYX command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

COPYX, lfn₁, lfn₂, x, b, c.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copy. If this parameter is omitted, file INPUT is assumed.
lfn ₂	Specifies the name of the file to receive the copy. If this parameter is omitted, file OUTPUT is assumed.
x	Provides additional copy specifications. If omitted, one record is copied. The value for x may be one of the following:
<u>x</u>	<u>Description</u>
n	Copy n records, when n is a decimal number.
00	Copy all records up to and including first zero-length record.
name	Copy all records up to and including record of specified name (record name is first seven characters of record or the name in the prefix table, if present).
type/name	Copy all records up to and including record of specified type and name (refer to Library Record Types in section 16 for list of valid record types).

<u>Parameter</u>	<u>Description</u>
b	Specifies the backspace control. If omitted, 0 is assumed.
	<u>b</u>
	<u>Description</u>
	0 Do not backspace.
	1 Backspace file lfn ₁ one record after copying completes.
	2 Backspace file lfn ₂ one record after copying completes.
	3 Backspace files lfn ₁ and lfn ₂ one record after copying completes.
c	Indicates that the copying to or from an S or L format tape should be performed in coded rather than binary mode. If coded mode is set on an SI tape, the system aborts the job. For mass storage files and I and F format tape files, the mode setting is ignored. c can be any alphanumeric, one- to seven-character string.

The COPYX command copies logical records from lfn₁ to file lfn₂ at the current position of lfn₁ until the condition specified by x is met. It then backspaces the files according to the value specified by the b parameter. If an EOF or EOI is encountered on lfn₁ before the condition specified by x is met, the operation terminates and the backspace parameter b is ignored. If lfn₁=lfn₂, the file is read until the termination condition is satisfied or an EOF or EOI is encountered.

If EOI is encountered on lfn₁ before the termination condition is satisfied, an additional EOR is written on lfn₂ only if data has been transferred since the previous EOR was written (or since the beginning of the copying if no EORs have been encountered).

DOCUMENT COMMAND

The DOCUMENT command extracts either the external or internal documentation from a file.

Format:

DOCUMENT, p₁, p₂, ..., p_n.

The parameters p_i can be in any order and must be in one of the following forms.

<u>Parameter</u>	<u>Description</u>
Omitted	The first default value is assumed.
a	The alternate default value is assumed.
a=x	x is substituted for the assumed value.

Any numeric parameter can be specified with a postradix character of either B or D. The values that P_i can assume are:

<u>P_i</u>	<u>Description</u>												
I=lf n_1	Name of the file that contains the page footing information; this must be a single statement in the following format.												
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>Column(s)</u></th> <th style="text-align: left;"><u>Contents</u></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Blank</td> </tr> <tr> <td>2-45</td> <td>Document title</td> </tr> <tr> <td>46-55</td> <td>Publication number</td> </tr> <tr> <td>56-60</td> <td>Revision level</td> </tr> <tr> <td>61-70</td> <td>Revision date</td> </tr> </tbody> </table>	<u>Column(s)</u>	<u>Contents</u>	1	Blank	2-45	Document title	46-55	Publication number	56-60	Revision level	61-70	Revision date
<u>Column(s)</u>	<u>Contents</u>												
1	Blank												
2-45	Document title												
46-55	Publication number												
56-60	Revision level												
61-70	Revision date												
S=lf n_2	Name of the file containing the source statement images from which to extract the documentation. This file is rewound by default unless the NR parameter is specified.												
L=lf n_3	Name of the file on which the output is to be written.												
N=nn	Number of copies to be produced.												
T=type	Documentation type:												
	<table border="0" style="margin-left: 40px;"> <tbody> <tr> <td style="padding-right: 20px;">INT</td> <td>Internal documentation (detailed description of the internal features of the software).</td> </tr> <tr> <td>EXT</td> <td>External documentation (detailed description of the external features of the software).</td> </tr> </tbody> </table>	INT	Internal documentation (detailed description of the internal features of the software).	EXT	External documentation (detailed description of the external features of the software).								
INT	Internal documentation (detailed description of the internal features of the software).												
EXT	External documentation (detailed description of the external features of the software).												
C=cc	Key character for documentation.												
P=pp	Number of print lines per page.												
NR	Disable rewind on the S (source) file.												
NT	Negate table generator.												
TC	List table of contents.												

The following are the default values for the parameters described.

<u>Parameter</u>	<u>First Default</u>	<u>Alternate Default</u>	<u>Comment</u>
I	0	INPUT	Page footing information; if I is 0, no footing information is printed.
S	COMPILE	SOURCE	Source statement images.
L	OUTPUT	OUTPUT	List file.
N	1	1	Number of copies (decimal).
T	EXT	INT	Documentation type.
C	*	03	Check character (two octal digits).
P	60	80	Number of print lines per page.
NR	REWIND	NO REWIND	Source file rewind status.
NT	ON	OFF	Table generator status.
TC	OFF	ON	Table of contents status.

Refer to Volume 4, Program Interface, for a detailed explanation of the documentation standards followed. Volume 4 also contains an example of external and internal documentation for a sample program.

EVICT COMMAND

The EVICT command releases file space for the specified files but does not release file assignment to the job unless you have the file assigned to a magnetic tape or cannot write on it. In this case, the system releases the file assignment to the job.

Format:

EVICT, lfn₁, lfn₂, ..., lfn_n.

lfn_i Name(s) of the file(s) to be evicted.

The operation that EVICT performs depends on the file characteristics.

<u>File</u>	<u>EVICT Action</u>
Permanent and primary files with write permission	Releases all file space except the first track and writes an EOI on the first sector of the first track, but keeps file assigned to the job.
Deferred routed file†	Releases all file space and clears all file routing information.

†Refer to the ROUTE command in this section.

<u>File</u>	<u>EVICT Action</u>
Read-only file	Unloads file.
Tape files	Releases tape from the job.
All other files	Releases file space, but keeps file assigned to job.

Evicting a tape file performs the same function as an UNLOAD command and so cannot be used to decrease the number of resource units scheduled via the RESOURC command.

FCOPY COMMAND

The FCOPY command converts a file from one code set to another. Currently, the only supported conversion is from 6/12 display code (used in interactive ASCII mode) to 12-bit ASCII code. Refer to appendix A for more information on code sets.

Format:

FCOPY,P=lf_{n1},N=lf_{n2},PC=cs₁,NC=cs₂,R.

<u>Parameter</u>	<u>Description</u>										
P=lf _{n1}	Specifies the file to be converted (default is OLD). Assign lf _{n1} to the job before performing the FCOPY operation.										
N=lf _{n2}	Specifies the file on which the converted data from lf _{n1} is written (default is NEW). If lf _{n2} is not assigned to the job, FCOPY creates it.										
PC=cs ₁	Specifies the code set of lf _{n1} . The default value for cs ₁ is ASCII.										
	<table border="1"> <thead> <tr> <th><u>cs₁</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>DIS</td> <td>6-bit display code with zero-byte line termination.</td> </tr> <tr> <td>ASCII</td> <td>6/12 display code with zero-byte line termination.</td> </tr> <tr> <td>ASCII8</td> <td>ASCII 8-bit characters, right-justified in 12-bit bytes with zero-byte line termination.</td> </tr> <tr> <td>ASC8</td> <td>ASCII 8-bit characters, right-justified in 12-bit bytes with ASCII unit separator line termination.</td> </tr> </tbody> </table>	<u>cs₁</u>	<u>Description</u>	DIS	6-bit display code with zero-byte line termination.	ASCII	6/12 display code with zero-byte line termination.	ASCII8	ASCII 8-bit characters, right-justified in 12-bit bytes with zero-byte line termination.	ASC8	ASCII 8-bit characters, right-justified in 12-bit bytes with ASCII unit separator line termination.
<u>cs₁</u>	<u>Description</u>										
DIS	6-bit display code with zero-byte line termination.										
ASCII	6/12 display code with zero-byte line termination.										
ASCII8	ASCII 8-bit characters, right-justified in 12-bit bytes with zero-byte line termination.										
ASC8	ASCII 8-bit characters, right-justified in 12-bit bytes with ASCII unit separator line termination.										
NC=cs ₂	Specifies the code set of lf _{n2} . The default value for cs ₂ is ASCII8. The possible values for cs ₂ are the same as for cs ₁ .										
R	Rewinds lf _{n1} and lf _{n2} before and after the conversion. The default is no rewinding.										

FCOPY reads lfn₁ to its EOI, preserving its EOR and EOF marks on the converted file. The maximum line length that can be processed is 160 12-bit codes or 320 6-bit codes. Lines that exceed the maximum length are truncated.

NOTE

If lfn₁ is written in 6/12 display code based on the 63-character set, it must be converted to the 64-character set by the CONVERT command before its conversion by the FCOPY command.

Files converted to 12-bit ASCII code can be listed on a local batch printer (refer to the ROUTE command) but cannot be listed at a interactive or remote batch terminal.

Example:

If you are at an interactive terminal and want to print a file (FILE1) created in ASCII mode, enter a COPYSBF command to prefix the file lines with appropriate carriage control characters. Then enter an FCOPY command to convert the file containing 6/12 display code (FILE2) to a file containing 12-bit ASCII code (FILE3). Finally, route the converted file (FILE3) to a line printer that prints the ASCII graphic 95-character set.

```
/ascii
/copy,file1.
AaBbCcDdEeFfGg
HhIiJjKkLlMmNn
EOI ENCOUNTERED.
/rewind,file1.
REWIND,FILE1.
/copysbf,file1,file2.
END OF INFORMATION ENCOUNTERED.
/rewind,file2.
REWIND,FILE2.
/copy,file2.
1AaBbCcDdEeFfGg
HhIiJjKkLlMmNn
EOI ENCOUNTERED.
/fcopy,p=file2,n=file3,r.
FCOPY COMPLETE.
/route,file3,dc=lp,ec=a9.
ROUTE COMPLETE.
```

The following is the local batch output from the ROUTE command.

```
AaBbCcDdEeFfGg
HhIiJjKkLlMmNn
```

LIST80 COMMAND

The LIST80 routine reads a file containing list output produced by the COMPASS assembler and compresses it to 80 columns, which fits on 8-1/2-inch by 11-inch printer paper.

Format:

```
LIST80,lfn1,lfn2,NR.
```

<u>Parameter</u>	<u>Description</u>
lf _{n1}	File from which to copy; if this parameter is omitted, file LIST is assumed.
lf _{n2}	File to which to copy; if this parameter is omitted, file OUTPUT is assumed.
NR	Parameter indicating that lf _{n1} should not be rewound.

The output listing omits the following information that appears on the COMPASS assembler output listing.

- COMPASS version number on the page heading.
- COMPASS assembler binary values.
- INVENTED SYMBOLS comment.
- Symbolic Reference Table block column that contains either the system text file name, the overlay name, or the name of the block containing the symbol.

Comments are truncated to column 65.

LOCK COMMAND

The LOCK command prevents writing on a file.

Format:

```
LOCK,lfn1,lfn2,...,lfni.
```

lf_{ni} Logical file name of a local file.

With the LOCK command, you can set the write interlock for a local file. Subsequently, the system allows only read operations on the file.

L072 COMMAND

The L072 command reformats files.

Format:

L072,p₁,p₂,.....p_n.

Enter any of the following parameters in any order:

<u>P_i</u>	<u>Description</u>
I	Reformat parameters are on file INPUT.
I=lf _n ₁	Reformat parameters are on file lf _n ₁ .
I=0	There is no input file of reformat parameters. If the I parameter is omitted, I=0 is assumed.
S	Data to be reformatted is on file SCR.
S=lf _n ₂	Data to be reformatted is on file lf _n ₂ . If the S parameter is omitted, SCR is assumed.
L	Reformatted data is listed on file OUTPUT.
L=lf _n ₃	Reformatted data is listed on file lf _n ₃ . If the L parameter is omitted, OUTPUT is assumed.
T	File to be reformatted is of type B.
T=x	File to be reformatted is of type x.

<u>x</u>	<u>Description</u>
M	Modify source data.
C	COMPASS source data.
B	Other source data.

If the T parameter is omitted, B is assumed.

H	Number of characters per output line is 72.
H=xxx	Number of characters per output line is xxx (maximum allowed is 160 characters). If the H parameter is omitted, 72 is assumed.

NOTE

H must be greater than or equal to the number of characters being moved (Nx) plus the starting column number of the destination field (Ox).

LP	Output is formatted for the line printer.
NR	Output file is not rewound.

<u>Pi</u>	<u>Description</u>
Nx=y	Specifies the number of characters to be moved (up to six fields).
x(1 to 6)	Number of field being moved.
y	Number of characters being moved.

NOTE

The following restrictions apply to the H, N, I, and O parameters.

(Nx+Ix).GT.150	Yields an error (1<x<6).
(Nx+Ox).GT.H	Yields an error (1<x<6).
H.GT.150	Yields an error.

Ix=y	Specifies the field from which the data originates.
x(1 to 6)	Number of field being moved.
y	Starting column of originating field.

Ox=y	Specifies the destination field to which the data is going.
x(1 to 6)	Number of the field to receive data.
y	Starting column of destination field.

IT Suppresses query to terminal asking if you want to change any of the input parameters before processing begins. If omitted, query is issued. This parameter is effective only from interactive origin jobs.

The following shows the default values assumed for the N, O, and I parameters for the various source types.

<u>Type</u>	<u>N1</u>	<u>I1</u>	<u>O1</u>	<u>N2</u>	<u>I2</u>	<u>O2</u>	<u>N3</u>	<u>I3</u>	<u>O3</u>
B	72	1	1	0	0	0	0	0	0
C	7	9	1	50	41	8	15	112	58
M	2	6	1	48	10	3	22	82	51

The remaining parameters of these types are defaulted to 0.

L072 reformats files (output files in general). You can rearrange each line (all lines must be formatted the same) in the format you choose. All default values compress output to 72 columns, which is appropriate for terminal output or 8-1/2-inch by 11-inch printer paper. If a l is encountered in column 1 (the page eject printer control character), the next two lines of source data are processed as a two-line header. This header is compressed to 72 columns for all source types. If no page eject control characters are encountered, no headers are processed.

The following values apply to the first line of header and cannot be changed.

N1=42, I1=8, O1=0 (if LP not specified; otherwise, O1=1).

N2=20, I2=90, O2=42.

N3=5, I3=115, O3=62.

N4=5, I4=121, O4=67.

The subheader lines for COMPASS and Modify listings are processed uniquely.

For B listings, the following values apply to the reformatting.

N1=43, I1=8, O1=0 (if LP not specified; otherwise, O1=1).

N2=29, I2=70, O2=43.

All parameters are passed to L072 by the command. If an input file is specified, L072 reads it for additional input parameters. If the job originates from an interactive terminal, and the IT parameter is not specified, you are asked if you wish to change any of the input parameters. If you enter YES, the system prints the current parameter values and allows you to change them individually. Pressing the carriage return key for any parameter leaves the parameter at its former value. In the following examples, the same input parameters are entered in three possible ways.

Examples:

Noninteractive entry of parameters from a terminal:

```
/lo72,i=0,s=source,t=b,l=out,n4=1,i4=2,o4=75,h=90
```

Interactive entry of parameters:

```
/lo72
DO YOU WANT TO CHANGE ANY CONTROL ARGUMENT VALUES-
ENTER: YES OR NO
? yes (CR)
ARGUMENT          VALUE
INPUT FILE NAME:          ? (CR)
SOURCE FILE NAME:   SCR   ? source (CR)
OUTPUT FILE NAME:   OUTPUT ? out (CR)
SOURCE FILE TYPE:   BATCH ? b (CR)
OUTPUT LINE LENGTH: 72 CHARS.? 90 (CR)
NO. OF MOVED FROM MOVED TO
CHARS.   COLUMN   COLUMN
(XO (NX) (IX)     (OX)
1. 72      1       1
2. 0       0       0
3. 0       0       0
4. 0       0       0
5. 0       0       0
6. 0       0       0
ENTER CHANGES IN THE FOLLOWING FORMAT
NX=AA*CR*
IX=BB*CR*
OX=CC*CR*
ETC.
TO CONTINUE, ENTER *CR* ONLY. ? n=4=1 (CR)
? i4=2 (CR)
? o4=75 (CR)
? (CR)
L072 COMPLETE.
```

Batch job entry:

```
L072,I.
```

where the input file has the following form:

```
S=SOURCE,L=OUT,T=B.
N4=1,I4=2,O4=75.
H=90.
-EOR-
```

Each line in the input file must end with a terminator.

NEW COMMAND

The NEW command creates a primary file.

Format:

NEW, lfn/ND.

<u>Parameter</u>	<u>Description</u>
lfn	Name of file to be made primary file.
ND	If this parameter is specified, no files currently assigned to the job are released.

The NEW command creates an empty file and makes it your new primary file. If you do not specify the ND parameter, the system releases all files assigned to the job, except those with a no-auto-drop status. Certain system files (for example, ZZZZZC0, ZZZZZC1, ZZZZZC2, and ZZZZZLD) always have this status. You can give any local file this status by using the SETFS command. When the ND parameter is specified, any currently existing primary file becomes a nonprimary temporary file.

The automatic rewinding of primary files is incompatible with some file manipulation commands. Refer to the note in PRIMARY command later in this section.

OUT COMMAND

The OUT command queues files that you or the system has given a deferred routing (refer to the ROUTE command, which is described later in this section). The command initiates the queueing that the system automatically does at job termination.

Format:

OUT.

or

OUT,*, lfn₁, lfn₂, ..., lfn_n.

The first format queues all eligible files. The second format queues all eligible files except those named (lfn_i). If no files are named, all eligible files are queued.

Any file given a deferred routing can be processed by the OUT command. Eligible files also include local mass storage files named OUTPUT, PUNCH, PUNCHB, and P8. If selected, OUTPUT is queued for printing and files PUNCH, PUNCHB, and P8 are queued for punching.

File PUNCH is punched as a coded deck in either 026 or 029 mode, depending on the origin of the job. If the job is a local batch job, punch is punched in the initial keypunch mode of the job's command record. For all other job origin types, punch is punched in the system default keypunch mode.

File PUNCHB is punched as a binary deck in system binary format. File P8 is punched as a binary deck in 80-column.

PACK COMMAND

The PACK command removes all EOR and EOF marks from a specified file and copies it as one record to another file.

Format:

PACK,lfn₁,lfn₂,x.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of file to be packed. From batch origin, lfn ₁ must be specified; from interactive origin, the primary file is used if lfn ₁ is omitted. The file must not be assigned to a interactive terminal (file type TT).
lfn ₂	Specifies the name of file to receive packed data. If lfn ₂ is omitted, the packed file is written on lfn ₁ .
x	Indicates that lfn ₁ should not be rewound before the pack occurs. x can be any alphanumeric, one- to seven-character string.

The input file, lfn₁, may consist of any number of records or files. If no third parameter is supplied, lfn₁ is read from the BOI to the EOI, and all EOR and EOF marks are removed. It is written to file lfn₂ at the current position as one record. File lfn₂ is rewound after the pack; lfn₁ is not.

NOTE

Do not use the PACK command with S, L, or F format tapes.

PRIMARY COMMAND

The PRIMARY command makes a local file the primary file, or it creates an empty primary file.

Format:

PRIMARY,lfn.

lfn	Specifies the name of local file.
-----	-----------------------------------

If lfn already exists, it must be a local mass storage file in order to be made the primary file. If lfn does not exist, the PRIMARY command creates it on mass storage. Any currently existing primary file (other than the lfn specified) becomes a nonprimary temporary file. If the specified file is already primary, the operation is ignored.

NOTE

The primary file is rewound before every command is processed. Therefore, the file manipulation commands cannot be used to position within the file. Also, the primary file is rewound after the completion of any of the commands. An attempt to add to the file using one of the copying commands may result in writing over existing data at the BOI.

RENAME COMMAND

The RENAME command changes the name of a local file.

Format:

RENAME,nfn₁=ofn₁,nfn₂=ofn₂,...,nfn_n=ofn_n.

<u>Parameter</u>	<u>Description</u>
nfn _i	Specifies the new name of the local file.
ofn _i	Specifies the existing name of the local file.

The RENAME command changes the name of the local file ofn_i to nfn_i. This does not change the names of files in the permanent file system. The renaming is done in the following manner:

- If local file nfn_i did not previously exist, file ofn_i is simply renamed.
- If a local file nfn_i already exists and is not the primary file, the system discards nfn_i and gives ofn_i the name nfn_i.
- If nfn_i is the primary file, then the system discards nfn_i, gives ofn_i the name nfn_i, and makes it the primary file. In this case, ofn_i must have a local file type.

For example, you have only two files assigned to the job. File A is a local mass storage file and file B has a queued file type. If you issue the following request

RENAME,X=A.

file A is renamed file X, and its file type (local) is not changed. If you issue the request

RENAME,B=A.

file B no longer exists; file A is renamed file B and retains its local file type.

REQUEST COMMAND

The REQUEST command assigns a file to receive checkpoint dumps or sends a message to the system operator requesting that the named file be assigned to the device described in the comment field.

Format:

REQUEST, lfn, checkpoint, comment

<u>Parameter</u>	<u>Description</u>						
lfn	Specifies the name of the file to be assigned to a specified device or to be used as a checkpoint file.						
checkpoint	Specifies that lfn is to be used as a checkpoint file.						
	<table><thead><tr><th><u>checkpoint</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>CK</td><td>Each new dump is written at EOI of lfn.</td></tr><tr><td>CB</td><td>Each new dump is written at BOI of lfn.</td></tr></tbody></table>	<u>checkpoint</u>	<u>Description</u>	CK	Each new dump is written at EOI of lfn.	CB	Each new dump is written at BOI of lfn.
<u>checkpoint</u>	<u>Description</u>						
CK	Each new dump is written at EOI of lfn.						
CB	Each new dump is written at BOI of lfn.						
comment	Specifies the comment that is displayed at the system console. In the comment field, you direct the operator to make the requested device assignment.						

If the REQUEST command is used to request assignment of file lfn to the equipment specified in the comment field, lfn must not be a local file.

If the REQUEST command is used to assign lfn for checkpoint dumps, lfn must be a local file and either the CK or CB keyword is specified. These keywords are used in conjunction with the CKP and RESTART commands; they do the following:

- Save all checkpoint dumps by appending each dump to checkpoint file lfn:
REQUEST, lfn, CK.
- Save only the last checkpoint dump by writing each dump at the beginning of checkpoint file lfn:
REQUEST, lfn, CB.
- Save two consecutive checkpoint dumps by alternately writing on two checkpoint files:
REQUEST, lfn₁, CB.
REQUEST, lfn₂, CB.

If the CK keyword is specified for different files or if more than two checkpoint files are specified, the job aborts. The CK and CB keywords specify a checkpoint file that is local to the job. You can make the checkpoint file permanent by placing a DEFINE command† before the REQUEST command.

```
DEFINE, lfn.  
  
REQUEST, lfn, CK.  
  
CKP.
```

†Any mass storage file used as a checkpoint file must have write permission.

You are not required to supply a REQUEST command to define a checkpoint file. You can use an ASSIGN or LABEL command or you can use default values.

If no REQUEST command specifying a checkpoint file has been detected when the first CKP command is encountered, the system requests a device, specifies a file name of CCCCCC, and selects the CK option. For a subsequent restart job, however, the system assumes you have made the checkpoint file available.

If lfn is a local file when you enter the REQUEST command, no new assignment is made and job processing continues with the next command. However, you can reassign lfn by issuing a RETURN command on the file before entering the REQUEST command.

You can use the REQUEST command to assign a file to a mass storage device. However, you must be authorized to assign a file to a magnetic tape or auxiliary device.† If you do not have this authorization and attempt to request a tape unit or auxiliary device, the system aborts the job.

The REQUEST command can also be used to create or access existing seven- or nine-track unlabeled tapes. If a magnetic tape assignment is needed to satisfy a REQUEST command, the MT or NT parameter should be specified. For a description of magnetic tape assignment with the REQUEST command, refer to section 12, Tape Management.

† Refer to LIMITS Command in section 7.

RESEQ COMMAND

The RESEQ command resequences source files which have leading sequence numbers or adds sequence numbers to an unsequenced file. The RESEQ command for interactive jobs acts only upon the primary file and has a different parameter order.

Format for noninteractive jobs:

RESEQ, lfn, type, start, step.

Format for interactive jobs:

RESEQ, start, step, type.

<u>Parameter</u>	<u>Description</u>										
lfn	Specifies the sorted file to be resequenced. The system does not sort lfn (refer to the SORT command).										
type	Specifies the type of file:										
	<table><thead><tr><th><u>type</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>B</td><td>BASIC source code. The system updates BASIC statements containing line references.</td></tr><tr><td>T</td><td>Text source information; a five-digit sequence number plus a blank is added at the beginning of each line. The file text, however, is not inspected. Thus, if line numbers currently exist, two set of line numbers result.</td></tr><tr><td>F</td><td>Files that contain FORTRAN source code. The system adds a five-digit line number to the beginning of each line without a line number but adds no blanks. If your FORTRAN source file does not have sequence numbers, specify this parameter to resequence the file.</td></tr><tr><td>other or omitted</td><td>Any number at the beginning of a line is considered a sequence number and is resequenced according to the start and step parameters; numbers are added to lines where no leading sequence numbers are present. You can use this option with a FORTRAN source file without destroying the integrity of statement labels if the file has leading sequence numbers.</td></tr></tbody></table>	<u>type</u>	<u>Description</u>	B	BASIC source code. The system updates BASIC statements containing line references.	T	Text source information; a five-digit sequence number plus a blank is added at the beginning of each line. The file text, however, is not inspected. Thus, if line numbers currently exist, two set of line numbers result.	F	Files that contain FORTRAN source code. The system adds a five-digit line number to the beginning of each line without a line number but adds no blanks. If your FORTRAN source file does not have sequence numbers, specify this parameter to resequence the file.	other or omitted	Any number at the beginning of a line is considered a sequence number and is resequenced according to the start and step parameters; numbers are added to lines where no leading sequence numbers are present. You can use this option with a FORTRAN source file without destroying the integrity of statement labels if the file has leading sequence numbers.
<u>type</u>	<u>Description</u>										
B	BASIC source code. The system updates BASIC statements containing line references.										
T	Text source information; a five-digit sequence number plus a blank is added at the beginning of each line. The file text, however, is not inspected. Thus, if line numbers currently exist, two set of line numbers result.										
F	Files that contain FORTRAN source code. The system adds a five-digit line number to the beginning of each line without a line number but adds no blanks. If your FORTRAN source file does not have sequence numbers, specify this parameter to resequence the file.										
other or omitted	Any number at the beginning of a line is considered a sequence number and is resequenced according to the start and step parameters; numbers are added to lines where no leading sequence numbers are present. You can use this option with a FORTRAN source file without destroying the integrity of statement labels if the file has leading sequence numbers.										
start	Specifies the new line number of the first line. The default is 100.										
step	Specifies the increment to be added to xxx for each succeeding line number. The default is 10.										

Files that have leading sequence numbers include FORTRAN and BASIC source files formatted for interactive processing. If the file has no leading sequence numbers, five-digit numbers are inserted at the beginning of each line. No line number can exceed 99999. When the system encounters line numbers longer than five digits, it uses only the first five digits to resequence the file.

When resequencing a BASIC source program, you must specify B for the file type parameter or enter the command under the BASIC subsystem (refer to section 8) so that the system changes the line number references within the source statements. The system supplies five-digit line numbers and line number references; excess surrounding blanks are used in the expansion of line number references.

Example:

File X contains the following BASIC source statements.

```
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+130
135 END
```

The following command changes the contents of file X.

```
RESEQ,X,B,90,10.
```

File X now has the following contents:

```
00090 ON SGN(A)+2 GOTO 00100,00120,00140 'COMMENT
00100 PRINT "A IS NEGATIVE"
00110 GOTO 00150                          'COMMENT
00120 PRINT "A IS ZERO"
00130 GOTO 00150                          'COMMENT
00140 PRINT "A IS POSITIVE"
00150 LET B=A+130
00160 END
```

The RESEQ command changes the line numbers and the line number references. Line numbers now begin at 90 and increment by 10. The comment on the first line is moved to the right to allow for the expanded line number references.

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

RETURN COMMAND

The RETURN command releases files assigned to a job and may release file space depending on the file type.

Format:

```
RETURN, lfn1, lfn2, ..., lfnn.
```

or

```
RETURN, *, lfn1, lfn2, ..., lfnn.
```

The first format returns the named files (lfn₁,lfn₂,...,lfn_n). The second format returns all files assigned to the job except the named files. If no files are named on the second format, the asterisk returns all files assigned to the job.

There are several exceptions. The RETURN command does not release any files that have the no-auto-drop status. Certain system scratch files (for example, ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD) have this status. You can give any local file this status by using the SETFS command.

RETURN performs the following operations according to the file type.

<u>Type</u>	<u>Operation</u>
Input	File name is changed to INPUT*. File space is not released (refer to Input File Control in section 3 for further information).
Queued	File space is released, and the file is no longer assigned to the job. The file also loses its queued status.
Local	File space is released, and the file is no longer assigned to the job.
Primary	Same as Local.
Library	File space remains, but the file is no longer assigned to the job.
Direct access	File space remains, but the file is no longer attached to the job.
Tape	Tape is no longer assigned to the job. Refer to the REWIND or UNLOAD command for a description of the processing that occurs if the previous operation was a write.

In addition, the return of a magnetic tape file or the return of your last direct access file on an auxiliary removable disk pack decrements the resource demand count as scheduled by the RESOURC command if, and only if, the total concurrent resource demand (tapes and removable packs) is presently assigned.

To release a file without decrementing the resource demand count, you can issue an UNLOAD command. To release file space without releasing the file from the job, you can issue an EVICT command.

REWIND COMMAND

The REWIND command rewinds files. A mass storage file is positioned at its BOI. A tape file is positioned at BOI (refer to appendix G). If the labeled tape file begins on a previous volume, the system notifies the operator to mount that volume.

Format:

REWIND,lfn₁,lfn₂,...,lfn_n.

or

REWIND*,lfn₁,lfn₂,...,lfn_n.

The first format rewinds the named files (lfn₁,lfn₂,...,lfn_n). The second format rewinds all files assigned to the job except the named files. If no files are named on the second format, the asterisk specification rewinds all files assigned to the job.

If the previous operation on the magnetic tape file was a write, a REWIND command causes the following operations to be performed.

- If the tape is ANSI labeled, the system writes a tape mark, an EOF1 label, and three tape marks and then rewinds the tape.
- If the tape is unlabeled and the data format specified on the ASSIGN, LABEL, or REQUEST command is S, L, or F, the system writes four tape marks and then rewinds the tape.
- If the tape is unlabeled and the data format is I or SI, the system writes a tape mark, an EOF1 label, and three tape marks and then rewinds the tape.

Refer to Magnetic Tape Files in section 2 and to section 12, Tape Management, for further information about tape files, and to appendix G for a description of EOF1 labels.

ROUTE COMMAND

The ROUTE command queues a file for disposal. The file routing may take effect when the command is processed, or it may be deferred. If deferred, the routing characteristics specified define the handling of the file in later job steps or at job termination. This command can also rescind a prior deferred ROUTE command.

PARAMETER DESCRIPTIONS

Format:

ROUTE,lfn,p₁,p₂,...,p_n.

lfn Specifies the file to route. File lfn must have a queued file type or a local file type; it cannot be a primary or direct access file. The lfn parameter is required on all ROUTE commands.

The remaining parameters are order-independent.

DC=dc Specifies the disposition of the queued file.

<u>dc</u>	<u>Meaning</u>
IN	Queue the file with an input disposition. Normal job input file format is required. If the job command within the file is in error, the file is not queued and remains a local file.
NO	Queue the file with an input disposition. Output not explicitly routed by the job is discarded at job completion. Normal job input file format is required. If the Job command within the file is in error, the file is not queued and remains a local file.

Description

<u>dc</u>	<u>Meaning</u>
TO	Queue the file with an input disposition and queue its output with a wait disposition unless the job routes it elsewhere. Normal job input file format is required. If the Job command within the file is in error, the file is not queued and remains a local file.
WT	Queue the file with a wait disposition.
LP	Queue the file for printing on any printer.
PR	Same as LP.
LR	Queue the file for printing on 580-12 printer.
LS	Queue the file for printing on 580-16 printer.
LT	Queue the file for printing on 580-20 printer.
SB	Queue the file for punching system binary format.
PB	Same as SB.
P8	Queue the file for punching in 80-column binary format.
PU	Queue the file for punching in coded format.
PH	Same as PU.
PL	Queue the file for plotting.
SC	Rescind prior routing and change the file type to local. If no prior routing exists, SC is ignored.

If the DC=dc parameter is omitted and lfn has been previously given a deferred routing, the file is queued with the disposition code previously specified. If the DC=dc parameter is omitted and lfn is not a deferred routed file, the file name may determine the default.

<u>If DC is omitted and lfn is:</u>	<u>ROUTE assumes DC=dc is:</u>
OUTPUT	DC=LP
PUNCH	DC=PU
PUNCHB	DC=SB
P8	DC=P8

<u>Pi</u>	<u>Description</u>
DEF	Indicates that queueing of the file is deferred to a later job step (OUT command or ROUTE command with the SC parameter specified) or end of job. If this parameter is specified, the file is created if it does not exist. DEF is not allowed if DC=IN or DC=TO.

EC=ec Defines external characteristics for files queued for printing or punching.

For files queued for printing, ec can be the following.

<u>ec</u>	<u>Meaning</u>
A4	Provided for NOS/BE compatibility. If you specify A4, the system uses the appropriate EC default (refer to following note).
A6	ASCII graphic 63/64-character set.
A9	ASCII graphic 95-character set. File lfn must be a 12-bit ASCII file. Refer to the FCOPY command earlier in this section.
B4	Provided for NOS/BE compatibility. If you specify B4, the system uses the appropriate EC default (refer to following note).
B6	CDC graphic 63/64-character set.

For files queued for punching, ec can be the following.

<u>ec</u>	<u>Meaning</u>
ASCII	Punch ASCII.
026 or 026	Punch 026 mode.
029 or 029	Punch 029 mode.
SB	Punch system binary.
80COL	Punch 80-column binary.

For files queued for plotting, ec can have the following values:

<u>ec</u>	<u>Meaning</u>
T6	Specifies 6-bit, transparent plot data.
T8	Specifies 8-bit, transparent plot data.

NOTE

If an invalid external characteristic is specified, the queued file processor cannot output the file. You must not specify a printing characteristic for a queued punching file or a punching file characteristic for a print file queued for printing. You also must not specify an external characteristic not available at the site. If EC is not specified, an appropriate EC default is set on the basis of the DC parameter setting and installation options.

- FC=fc Specifies routing to the output device that the system operator assigned the forms code fc. This parameter prevents output of a file before its special forms are placed in the output device. fc can be any two alphanumeric characters. A value of null results when no FC=fc parameter is specified.
- FID=fid Specifies a NOS/BE parameter, included for compatibility. It produces an informative message under NOS.
- FM=fm Indicates routing to a remote batch or interactive terminal logged in with the family name specified. The one- to seven-character family name must be alphanumeric. If you specify only the keyword FM, implicit remote routing results (refer to Implicit Routing following the last parameter description).
- IC=ic Specifies internal characteristics of the file; the value ic is one of the following:

<u>ic</u>	<u>Meaning</u>
DIS	Display code.
ASCII	ASCII code.
BIN	Binary.

Pi

Description

This parameter is normally not specified since its default is automatically established through the disposition code (DC=dc) and the external characteristic (EC=ec) parameters. The following lists show the default for ic for the various ec values.

<u>ec for printing</u>	<u>Default ic</u>
A4	ASCII
B4	DISPLAY
B6	DISPLAY
A6	DISPLAY
A9	ASCII

<u>ec for punching</u>	<u>Default ic</u>
SB	BINARY
80COL	BINARY
026	DISPLAY
029	DISPLAY
ASCII	ASCII

- ID=id Selects the local device with identifier id. Identifier id can be 0 to 67B (octal default). If you specify only the keyword ID, implicit remote routing results (refer to Implicit Routing following the last parameter description).
- PRI=pri File priority. This is a NOS/BE parameter included for compatibility. It produces an informative message under NOS.
- REP=rep Specifies the number of additional file copies to be routed to a destination. The range for rep is from 0 to 31; therefore, the number of copies that can be sent ranges from 1 to 32. Values for rep beyond its range are set to zero, an informative message is sent, and one copy is routed to the destination. The default value is zero; only one copy is routed.
- SC=sc Specifies a spacing code, specifying a programmable format control (PFC) array for the 580 PFC printer. The system is released with two PFC arrays, a default (SC=0), and an alternate (SC=1). The installation can define other PFC arrays. You can enter any spacing code from 0 to 77₈ (octal default), but if an array is not defined for that code, the default array (SC=0) is used. For more information on spacing codes, refer to the NOS 2 System Maintenance Reference Manual.
- TID=tid Indicates implicit remote routing (refer to Implicit Routing following the last parameter description), except for TID=C. TID=C specifies routing to the central site. In all other cases, the system ignores tid (terminal identifier). The TID=tid form of the parameter provides compatibility with NOS/BE.

<u>pi</u>	<u>Description</u>
UJN=ujn	Specifies a user job name (UJN) for the file to be routed. With one exception, the default is the UJN of the job from which you enter the command. In the case where you specify an input disposition (DC=IN, DC=NO, or DC=TO), the Job command of the queued file determines the UJN.
UN=un	Specifies the user name of the remote batch or interactive user to whom the system routes the file. If you specify only the keyword UN, implicit remote routing occurs (refer to Implicit Routing following the last parameter description).

IMPLICIT ROUTING

For remote batch origin jobs, the following action is taken.

- Specifying an ID, ID=id, or TID=C parameter causes routing to the central site.
- Specifying an FM=fm or UN=un parameter with valid arguments causes routing to the specified remote batch or interactive terminal.
- In all other cases, the system routes the file to the originating remote batch terminal.

For jobs of any origin other than remote batch, the following action is taken.

- Specifying an ID, ID=id, or TID=C parameter causes routing to the central site.
- Specifying UN, TID, or FM causes routing to the terminal specified by the job's family name and user name at the time of the processing of the ROUTE command.
- Specifying UN=un or FM=fm with valid arguments causes routing to the specified remote batch or interactive terminal.

If a ROUTE command specifies an input disposition for the file (DC=IN, DC=NO, or DC=TO) and the file has an invalid USER command, the job containing the ROUTE command aborts with no exit processing (a terminal is logged off). The security count for the user name that did the ROUTE is decremented accordingly.

SCOPY COMMAND

The SCOPY command copies a specified number of coded files from one file to another and, unless explicitly prohibited, indicates the structure of the file being copied by displaying the EORs and EOFs in the receiving file. A coded file is defined as a file containing lines of 160 characters or less, each terminated by a zero byte (12 zero bits in the lowest byte of a word).

NOTE

The SCOPY command is not recommended for use with S, L, or F format tapes because it does not have the data specification parameters needed to accommodate the variety of data formats possible with those tape formats. For an S, L, or F format tape copying operation, issue a COPY command with the appropriate parameter specifications.

The SCOPY command cannot copy SI format tapes. If coded mode is set for an SI tape, the system terminates the job. The TCOPI utility converts SI coded tape files.

When you copy a file with no EOR or EOF file marks, the resulting file may have a different logical file structure. In particular, when copying such a file to tape, the system adds an EOR mark to the end of the file. Such changes do not affect file content but may cause verification errors.

Format:

SCOPY, lfn₁, lfn₂, n, fchar, lchar, na, R, fcs, fline, lline, ns.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the file to copied. You must specify this parameter.
lfn ₂	Specifies the name of the file onto which lfn ₁ is copied; if this parameter is omitted, file OUTPUT is assumed.
n	Specifies the number of files (decimal) to copy; if this parameter is omitted, 1 is assumed.

<u>Parameter</u>	<u>Description</u>
fchar	Specifies the first character position (1 to 160 for 6-bit characters) [†] of each line to copy; if this parameter is omitted, the copy begins at character position 1.
lchar	Specifies the last character position (1 to 160 for 6-bit characters) [†] of each line to copy; lchar must be greater than or equal to fchar. If this parameter is omitted, the copy ends at character position 136. ^{††}
na	Specifies that the job step should not abort when a line terminator does not appear before an EOR. na can be any string of one to seven alphanumeric characters.
R	Specifies that both lfn ₁ and lfn ₂ be rewound before copying.
fcs	Specifies the character set code of lfn ₁ . The value 0 indicates display code or 6/12 display code. 0 is also the default. No other values are allowed.
fline	Specifies the line number of the first line you want to copy from lfn ₁ if lfn ₁ is a sequenced file. If you do not specify this parameter or if lfn ₁ is not a sequenced file (no line numbers), the copying begins with the first line of lfn ₁ .
lline	Specifies the line number of the last line you want to copy from lfn ₁ if lfn ₁ is a sequenced file. If you do not specify this parameter or if lfn ₁ is not a sequenced file (no line numbers), the copying is controlled by parameter n.
ns	Specifies no structure reporting. If you specify a nonnull value, the system does not display the EORs and EOFs in lfn ₂ .

If EOI is encountered before the file count is satisfied, an EOF is written on lfn₂, and the operation terminates. If a line is encountered that has more than lchar characters, the excess characters are truncated.

[†]Since many characters in 6/12 display code require 12 bits rather than 6, this parameter may produce unforeseen results when copying a file containing 6/12 display code data.

^{††}Refer to Appendix A for a description of character set codes.

SCOPY writes lines with an even number of characters. If an input line has an odd character count and the last character is a blank not immediately preceded by a colon, the last character is removed. If an input line has an odd character count and the last character is not a blank or is a blank immediately preceded by a colon, an additional trailing blank is appended.

If SCOPY attempts to copy a line longer than 160, 6-bit characters, the line is truncated, and an informative message is issued to the dayfile after the copy completes.

If the last line of a record does not have an end-of-line terminator, SCOPY issues a dayfile message. If the na parameter is not specified, the job step then aborts.

Example:

```
old,filea
  READY.
list
```

```
RECORD 1
AAA
BBB
CCC
RECORD 2
DDD
EEE
FFF
RECORD 3
GGG
HHH
III
```

```
  READY.
scopy,filea
```

```
RECORD 1
AAA
BBB
CCC
-EOR-
RECORD 2
DDD
EEE
FFF
-EOR-
RECORD 3
GGG
HHH
III
-EOR-
-EOF-
```

```
  READY.
```

SETFS COMMAND

The SETFS command sets the auto-drop or no-auto-drop status on files assigned to your job. If given the no-auto-drop status, a file is not released from your job by commands like NEW, OLD, CLEAR, RETURN(*), and UNLOAD(*) .

Format:

```
SETFS, lfn1, lfn2, ..., lfnn / FS = fs.
```

or

```
SETFS *, lfn1, lfn2, ..., lfnn / FS = fs.
```

<u>Parameter</u>	<u>Description</u>
lfn _i	Specifies the local file to be affected.
FS=fs	Specifies either the auto-drop (FS=AD) or no-auto-drop (FS=NAD) status. The default status for user-created files is auto-drop. Certain system files (for example, ZZZZCO, ZZZZC1, ZZZZC2, and ZZZZLD) always have the no-auto-drop status.

The first format of the command gives the specified status to the files lfn_i. The second format gives the specified status to all files assigned to the job, except files lfn_i.

The CLEAR command never releases a file that has a no-auto-drop status, the RETURN and UNLOAD commands release such a file only if you specify it, and the NEW or OLD command releases such a file only if it has the same name as the new primary file.

SKIPEI COMMAND

The SKIPEI command directs the system to position the specified file at the EOI.

Format:

```
SKIPEI, lfn.
```

lfn Specifies the name of the file to be positioned.

On magnetic tapes where no EOI is defined, the operation stops at an EOF.

The SKIPEI command has no effect on a primary file since the file is rewound before every operation.

SKIPF COMMAND

The SKIPF command directs the system to bypass, in a forward direction, the specified number of files from the current position of the named file.

Format:

SKIPF, lfn, n, m.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the file to be positioned.
n	Specifies the number (decimal) of files to be skipped; if the parameter is omitted, the system assumes 1.
m	Specifies the file mode: C for coded, B for binary. If m is omitted, the system assumes the file is in binary mode. If coded mode is set on an SI tape, the system aborts the job.

If an EOI is encountered before n files are bypassed, file lfn remains positioned at the EOI.

The SKIPF command has no effect on a primary file since the file is rewound before every operation.

SKIPFB COMMAND

The SKIPFB command directs the system to bypass, in the reverse direction, the specified number of files from the current position of the named file.

Format:

SKIPFB, lfn, n, m.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the file to be positioned.
n	Specifies the number (decimal) of files to be skipped; if the parameter is omitted, the system assumes 1.
m	Specifies the file mode: C for coded, B for binary. If m is omitted, the system assumes the file is in binary mode. If coded mode is set on an SI tape, the system aborts the job.

If the system encounters BOI (or the load point of the current volume of a multivolume tape file), SKIPF command processing ends.

The SKIPFB command has no effect on a primary file since the file is rewound before every operation.

SKIPR COMMAND

The SKIPR command directs the system to bypass, in a forward direction, the specified number of logical record or file marks from the current position of the named file.

Format:

SKIPR,lfn,n,level,m.

<u>Parameter</u>	<u>Description</u>
lfm	Specifies the name of the file to be positioned.
n	Specifies the number (decimal) of record and file marks to be skipped; if this parameter is omitted, the system assumes 1.
level	Specifies the level number; $0 < \text{level} < 17$. If $0 < \text{level} < 16$, both EOR and EOF marks are counted. If level is 17, only EOF marks are counted.
m	Specifies the file mode: C for coded, B for binary. If m is omitted, the system assumes the file is in binary mode. If coded mode is set on an SI tape, the system aborts the job.

Consecutive EOR or EOF marks are counted separately. If the EOI is encountered before n EOR and EOF marks are bypassed, file lfn remains positioned at the EOI.

The SKIPR command has no effect on a primary file since the file is rewound before every operation.

SORT COMMAND

The SORT command sorts a file of lines in numerical order based on leading line numbers consisting of a specified number of digits.

Format:

SORT,lfm,NC=n.

<u>Parameter</u>	<u>Description</u>
lfm	Specifies the file to be sorted. lfm may be a local file or a direct access permanent file assigned to your job.
NC=n	Specifies the number of leading line number digits to use in sorting the file. $n \leq 10$. The default is 5.

In the case of duplicate line numbers, all lines other than the first are considered correction lines. All lines with the same number are deleted from the file except the last line encountered.

For input from an interactive terminal, SORT deletes a line if a line number is followed by an empty line or a line number is followed only by a blank. However, the system retains a line with more than one blank after the line number.

For input from cards, SORT deletes a line if a card containing only the line number is submitted.

If a line number contains more than n digits, you can delete the line either by entering the first n digits of the line number and pressing the carriage return (terminal input) or by submitting a card containing only the first n digits of the line number (batch input).

After the sort, lfn is packed and set at EOI.

TCOPY COMMAND

The TCOPY command copies X (external) format binary tapes or E (line image), B (blocked), or SI (system internal) format coded tapes to mass storage, to an I format tape, or to an SI binary format tape. It also writes E or B format tapes converted from files on mass storage, I format tape, or SI format binary tape. The X binary and E, B, and SI coded tape formats were supported under earlier versions of NOS. Refer to appendix J for the tape formats of B, E, and X format coded tapes. Now, to access data or write data in one of these formats, the tape must be assigned as an S (stranger) format tape (refer to the tape assignment commands in section 12) and the file copied using the TCOPY command.

Order-dependent format:

TCOPY, lfn₁, lfn₂, format, tc, copycnt, charcnt, erlimit, p₁p₂, lfn₃.

Order-independent format:

TCOPY, I=lfn₁, O=lfn₂, F=format, TC=tc, N=copycnt, CC=charcnt, EL=erlimit, PO=p₁p₂, L=lfn₃.

The parameters on the TCOPY command can appear in order-dependent format, order-independent format, or a combination of both.

If order-dependent and order-independent parameters are mixed in one TCOPY command, the order-dependent parameters must appear in their proper position. All parameters are optional. However, the specification of certain parameters precludes the application of others. A nonapplicable parameter may be ignored or may generate an error. This is stated in the individual descriptions of the parameters.

The parameters are defined as follows:

<u>Parameter</u>	<u>Description</u>	<u>Default</u>				
I=lfn ₁	Specifies the file to copy.	INPUT				
O=lfn ₂	Specifies the file to receive the copy.	OUTPUT				
F=format	Data format that specifies the type of conversion for the copying operation. This can be any one of the following.	X				
	<table border="1"> <thead> <tr> <th><u>format</u></th> <th><u>Conversion</u></th> </tr> </thead> <tbody> <tr> <td>E</td> <td>Copy an E format tape to mass storage, an I, or an SI binary tape file, or generate a new E format tape from mass storage, an I, or an SI binary tape file. The E tape must be unlabeled and assigned as S format.</td> </tr> </tbody> </table>	<u>format</u>	<u>Conversion</u>	E	Copy an E format tape to mass storage, an I, or an SI binary tape file, or generate a new E format tape from mass storage, an I, or an SI binary tape file. The E tape must be unlabeled and assigned as S format.	
<u>format</u>	<u>Conversion</u>					
E	Copy an E format tape to mass storage, an I, or an SI binary tape file, or generate a new E format tape from mass storage, an I, or an SI binary tape file. The E tape must be unlabeled and assigned as S format.					

<u>Parameter</u>	<u>Description</u>	<u>Default</u>
	<u>format</u> <u>Conversion</u>	
	B Copy a B format tape to mass storage, an I, or an SI binary tape file, or generate a new B tape from mass storage, an I, or an SI binary tape file. The B tape must be unlabeled and assigned as S format.	
	X Copy an X format tape to mass storage, an I, or an SI binary tape file. The unlabeled input tape must be assigned an S format, with noise size of eight frames for seven-track tape or six frames for nine-track tape (refer to NS parameter on tape assignment command).	
	SI Copy an SI coded format tape to mass storage, an I, or an SI binary tape file. The labeled or unlabeled input tape must be assigned as S format, with noise size of eight frames for seven-track tape or six frames for nine-track tape (refer to NS parameter on tape assignment command). If the SI coded input tape is completed before EOI is encountered; the position of the input tape after the copying is indeterminate. This is because control words are used on the SI coded tape read via S format (EOF on an SI coded tape is a level 17g block terminator, whereas EOF on an S tape is a tape mark).	
TC=tc	Specifies the termination condition for copying used in conjunction with N=copycnt. The termination condition can be specified as follows:	Copy to double EOF (TC=D or TC=EOD)
	F or EOF When this TC=tc value is set, the N keyword specifies the number of files to copy.	
	I or EOI This specifies copying to the end of information. The N keyword is ignored.	
	D or EOD When this TC=tc value is set, the N keyword is the number of double EOFs to which to copy.	
N=copycnt	Specifies the copy count used by the termination condition parameter TC=tc.	1

<u>Parameter</u>	<u>Description</u>	<u>Default</u>
CC=charcnt	Specifies the character count which determines maximum block size (line length) in characters for an E or B format tape. This parameter can only be specified for copying an E or B format tape.	136 characters for E format; 150 characters for B format.
EL=erlimit	Specifies the error limit which sets the number of nonfatal errors allowed before abort. This includes parity errors and block-too-large errors which are returned by the tape subsystem after completing recovery procedures. It also includes illegal block format errors (invalid byte-count coded format and/or unused bit count) for X format and SI coded format tapes. Error limit is ignored when generating an E or B format tape from mass storage, an I format, and an SI binary format file since control word read is not used. Error limit is likewise ignored if the input file device does not support control word read (terminals). In that case, any error aborts the job.	Zero
PO=p ₁ p ₂	Specifies one or both of the following processing options (not separated by commas).	
	E Input blocks with parity errors or block-too-large errors are processed (copied).	Error blocks are skipped.
	T When generating a B or E format tape, blocks exceeding the maximum block size (refer to the CC=charcnt parameter) are truncated. PO=T is not allowed for other file conversions.	Lines exceeding the maximum line size are split into multiple blocks.
L=lfn ₃	Specifies the name of an alternate output file to receive parity error messages when extended error processing is in effect (nonzero EL specified), in which case, the file name lfn ₃ must not be the same as lfn ₂ .	OUTPUT

Example:

The following TCOPY command combines order-dependent and order-independent parameters:

```
TCOPY,TAPE1,FILE2,E,CC=200,EL=12.
```

The input file TAPE1 is an E format tape (assigned as an S format tape). It has a maximum of 200 characters per line. The copying terminates when a double EOF is encountered (default). The output file FILE2 can be a mass storage file or an I or SI binary format tape. The error limit allows up to 12 nonfatal errors (parity/block-too-large), and the bad data is skipped (default) with informative error messages written to the file OUTPUT (default).

The TCOPY command begins a copying operation at the current position of both files and continues until the copy termination condition for copying is met or EOI is encountered. This termination condition can be a file count, double EOF count, or EOI. If the copying is terminated by a double EOF (for TC=EOD parameter), the second EOF is detected on lfn₁ but is not transferred to lfn₂. If lfn₁=lfn₂, the named file is read until the termination condition is satisfied or EOI is encountered. An SI coded tape can be positioned correctly only to EOI (refer to the F=SI parameter description).

If you specify a file count TC=EOF, and EOI is encountered on the input file before the file count is satisfied, an additional EOF is written on the output file only if data or records have been transferred since the previous EOF was written (or since the beginning of the copying, if no EOFs have been encountered).

The EL=erlimit or PO=E parameters provide extended error processing. This allows the processing or skipping of blocks with parity errors or block-too-large errors. The block count for the first block to be copied is initially set to zero and is incremented by one for every block and every EOF processed.

When creating a B format tape from a mass storage, I, or SI binary format file, a block shorter than the noise size specified on the tape assignment statement is blank filled to the noise size. A noise block could also be generated when a block exceeding the maximum block size for the B format tape is split into multiple blocks. If the PO=T parameter is specified, blocks exceeding the maximum block size are truncated.

When creating an E format tape from a mass storage, I, or SI binary format file, blocks that exceed the maximum block size for the E format tape are split into multiple blocks. If a continuation block contains only the end-of-line indicator (zero word), the continuation block is discarded. If the PO=T parameter is specified, blocks exceeding the maximum block size are truncated (all continuation blocks are discarded).

TDUMP COMMAND

The TDUMP command lists a file in octal or alphanumeric format. It dumps the entire file or the specified number of lines, records, or files. If more than one limit is set, the limit reached first overrides the others. If you are in ASCII mode (refer to the ASCII command in section 8) and the output file is assigned to your terminal, TDUMP interprets input and displays alphanumeric output as 6/12 display code.

NOTE

TDUMP produces unpredictable results when dumping an S, L, or F format tape file. Use the COPY command to convert the S, L, or F format tape file to a mass storage file or to an I or SI binary format tape file before attempting to dump the file using TDUMP.

Format:

TDUMP,P₁,P₂,...,P_n.

<u>P_i</u>	<u>Description</u>
I=lf _n ₁	Specifies the local file to be dumped (default is TAPE1).
L=lf _n ₂	Specifies the local file to which the output is written (default is OUTPUT). If lf _n ₂ is not a local file, TDUMP creates it. It does not rewind lf _n ₂ before or after the dump.
O	Specifies an octal dump only.
A	Specifies an alphanumeric dump only.
	If both O and A are specified, the last one overrides. If neither O nor A is specified, TDUMP lists both an octal and an alphanumeric dump.
R=r _{count}	Specifies the maximum decimal number of records to be dumped. If R is omitted or set to zero, the dump continues to EOI.

NOTE

The record count restarts at each EOF.

F=f _{count}	Specifies the maximum decimal number of files to be dumped. If F is omitted, the dump continues to EOI. If F=0, dump continues until an empty file (double EOF) or EOI is encountered.
N=lines	Specifies the maximum decimal number of lines to be dumped. If N is omitted or set to zero, the dump continues to EOI. The blank line output with the end-of-record, end-of-file, end-of-information, and ABOVE LINE REPEATED messages is included in the line count.
NR	Do not rewind file lf _n ₁ before dump (default is to rewind lf _n ₁).

Example:

Two lines, each containing the alphabet, were input to file X from an interactive terminal. File X was dumped to file Y producing the following output.

```
- FILE DUMP - TDUMP,I=X,L=Y. yy/mm/dd. 08.05.51. PAGE 1.
F 1 R 1 W 0- 0102 0304 0506 0710 1112 1314 1516 1720 2122 2324 2526 2730 3132 0000 0000 0102 0304 0506 0710 1112
          A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
F 1 R 1 W 4- 1314 1516 1720 2122 2324 2526 2730 3132 0000 0000
          K L M N O P Q R S T U V W X Y Z
-- END OF RECORD --
-- END OF INFORMATION --
-- END OF DUMP --
```

The prefix

F 1 R 1 W 0

means file 1, record 1, word 0. The zeros following each alphabet are the end-of-line indicators.

UNLOAD COMMAND

The UNLOAD command releases files assigned to the job and may release file space (depending on the file type).

Format:

UNLOAD, lfn₁, lfn₂, ..., lfn_n.

or

UNLOAD, *, lfn₁, lfn₂, ..., lfn_n.

The first format unloads the named files (lfn₁, lfn₂, ..., lfn_n). The second format unloads all files assigned to the job except the named files. If no files are named on the second format, the asterisk specification unloads all files assigned to the job. There are some exceptions. The UNLOAD command does not release any files that have the no-auto-drop status. Certain system scratch files (for example, ZZZZCO, ZZZZC1, ZZZZC2, and ZZZZLD) have this status. You can give any local file this status by using the SETFS command.

The UNLOAD command performs the same function as the RETURN command except as noted below. Refer to the description of the RETURN command given earlier in this section to determine the operation performed for each file type.

If the file being unloaded is a magnetic tape file or a direct access file residing on an auxiliary removable pack and the job requires more than one tape or pack concurrently, the UNLOAD command differs from the RETURN command. First, it does not decrease the number of tapes or packs that were scheduled for the job with the RESOURC command. Second, if the file is a magnetic tape file card and if the previous operation was a write, the following operations are performed:

- If the tape is ANSI labeled, the system writes a tape mark, an EOF1 label, and three tape marks and then unloads the tape.
- If the tape is unlabeled and the data format specified on the ASSIGN, LABEL, or REQUEST command is S, L, or F, the system writes four tape marks and then unloads the tape.
- If the tape is unlabeled and the data format is I or SI, the system writes a tape mark, an EOF1 label, and three tape marks and then unloads the tape.

Refer to Magnetic Tape Files in section 2 and Tape Management commands in section 12 for further information about tape files, and to appendix G for a description of an EOF1 label.

UNLOCK COMMAND

The UNLOCK command rescinds the LOCK command and clears the write interlock for the specified file. This command can only be used on local mass storage files.

Format:

UNLOCK, lfn₁, lfn₂, ..., lfn_n.

lfn_i Specifies the name of a local file.

Library files cannot be unlocked.

VERIFY COMMAND

The VERIFY command performs a binary comparison of all data from the current position of the files specified. The comparison is meaningful if the files are within the range of compatible formats listed in table 9-2.

Format:

VERIFY, lfn₁, lfn₂, p₁, p₂, ..., p_n.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of the first file; if this parameter is omitted, TAPE1 is assumed.
lfn ₂	Specifies the name of the second file; if this parameter is omitted, TAPE2 is assumed.
<u>P_i</u>	<u>Optional parameters</u>
	<u>pi</u> <u>Description</u>
N=0	Terminate verification on the first empty file encountered on either file.
N=x	Verify x files; default is N=1.
N	Terminate verification when end of information is encountered on both files.
E=y	List the first y errors encountered on the comparison. If E is omitted, the system assumes E=100.
E	Same as E, no errors are listed.
L=lfn ₃	List errors on file lfn ₃ . If L is omitted, the system assumes L=OUTPUT.
A	Abort after verification completed if errors occurred.
R	Rewind both files before and after the verification.

<u>Pi</u>	<u>Description</u>
C	Set coded file mode on both files. This is applicable only to S and L format tapes. If coded mode is set on an SI tape, the system aborts the job.
C1	Set coded file mode on the first file only. This is applicable only to S and L format tapes. If coded mode is set on an SI tape, the system aborts the job.
C2	Set coded file mode on the second file only. This is applicable only to S and L format tapes. If coded mode is set on an SI tape, the system aborts the job.
BS=bsize	Defines the maximum block size (PRU size) in central memory words for an S or L tape. This parameter is legal only for the verification of S and L tape verifies. The default for an S tape is 1000 _g words, and for an L tape, it is 2000 _g words.

Whenever words on lfn₁ and lfn₂ do not match, the VERIFY command lists the following.

- Record number.
- Word number within the record.
- Words from both files that do not match.

In a verification involving S, L, or F format tapes, the VERIFY command first clears the extraneous data in the last word of each block (as specified by the byte count and the unused bit count) and then makes the comparison. On these formats, every block is considered a record (returns EOR status).

If a verification of an L or F format tape requires additional field length, the VERIFY command increases the field length as needed.

The maximum block size for an L format tape is specified by the BS=bsize or its default. The maximum block size for an F format tape is calculated from the frame or character count specified on the command when the file is assigned.

Table 9-2. Compatible File Structures for the VERIFY Command

		SECOND FILE (Ifn ₂)						
		Mass Storage	Tape Formats					
			I	SI	S	L	F	
Mass Storage		Yes	Yes	Yes	No	No	No	
FIRST FILE (Ifn ₁)	Tape Format	I	Yes	Yes	Yes	No	No	No
		SI	Yes	Yes	Yes	No	No	No
		S	No	No	No	Yes	No	No
		L	No	No	No	No	Yes	No
		F	No	No	No	No	No	Yes

NOTE

The No entries indicate that the logical structures of the files compared are incompatible. The VERIFY command may accept those combinations, but the results require you to make a knowledgeable correlation of results with the format descriptions in section 12. In some cases, the verification of an incompatible pair may result in a VERIFY GOOD message; otherwise, a VERIFY ERRORS message is listed.

WRITEF COMMAND

The WRITEF command directs the system to write a specified number of file marks on the named file.

Format:

WRITEF, lfn, x.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the file to be written on.
x	Specifies the number of filemarks to be written. If this parameter is omitted, the system assumes x is 1.

If the last operation to the file was a write that did not end with the writing of an EOR or EOF, WRITEF writes a record mark before it writes the specified number of file marks. For all other cases, WRITEF writes the file marks without a preceding record mark.

WRITER COMMAND

The WRITER command directs the system to write a specified number of empty records on the named file.

Format:

WRITER, lfn, x.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the file to receive the empty records.
x	Specifies the number of empty records to be written. If this parameter is omitted, the system assumes x is 1.

With permanent file commands, you can create, access, and purge permanent files. You can also control other users' access to your files.

The following are the permanent file commands and their functions.

<u>Command</u>	<u>Function</u>
APPEND	Appends data to an indirect access permanent file.
ATTACH	Assigns a direct access permanent file to a job.
CATLIST	Lists permanent file information.
CHANGE	Changes permanent file characteristics.
COMMON	Accesses library type files.
DEFINE	Creates a direct access permanent file.
GET	Retrieves a copy of an indirect access permanent file as a local file.
OLD	Retrieves a copy of an indirect access permanent file as the primary file.
PACKNAM	Specifies an auxiliary device from which permanent files are to be accessed.
PERMIT	Grants permanent file access permission to other users.
PURGALL	Purges all files having the specified characteristics.
PURGE	Purges the named files.
REPLACE	Copies a local file over an indirect access permanent file.
SAVE	Creates an indirect access permanent file.

If an error occurs in an operation on one file of a multifile request, the operation is not performed on the subsequent files. For example, if an error occurs in the processing of file B on the following command:

```
GET,A,B,C,D.
```

files C and D are not processed.

For more information on permanent files, refer to Permanent Files, Mass Storage Files, and Mass Storage File Residence in section 2.

To determine the meaning of a permanent file error message, refer to appendix B.

COMMON PARAMETERS

Some permanent file commands can process several files. Each file is identified by a one-to seven-character file name. The permanent file operation is performed on each file in the sequence specified on the command. If the NA parameter is not specified on the command, and if an error occurs while one of the files is being processed, the job step terminates.

When this happens, the files preceding the file in error are processed, but the files succeeding it are not. If the NA parameter is specified on the command, the job step does not terminate as a result of the error, and all files are processed except the file in error.

A slash separates the file names from other parameters. The parameters specified after the slash are optional and order independent. Parameters specified after the slash affect all files named before the slash.

Permanent file access is controlled by the access category (CT), user name (UN), file access password (PW), and access mode (M) parameters. Permanent file residence is determined by the preferred residence (PR), auxiliary device (PN), and device type (R) parameters. Processing options include the no abort (NA), wait-if-busy (WB), subsystem selection (SS), and backup copy (BR) parameters.

In the following list, parameters common to more than one permanent file command are described in alphabetic order. For the format of each command, refer to the command descriptions following the parameter descriptions.

<u>Parameter</u>	<u>Description</u>								
BR=br	Specifies the backup copy requirement. You can request that a tape backup copy, an MSF backup copy, or no backup copy be kept of the permanent file. If BR=br is omitted when the file is created, a backup copy of the file is stored on tape. For more information, refer to Mass Storage Facility in section 2.								
	<table><thead><tr><th><u>br</u></th><th><u>Backup Requirement</u></th></tr></thead><tbody><tr><td>Y</td><td>A tape backup copy should be kept, even if a copy of the file exists on MSF.</td></tr><tr><td>MD</td><td>A tape backup copy need not be kept, if a copy of the file exists on MSF.</td></tr><tr><td>N</td><td>A backup copy need not be kept for this file.</td></tr></tbody></table>	<u>br</u>	<u>Backup Requirement</u>	Y	A tape backup copy should be kept, even if a copy of the file exists on MSF.	MD	A tape backup copy need not be kept, if a copy of the file exists on MSF.	N	A backup copy need not be kept for this file.
<u>br</u>	<u>Backup Requirement</u>								
Y	A tape backup copy should be kept, even if a copy of the file exists on MSF.								
MD	A tape backup copy need not be kept, if a copy of the file exists on MSF.								
N	A backup copy need not be kept for this file.								
CT=ct	Specifies the file access category. If CT=ct is omitted when the file is created, the file is private. The file category can be changed later.								

<u>Parameter</u>	<u>Description</u>	<u>Category</u>
		<u>ct</u>
P, PR, or PRIVATE	Private files; available for access only by their creator and by those granted explicit access permission by the file creator (refer to the PERMIT command).	
S or SPRIV	Semiprivate files; available for access by a user who knows the file name, user name, and password and who has not been explicitly denied permission to the file (M=NULL parameter on a PERMIT command). The system records the user name of each user who accessed the file, the number of accesses made, and the date and time of the last access by each user. The file creator can list this information with the CATLIST(LO=FP) command.	
PU or PUBLIC	Public files; available for access by all users who know the file name, user name, and password. The system records the number of times the file was accessed and the date and time of the last access, but does not record user names.	
NA	Specifies the no abort option. Normally, an error in command processing terminates the job. However, if NA is specified, the job does not terminate as a result of the error. The error is handled in one of two ways. <ul style="list-style-type: none"> • If the error condition is temporary, job processing is suspended until the error condition ends (for example, when a requested direct access file is busy or a requested auxiliary device is not available). In a multimainframe environment, job processing may be suspended until some period after the error condition ends. • If the error condition is not temporary, the job continues with the next operation. The job processes the next file listed on the command or, if no more files are listed, processes the next command. <p>You may not specify both the NA and WB parameters.</p>	
PN=packname	Specifies the one- to seven-character pack name used in conjunction with the R parameter to identify the 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.	

Parameter

Description

An auxiliary device is a mass storage device that supplements the normal family of permanent file devices. A RESOURC command must be included in any job that concurrently uses two or more auxiliary disk packs or an auxiliary pack and a magnetic tape.

If you specify PN=0 on a permanent file command, it overrides any previously issued PACKNAM commands. You can then access files on the permanent file family device. PN=0 does not affect subsequent commands, only the current command.

PR=pr

Specifies the preferred residence. If PR=pr is omitted when a direct access file is defined, no preference is assumed. For more information, refer to Mass Storage Facility in section 2.

pr

Preference

M You prefer that the direct access file be copied to MSF between accesses. However, the system might not copy the file to MSF despite the specified preference.

N You have no preference.

PW=password

Specifies the one- to seven-character file password. If a password is assigned to the file, users other than the file creator must specify the password when accessing the file.

If only the keyword PW is specified, you must include the password as a single-line record in the input file. In a interactive job, the password is requested by a ? prompt at the terminal.

R=r

Specifies the device type on which the file resides or is to reside.

r

Device

- DE Extended memory
- DIi 844-21 Disk Storage Subsystem (1<i>8) (half-track)
- DJi 844-41 or 844-44 Disk Storage Subsystem (1<i>8) (half-track)
- DKi 844-21 Disk Storage Subsystem (1<i>8) (full-track)
- DLi 844-41 or 844-44 Disk Storage Subsystem (1<i>8) (full-track)
- DMi 885 Disk Storage Subsystem (1<i>3) (half-track)
- DQi 885 Disk Storage Subsystem (1<i>3) (full-track)
- DP † Distributive data path to extended memory (not supported on models 825, 835, and 855)
- DV †† 819 Disk Storage Subsystem (single-density)
- DW †† 819 Disk Storage Subsystem (double-density)

†The job must be of system origin or you must have system origin privileges.

††Model 176 only.

Parameter

Description

The R parameter can be specified on any permanent file command to identify (along with the PN parameter or with a previous PACKNAM command) the auxiliary device on which the permanent file resides. R is specified only if the installation defines the auxiliary device as removable, and if the desired device type differs from the installation-defined default device type. R is not required if a previous PACKNAM command specified the R parameter. If the R parameter is specified without the PN parameter in a command, R is ignored on all commands except the DEFINE command (refer to the DEFINE command in this section).

SS=subsystem

Specifies the interactive subsystem to be associated with the file. The subsystem can be specified on a SAVE or CHANGE command with one of the following entries or its abbreviation (the abbreviation is the first three characters of the entry).

<u>Subsystem</u>	<u>Interactive Subsystem</u>
BASIC	BASIC
BATCH	Batch
EXECUTE	Execute
FORTRAN	FORTRAN 5
FTNTS	FORTRAN Extended 4
NULL	Null

If the SS keyword is specified without a subsystem, the currently active subsystem is associated with the file. If the SS keyword is omitted, no subsystem (null) is associated with the file, unless the file is the primary file. In that case, the currently active subsystem is associated with the file.

UN=username

Specifies the user name. This parameter must be specified if the requested permanent file is in another user's catalog. To access the file, you must have one of the following permissions.

- Explicit permission. The owner of the file granted access permission to the user with a PERMIT command.
- Implicit permission. The file is semiprivate or public and so can be accessed by users who know the file name, user name, and password and who have not been explicitly denied permission.
- Automatic permission. You have automatic permission to access files in the catalog of another user if your user name contains asterisks, and if all nonasterisk characters match characters in the other user's user name.

The UN keyword establishes alternate access validation, unless the specified user name is the one under which the job is currently being run. In this case, the specified user name is ignored.

WB

Specifies the wait-if-busy option. WB causes the job to wait for busy files and the mounting of disk packs but to abort on other error conditions. You may not specify both the NA and WB parameters.

APPEND COMMAND

The APPEND command adds information to the end of an existing indirect access file without retrieving the file. The APPEND command cannot append data to direct access files.

Format:

APPEND,pfn,lfn₁,lfn₂,...,lfn_n/UN=username PW=password,PN=packname,R=r,NA,WB.

<u>Parameter</u>	<u>Description</u>
pfn	Specifies the name of the indirect access permanent file to which data is appended. You do not retrieve the indirect access file before the append operation, but you must have permission to access the file in append, modify, or write mode. If a local copy of the indirect access file is assigned to the job, the append operation does not change the local copy.
lfn _i	Specifies the name of a local mass storage file to be appended to pfn. The file must be assigned to the job.

The full descriptions of the following optional parameters are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
UN=username	Specifies the user name. Specified if pfn is in another user's catalog.
PW=password	Specifies the file password. Specified if UN=username is specified and pfn has a password.
PN=packname	Specifies the auxiliary device name. Specified if pfn resides on an auxiliary device.
R=r	Specifies the device type. Specified if an auxiliary device on a device type other than the installation-defined default is to be used.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

The logical structure of the files is retained; that is, EORs and EOFs are appended as well as data. The append operation removes the EOI mark at the end of file pfn, but keeps all EOR and EOF marks.

ATTACH COMMAND

The ATTACH command assigns a direct access permanent file to a job.

Format:

ATTACH,lfn₁=pfn₁,lfn₂=pfn₂,...,lfn_n=pfn_n/M=m,UN=username,PW=password,
PN=packname,R=r,NA,RT,WB.

<u>Parameter</u>	<u>Description</u>
lfn _i =pfn _i	<p>Specifies the one- to seven-character file name lfn_i references direct access file pfn_i while the file is assigned to the job. If lfn_i= is omitted, the direct access file is referenced by its permanent file name, pfn_i. User access is directly to the permanent file; no working copy is generated.</p> <p>If a local file name lfn_i exists when this command is processed, it is discarded (even if command processing does not complete due to an error).</p>

The full descriptions of the following optional parameters (except M=m and RT) are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>		
M=m	File access mode requested. If M=m is omitted, M=READ is assumed.		
	<u>If m is:</u>	<u>You can:</u>	<u>Concurrently, other users can:</u>
	E or EXECUTE (execute mode)	Execute the file.	Read or execute the file.
	R or READ (read mode)	Read or execute the file.	Read or execute the file.
	RA or READAP (read append mode)	Read or execute the file.	Read or execute the file; one user can lengthen the file.
	RM or READMD (read modify mode)	Read or execute the file.	Read or execute the file; one user can lengthen or rewrite the file.
	A or APPEND (append mode)	Read, execute, or lengthen the file.	Read or execute the file.
	M or MODIFY (modify mode)	Read, execute, lengthen, or re-write the file.†	Read or execute the file.
	W or WRITE (write mode)	Read, execute, lengthen, rewrite, or shorten the file.	No access allowed.

You cannot change the file unless you attach it in append, modify, or write mode. Refer to table 10-1 for the access mode granted when the file is already attached by another user.

If the file is attached in append, modify, or write mode, the current date is recorded as the last modification date even if the file is not altered.

†You cannot rewrite the file using the copying commands described in section 9. To rewrite the file while in modify mode, you must write your own program using the REWRITE macro (refer to Volume 4, Program Interface).

<u>Parameter</u>	<u>Description</u>
UN=username	Specifies the user name. Specified if the permanent file(s) is in another user's catalog.
PW=password	Specifies the file password. Specified if UN=username is specified, and if the permanent file has a password.
PN=packname	Specifies the auxiliary device name. Specified if the permanent file(s) resides on an auxiliary device.
R=r	Specifies the device type. Specified if an auxiliary device on a device type other than the installation-defined default is used.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job; the job also waits for the mounting of disk packs and busy files.
RT	Specifies the real-time processing option. If RT is specified, the job continues processing after requesting staging of a direct access file from MSF to disk. If staging is not required (the file is already resident on disk), the file is assigned to the job. If the file must be staged, you must issue a second ATTACH command to assign the file to the job. If RT is omitted and the file resides only on the MSF, job processing is suspended while the MSF file is staged to disk and assigned to the job. For more information on the MSF, refer to Mass Storage Facility in section 2.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs and busy files.

If an auxiliary device has been previously specified by a PACKNAM command, the system attempts to find pfn_i on the auxiliary device rather than on a family device.

In the first column of table 10-1 is the access mode in which the first user attached a direct access file. The remaining columns are the access modes that may be requested by another user. At the intersection of the row and column is the access mode granted to the other user (assuming the user has the appropriate access permission). Busy means that the other user is sent a message that the file is busy and the job step is aborted unless NA is specified.

Assuming that no user has the file attached in append, modify, or write mode, 12285 users can attach the file concurrently (4095 in read mode, 4095 in read append mode, and 4095 in read modify mode). If a user has the file attached in append mode, 8190 other users can attach the file (4095 in read append mode and 4095 in read modify mode). If a user has the file attached in modify mode, 4095 other users can attach it in read modify mode. If a user has the file attached in write mode, no other user can attach the file.

Table 10-1. Access Mode Granted When Attaching a Currently Attached Direct Access File

Mode in Which First User Attached File	Access Mode Requested by Another User						
	Write	Modify	Append	Read	Read Modify	Read Append	Execute
Write	Busy	Busy	Busy	Busy	Busy	Busy	Busy
Modify	Busy	Busy	Busy	Busy	Read modify	Busy	Busy
Append	Busy	Busy	Busy	Busy	Read modify	Read append	Busy
Read	Busy	Busy	Busy	Read	Read modify	Read append	Execute
Read Modify	Busy	Modify	Append	Read	Read modify	Read append	Execute
Read Append	Busy	Busy	Append	Read	Read modify	Read append	Execute
Execute	Busy	Busy	Busy	Read	Read modify	Read append	Execute

CATLIST COMMAND

The CATLIST command lists information about your permanent files or the permanent files you can access in the catalogs of other users.

Format:

CATLIST, LO=p, FN=pfm, UN=username, PN=packname, R=r, DN=dn, NA, L=lfm, WB.

LO=p One of the following list options. If LO is omitted, LO=0 is assumed.

P

Description

F

Lists pertinent information about each file in your catalog. An example of this list option is given following the parameter descriptions.

If another user name is specified (UN=username), you receive a listing of all the files you can access in the other user's catalog. The passwords for files in another user's catalog are not listed. Passwords must be obtained from the user.

Parameter

Description

<u>P</u>	<u>Description</u>
FP	<p>Lists the access permissions granted for the file specified on the FN=pfm parameter. If a user name is also specified (UN=username), only the file permission granted to that user is listed.</p> <p>The user names listed include those that have been granted explicit permission to access the file (private files only) and those that have accessed the file through implicit permission (semiprivate files only). (User names are not recorded for accesses to public files.) An asterisk follows the user name/permission mode if explicit permission has been granted this user.</p>
0 (zero)	<p>Lists alphabetically the names of the indirect access files and direct access files in your catalog. If you specify a user name (UN=username), you receive a list of the files you can access in the other user's catalog.</p> <p>An asterisk preceding a file name indicates an error status is set in the catalog entry for the file. The cause of the error may be one of the following.</p> <ul style="list-style-type: none">● EOI was altered during mass storage recovery.● Error exists in file data and/or permit entries. <p>To clear an error status flag, refer to CHANGE Command in this section.</p>
P	<p>Lists only the user names of users who have access to the specified private file or who have accessed the specified semiprivate file. This option requires that a file name be specified (FN=pfm).</p>
FN=pfm	<p>Specifies the permanent file name. This parameter is required when listing access permission data (when LO=FP or LO=P is specified).</p> <p>If FN=pfm and LO=0 are specified, a message indicates whether or not the file is found in your catalog. If FN=pfm, LO=P, and UN=username are specified, a message indicates whether or not permission to access that file has been granted to that user name (username FOUND.).</p> <p>If pfm contains one or more asterisks, the CATLIST command lists catalog information for the subset of files whose names match except where the asterisks appear. For example, FN=***OPL lists all six-character file names ending in OPL. The asterisk is invalid when listing permit information with the LO=FP and LO=P list options.</p>

<u>Parameter</u>	<u>Description</u>
UN=username	<p>Specifies the user name. This parameter has two purposes.</p> <ul style="list-style-type: none"> • For list options LO=F and LO=0, it specifies the alternate catalog for which you desire catalog information. • For list options LO=FP and LO=P, it requests the permission information recorded for the specified alternate user. <p>When the short list options, LO=0 and LO=P, are specified, a message informs you whether or not the file or user name has been found.</p>
PN=packname	Specifies the auxiliary device name. This parameter specifies that CATLIST should use the catalog information on the named auxiliary device.
R=r	Specifies the device type on which the permanent file catalog resides. R=r is used with the PN and NA parameters (refer to the R parameter description at the beginning of this section).
DN=dn	Specifies the device number assigned to a device at system initialization. If specified, the CATLIST command lists catalog information from that device.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
L=lfm	<p>Specifies the output file name. lfm is the name of the file assigned to the job on which CATLIST information is written. If L=lfm is omitted, the system assumes L=OUTPUT.</p> <p>lfm is not rewound before or after the CATLIST operation.</p>
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

Example 1:

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

```

CATALOG OF BJK2201                FM/NOSCLSH yy/mm/dd. 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 2:

Listing of current files that begin with the letters PROC in the catalog of WAC3651. The command is entered by user WAC3651 in the form: CATLIST,FN=PROC***.

```
CATALOG OF WAC3651          FM/NOSCLSH yy/mm/dd. 13.27.02.
INDIRECT ACCESS FILE(S)
PROCART  PROCFIL  PROC1    PROC1A
      4 INDIRECT ACCESS FILE(S),  TOTAL PRUS =      15.
```

Example 3:

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 yy/mm/dd. 07.48.55.
      FILE NAME PRIME
      USER NUMBER(S)
CML2011  JLC2016  KXK4277
      3 USER(S)
```

Example 4:

Listing of current files in the catalog of SAH3523. The command is entered by user SAH3523 in the form: CATLIST,LO=F.

```
CATALOG OF SAH3523          FM/NOSCLSH yy/mm/dd. 13.49.09  PAGE 1
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 FJOB~      IND. PRIVATE      21      yy/mm/dd
      22      WRITE      13.31.46.  11.10.23.  11.10.23.
  N MD D
2 PROGF      DIR. PRIVATE      88      * yy/mm/dd
      0      WRITE      13.40.52.  13.40.52.  13.40.52.
  N MD A
      1 INDIRECT ACCESS FILE(S),      TOTAL PRUS =      21.
      1 DIRECT ACCESS FILE(S),      TOTAL PRUS =      88.
```

The page heading gives the user name, the family name, and the date and time. If the PN=packname parameter is specified, the family name in the heading is replaced by PN/packname.

Column headings are printed when the LO=F parameter is specified.

The following are the column headings and their meanings.

<u>Heading</u>	<u>Meaning</u>
FILE NAME	Permanent file name.
ACCESS MD/CNT	ACCESS MD is the permanent file type; direct access file (DIR) or indirect access file (IND). ACCESS CNT is the number of times the file has been accessed. It is listed on the line below ACCESS MD.
FILE-TYPE	File access category. The entry is PRIVATE, SPRIV, or PUBLIC.
LENGTH	Length of the file in decimal PRUs.
DN	Device number of the mass storage device on which the direct access file is stored. If the file resides on the master device, this field contains an asterisk.
PASSWORD	File password. It is not listed if the file belongs to another user.
INDEX	This heading is not used by the CATLIST command. It is used by the PFCAT system utility.
PERM.	Permission mode allowed the user. The entry is WRITE, MODIFY, APPEND, READ, READMD, READAP, or EXECUTE.
SUBSYS	Interactive subsystem associated with the file. The possible entries are FORT., FTNTS, BASIC, EXEC., and BATCH. If this field contains no entry, a subsystem is not associated with the file (null).
CREATION DATE/TIME	File creation time and date in the following format. 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 of the file data.
PR	Preferred residence for this file. M means MSF and N means no preference. For more information, refer to the description of the PR parameter at the beginning of this section.
BR	Requested backup copy for this file. Y means tape backup, MD means MSF or tape backup, and N means no backup required. For more information, refer to the description of the BR parameter at the beginning of this section.

HeadingMeaning

RS Actual residence of the file. D means disk, A means MSF, and B means copies of the file exist on disk and MSF. For more information, refer to Mass Storage Facility in section 2.

CHANGE COMMAND

The CHANGE command changes certain characteristics of a permanent file. The file need not be assigned to the job.

Format:

CHANGE,nfn₁=ofn₁,nfn₂=ofn₂,...,nfn_n=ofn_n/PW=password,CT=ct,M=m,BR=br,PR=pr,
SS=subsystem,PN=packname,R=r,NA,CE,WB.

nfn_i=ofn_i Specifies the file name nfn_i replaces old permanent file name ofn_i. If no name change is desired, only ofn_i is specified.

The full descriptions of the following optional parameters (except CE and M=m) are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
PW=password	Specifies the new password. If PW=0 is specified, the CHANGE command clears the old password without setting a new password.
CT=ct	Specifies the new access category for the file. Entries are P, PR, or PRIVATE for private; S or SPRIV for semiprivate; and PU or PUBLIC for public. If CT=ct is not specified, CT=PRIVATE is assumed.
M=m	Specifies the new alternate user permission mode for semiprivate and public files. For direct access files, refer to the permission modes described in the DEFINE command description; for indirect access files, refer to the permission modes described in the SAVE command description.
BR=br	Specifies the new backup copy selection; entries are tape (Y), MSF (MD), and no backup (N).
PR=pr	Specifies the new preferred residence; entries are MSF (M) and no preference (N).
SS=subsystem	Specifies the new interactive subsystem to be associated with the file; entries are BASIC, BATCH, EXECUTE, FORTRAN, FTNTS, and NULL.
PN=packname	Specifies the auxiliary device on which the file resides. This parameter cannot specify a new file residence.
R=r	Specifies the device type on which the file resides. This parameter cannot specify a new file residence.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.

<u>Parameter</u>	<u>Description</u>
CE	Clears file error code. It clears the recovery error field in the catalog entry. If the job is system origin, it also clears the MSS permanent file errors.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

COMMON COMMAND

The COMMON command accesses a library type file.

Format:

COMMON, lfn₁, lfn₂, ..., lfn_n.

lfn Specifies the library file name.

You must be authorized to access library files. If a file with the specified name is already local, no action is taken. Otherwise, the system searches for a library file with that name. If the system finds the file, it assigns the file to your job as a library type file. The system issues an error message if it does not find the file.

DEFINE COMMAND

The DEFINE command can create an empty direct access permanent file. It can also change a file of the local file type into a direct access file, if the file resides on a permanent file device.

Format:

DEFINE, lfn₁=pfn₁, lfn₂=pfn₂, ..., lfn_n=pfn_n/PW=password, CT=ct, M=m, BR=br, PR=pr, PN=packname, R=r, S=space, NA, WB.

<u>Parameter</u>	<u>Description</u>
lfn _i =pfn _i	<p>If the DEFINE command creates an empty direct access permanent file, lfn_i is specified if you want (in the current job) to reference the file by a name other than its permanent file name, pfn_i. Each file name can be from one to seven characters.</p> <p>If the DEFINE command defines an existing local file as a direct access file, lfn_i is the name of the local file, and pfn_i is its new permanent file name. If lfn_i= is omitted, pfn_i is assumed to be the local file name and the permanent file name.</p>

The full descriptions of the following optional parameters (except M=m and S=space) are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
PW=password	Specifies the one- to seven-character password that others must specify to access the file.
CT=ct	Specifies the access category of the defined file. Entries are P, PR, or PRIVATE for private; S or SPRIV for semiprivate; and PU or PUBLIC for public. If CT=ct is not specified, CT=PRIVATE is assumed.
M=m	Specifies the file access mode permitted to other users if the file is semiprivate or public, and if explicit access permission has not been granted to that user. If M=m is omitted, M=READ is assumed.
	<p><u>If m is:</u> <u>Other users can attach the file in the following modes (refer to the ATTACH command description):</u></p> <p>E or EXECUTE Execute.</p> <p>R or READ Read or execute.</p> <p>RA or READAP Read append, read, or execute.</p> <p>RM or READMD Read modify, read append, read, or execute.</p> <p>A or APPEND Append.</p> <p>M or MODIFY Modify, append, read modify, read append, read, or execute.</p> <p>W or WRITE Write, modify, append, read modify, read append, read, or execute.</p> <p>N or NULL None.</p>
	<p>Special care should be taken when using the read modify or read append mode. Programs using access techniques that do not expect concurrent updating of a file may get erroneous results if these modes are used.</p> <p>CRM AAM (refer to the CYBER Record Manager Advanced Access Methods Reference Manual) does not anticipate concurrent updating of a file by another user. Therefore, if an attempt is made to alter and read a file by a concurrent user, a warning diagnostic message is issued stating that the file is bad when, in fact, it is not.</p> <p>After a file is defined, it is always assigned to the job in write mode.</p>
BR=br	Specifies the backup copy requirement; entries are tape (Y), MSF (MD), and no backup (N). If BR=br is omitted, a backup copy of the file is stored on tape.
PR=pr	Specifies the preferred file residence; entries are MSF (M) or no preference (N). If PR=pr is omitted, no preference is assumed.
PN=packname	Specifies the name of an auxiliary device on which the direct access file is to reside. If PN=packname is omitted, the file residence is determined by the PR, R, and S parameters.

<u>Parameter</u>	<u>Description</u>
R=r	Specifies the device type on which the permanent file is to reside. The device must be a permanent file mass storage device on which direct access files are allowed. If lfn _i already exists on a device other than that specified or if an illegal device is specified, a dayfile message so informs you. If an auxiliary device name is not specified by the PN=packname parameter or a previous PACKNAM command, the file is defined on a family device.
S=space	Specifies the decimal number of PRUs requested for the file. It cannot be larger than your validation limit (refer to LIMITS Command in section 7). If no device has the specified amount of space available, a dayfile message so informs you. This parameter ensures that the file is assigned to a device that has the requested space available at the time the file is defined. It does not guarantee that the space will be available when the file is written.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

If lfn_i does not exist, the device on which pfn_i resides depends on the R=r and S=space parameters.

<u>R=r</u>	<u>S=space</u>	<u>File Residence</u>
Specified	Not specified	The file resides on the device of type r with the most space available.
Specified	Specified	The file resides on the device of type r with the most space available, provided that device has as many PRUs available as specified by the space parameter.
Not specified	Specified	The file resides on the device with the most space available, provided that device has as many PRUs available as specified by the space parameter.
Not specified	Not specified	The file resides on the device with the most space available.

If an auxiliary device has been previously specified by a PACKNAM command, the file resides on that auxiliary device rather than a family device.

After the DEFINE command has been processed, the new direct access file remains attached to the job in write mode. After the file is returned, you must issue an ATTACH command to access the direct access file. If you purge an attached direct access file, the file remains attached to the job, although it has been removed from your permanent file catalog. Until you return the purged file, you cannot define a direct access file having the same local file name as the purged file.

GET COMMAND

The GET command retrieves copies of indirect access permanent files for use as local files.

Format:

```
GET, lfn1=pfn1, lfn2=pfn2, ..., lfnn=pfnn / UN=username, PW=password, PN=packname,  
R=r, NA, WB.
```

lfn_i=pfn_i Specifies the local file name lfn_i given the retrieved copy of indirect access permanent file pfn_i. If lfn_i is omitted, the local copy of the permanent file is called pfn_i. If no files are named, the system uses the primary file name for the permanent file.

The full descriptions of the following optional parameters are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
UN=username	Specifies the user name. Specified if the permanent file(s) is in another user's catalog. You must have permission to read or execute the file(s) (refer to SAVE Command in this section). If only execute permission has been granted, the file is retrieved in execute-only mode.
PW=password	Specifies the file password. Specified if UN=username is specified, and if the permanent file has a password.
PN=packname	Specifies the auxiliary device name. Specified if the permanent file(s) resides on an auxiliary device.
R=r	Specifies the device type. Specified if PN=packname is specified, or if a PACKNAM command has been processed and the device type is other than the system default.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

Each permanent file named must be an indirect access file. If the file, lfn_i, is assigned to the job before this command is processed, it is returned. The new local file is rewound after it is retrieved. More than one user can have local copies of an indirect access file assigned to their jobs simultaneously.

If your current primary file is specified as an lfn on the command, a copy of the associated permanent file, pfn, becomes the primary file. The subsystem associated with the permanent file, pfn, becomes the job's current subsystem (refer to section 8).

If an auxiliary device has been previously specified by a PACKNAM command, the system attempts to find pfn_i on the auxiliary device rather than on the family device.

OLD COMMAND

The OLD command retrieves a copy of an indirect access permanent file and makes it the primary file.

Format:

OLD, lfn=pfm/UN=username, PW=password, PN=packname, R=r, NA, ND, WB.

lfn=pfm Specifies the one- to seven-character file name lfn is given to the primary file copy of indirect access permanent file pfm. If lfn= is omitted, the primary file is named pfm.

The full descriptions of the following optional parameters (except ND) are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
UN=username	Specifies the user name. Specified if the indirect access permanent file is in another user's catalog.
PW=password	Specifies the file password. Specified if UN=username is specified, and if the permanent file has a password.
PN=packname	Specifies the auxiliary device name. Specified if the permanent file resides on an auxiliary device.
R=r	Specifies the device type. Specified if an auxiliary device of a device type other than the installation-defined default is to be used.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
ND	Specifies the no drop option. If ND is specified, OLD changes the former primary file into a local file, but does not return any files. If ND is omitted, OLD returns all files that do not have the no-auto-drop status.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

If an auxiliary device has been previously specified by a PACKNAM command, the system attempts to find the permanent file, pfm, on the auxiliary device rather than on the family device.

If you do not specify the ND parameter, the system releases all files assigned to the job, except those with a no-auto-drop status. Certain system files (for example, ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD) always have this status. You can give any local file this status by using the SETFS command. A copy of the indirect access permanent file named on the OLD command becomes the primary file. The primary file is positioned at its BOI.

The primary file is rewound before every job step. Therefore, the file positioning commands have no effect on the primary file. Also, when two commands are issued to write on the primary file, the second writes over the data written by the first because the primary file is rewound between commands.

PACKNAM COMMAND

The PACKNAM command directs subsequent permanent file requests to the specified auxiliary device.

Format:

PACKNAM,PN=packname,R=r.

or

PACKNAM,packname,R=r.

or

PACKNAM.

<u>Parameter</u>	<u>Description</u>
PN=packname	Specifies the one- to seven-character identifier of the auxiliary device to be accessed in subsequent permanent file commands.
R=r	Specifies the device type of the auxiliary device specified by packnam. You specify this parameter if you want to use an auxiliary device whose device type differs from the installation-defined default. If you do not specify this parameter for a nondefault type pack, you must specify this parameter on each subsequent permanent file command. A full description of this parameter appears at the beginning of this section.

PACKNAM allows you to omit the PN=packname parameter on command requests for files that reside on the specified packnam device. However, if you request permanent files on another auxiliary device, you must specify the PN=packname parameter on the file request, or you can enter another PACKNAM command before the request. Refer to Mass Storage File Residence in section 2 for information concerning auxiliary permanent file devices.

You can access permanent files residing on the family system devices while the PACKNAM request is in effect by specifying PN=0 on permanent file commands. To clear the effect of a PACKNAM command, include a PACKNAM command in either of the following formats.

PACKNAM.

or

PACKNAM,PN=0.

PERMIT COMMAND

The PERMIT command explicitly permits another user to access a private file in your permanent file catalog. The PERMIT command can also change the mode in which another user can access a semiprivate file.

Format:

PERMIT,pfn,username₁=m₁,username₂=m₂,...,username_n=m_n/PN=packname,R=r,NA,WB.

<u>Parameter</u>	<u>Description</u>
pfn	Name of the private or semiprivate file for which access permission is granted.
username _i =m _i	Specifies that user name username _i is granted the access permissions indicated by access mode m _i . If m _i is omitted, the read access mode is assumed. If m _i is NULL, the user is explicitly denied permission to access the file. For the available access modes, refer to DEFINE Command or SAVE Command in this section.

The full descriptions of the following optional parameters are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
PN=packname	Auxiliary device name. Specified if the permanent file resides on an auxiliary device.
R=r	Device type. Specified if an auxiliary device on a device type other than the installation-defined default is to be used.
NA	No abort option. If NA is specified, processing errors do not terminate the job.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

PURGALL COMMAND

The PURGALL command purges all permanent files in your catalog that satisfy the criteria specified by the parameters.

Format:

PURGALL, TY=ty, CT=ct, AD=ad, MD=md, CD=cd, AF, TM=tm, DN=dn, PN=packname, R=r, NA, WB.

<u>Parameter</u>	<u>Description</u>								
TY=ty	Specifies the file type to be purged.								
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>ty</u></th> <th style="text-align: left;"><u>Action</u></th> </tr> </thead> <tbody> <tr> <td>I or INDIR</td> <td>Purges all indirect access files.</td> </tr> <tr> <td>D or DIRECT</td> <td>Purges all direct access files.</td> </tr> <tr> <td>A or ALL</td> <td>Purges all files.</td> </tr> </tbody> </table> <p>If this parameter is omitted, but other parameters are specified, the system assumes ty is ALL. To purge all files if no other parameters are specified, you must specify TY=A.</p>	<u>ty</u>	<u>Action</u>	I or INDIR	Purges all indirect access files.	D or DIRECT	Purges all direct access files.	A or ALL	Purges all files.
<u>ty</u>	<u>Action</u>								
I or INDIR	Purges all indirect access files.								
D or DIRECT	Purges all direct access files.								
A or ALL	Purges all files.								
CT=ct	Specifies the file category to be purged. Entries are P, PR, or PRIVATE for private; S or SPRIV for semiprivate; and PU or PUBLIC for public.								
AD=ad	Specifies the last access date; its format is yymmdd. All files last accessed before this date are purged, unless the AF parameter is specified.								
MD=md	Specifies the last modification date; its format is yymmdd. All files last modified before this date are purged, unless the AF parameter is specified.								
CD=cd	Specifies the creation date; its format is yymmdd. All files created before this date are purged, unless the AF parameter is specified.								
AF	Specifies that files accessed after the date specified by the AD=ad parameter, modified after the date specified by the MD=md parameter, or created after the date specified by the CD=cd parameter are purged.								
TM=tm	Specifies the time of day on the date specified by the AD, MD, or CD parameter; its format is hhmmss.								
DN=dn	Specifies the device number assigned to a device during system initialization. Only files on that device are purged.								
PN=packname	Specifies the name of the auxiliary device on which the files to be purged reside. The PN parameter cannot be specified if a device number is specified (DN=dn).								
R=r	Specifies the type of auxiliary device on which the files to be purged reside. The R parameter cannot be specified if a device number is specified (DN=dn).								

<u>Parameter</u>	<u>Description</u>
NA	Specifies the no abort option. If the specified auxiliary device is not available, the job is suspended until it becomes available.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.

To purge all files in your catalog, you must enter:

PURGALL, TY=A.

AF, CT, DN, NA, R, TY, TM, and one date (either AD, MD, or CD) can be entered on a single PURGALL command.

PURGE COMMAND

The PURGE command names files to be removed from the permanent file device.

Format:

PURGE, pfn₁, pfn₂, ..., pfn_n / UN=username, PW=password, PN=packname, R=r, NA, WB.

pfn _i	Specifies the name of a permanent file to be purged. If no file is named, and if a permanent file exists that has the same name as the primary file, that permanent file is purged; the primary file remains assigned to the job.
------------------	---

The full descriptions of the following optional parameters are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
UN=username	Specifies the user name. Specified if the file(s) to be purged is in another user's catalog. To purge a file, you must have write permission for that file.
PW=password	Specifies the file password. Specified if UN=username is specified, and if the permanent file to be purged has a password.
PN=packname	Specifies the auxiliary device name. Specified if the permanent file resides on an auxiliary device.
R=r	Specifies the device type. Specified if an auxiliary device on a device type other than the installation-defined default is to be used.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs and busy files.

When a PURGE command is issued for any direct access file, the file is purged and the permanent file catalog is altered accordingly. However, if the direct access file is attached to a job, it remains attached to the job until you return it.

REPLACE COMMAND

The REPLACE command can purge an indirect access permanent file and replace it with a copy of a local file on mass storage. It can also save a copy of a local file on mass storage as a new indirect access permanent file.

Format:

```
REPLACE, lfn1=pfn1, lfn2=pfn2, ..., lfnn=pfnn / UN=username, PW=password, PN=packname,  
R=r, NA, WB.
```

<u>Parameter</u>	<u>Description</u>
lfn _i =pfn _i	Specifies that a copy of local file lfn _i becomes an indirect access permanent file named pfn _i (one- to seven-character name). If an indirect access file named pfn _i already exists, it is replaced. If lfn _i is omitted, the name of the local file is assumed to be pfn _i . If no files are named, a copy of the primary file becomes an indirect access permanent file, replacing any existing indirect access permanent file by that name.

The full descriptions of the following optional parameters are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
UN=username	Specifies the user name. Specified if the file to be replaced is in another user's catalog. To replace another user's file, you must have write permission and be validated to create indirect access permanent files (refer to LIMITS Command in section 7).
PW=password	Specifies the file password. Specified if the UN=username is specified, and if the permanent file to be replaced has a password.
PN=packname	Specifies the auxiliary device name. Specified if the permanent file to be replaced resides on an auxiliary device.
R=r	Specifies the device type. Specified if an auxiliary device on a device type other than the installation-defined default is to be used.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
WB	Specifies the wait-if-busy option. The job waits for the mounting of disk packs.

The local files, lfn_i, are rewound before and after the replace operation.

The indirect access file created has the same access category as the file it replaces. Permission information and alternate user access data for the file are retained when a file is replaced. If the file created is a new file, it is created as a private file.

SAVE COMMAND

The SAVE command retains a copy of a local file on mass storage as an indirect access file.

Format:

SAVE, lfn₁=pfn₁, lfn₂=pfn₂, ..., lfn_n=pfn_n / PW=password, CT=ct, M=m, SS=subsystem,
BR=br, PN=packname, R=r, NA, WB.

lfn_i=pfn_i Specifies that a copy of local file lfn_i becomes an indirect access permanent file named pfn_i (one- to seven-character name). If lfn_i is omitted, the name of the local file is assumed to be pfn_i. If no files are named, a copy of the primary file becomes an indirect access permanent file with the same name as the primary file.

The full descriptions of the following optional parameters (except M=m) are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>														
PW=password	Specifies the one- to seven-character password that other users must specify to access the file.														
CT=ct	Specifies the file access category; entries are P, PR, or PRIVATE for private; S or SPRIV for semiprivate; and PU or PUBLIC for public. If CT=ct is omitted, CT=PRIVATE is assumed.														
M=m	Specifies the file access mode permitted to other users if the file is public or semiprivate, and if explicit access permission has not been granted to that user. If M=m is omitted, M=READ is assumed.														
	<table><thead><tr><th><u>If m is:</u></th><th><u>Other users can:</u></th></tr></thead><tbody><tr><td>R or READ RA or READAP RM or READMD</td><td>Retrieve a copy of the file. This copy can be read or executed.</td></tr><tr><td>E or EXECUTE</td><td>Retrieve a copy of the file. This copy can only be executed.</td></tr><tr><td>A or APPEND</td><td>Append data to the file with the APPEND command.</td></tr><tr><td>M or MODIFY</td><td>Retrieve a copy of the file or append data to the file. The user can enter GET, OLD, NEW, and APPEND commands, but not a REPLACE command, for the file.</td></tr><tr><td>W or WRITE</td><td>Retrieve a copy of the file, append data to it, replace it, or purge it.</td></tr><tr><td>N or NULL</td><td>No access is allowed.</td></tr></tbody></table>	<u>If m is:</u>	<u>Other users can:</u>	R or READ RA or READAP RM or READMD	Retrieve a copy of the file. This copy can be read or executed.	E or EXECUTE	Retrieve a copy of the file. This copy can only be executed.	A or APPEND	Append data to the file with the APPEND command.	M or MODIFY	Retrieve a copy of the file or append data to the file. The user can enter GET, OLD, NEW, and APPEND commands, but not a REPLACE command, for the file.	W or WRITE	Retrieve a copy of the file, append data to it, replace it, or purge it.	N or NULL	No access is allowed.
<u>If m is:</u>	<u>Other users can:</u>														
R or READ RA or READAP RM or READMD	Retrieve a copy of the file. This copy can be read or executed.														
E or EXECUTE	Retrieve a copy of the file. This copy can only be executed.														
A or APPEND	Append data to the file with the APPEND command.														
M or MODIFY	Retrieve a copy of the file or append data to the file. The user can enter GET, OLD, NEW, and APPEND commands, but not a REPLACE command, for the file.														
W or WRITE	Retrieve a copy of the file, append data to it, replace it, or purge it.														
N or NULL	No access is allowed.														

<u>Parameter</u>	<u>Description</u>
SS=subsystem	Specifies the interactive subsystem associated with the file. This subsystem becomes associated with the permanent file pfn, and is selected automatically each time the file is retrieved for use as a primary file. If SS=subsystem is omitted, SS=NULL is assumed unless lfn is the primary file. In that case, the file is associated with the currently active subsystem. If SS is specified without a subsystem, the file is associated with the currently active subsystem.
BR=br	Specifies the backup copy requirement; entries are tape (Y), tape or MSF (MD), or no backup (N). If BR=br is omitted, a backup copy of the file is stored on tape. Because indirect access files are not stored on the MSF, BR=MD on a SAVE command is equivalent to BR=Y.
PN=packname	Specifies the name of the auxiliary device on which the indirect access file is to reside.
R=r	Specifies the device type on which the indirect access file is to reside. The device must be a permanent file mass storage device on which indirect access files are allowed.
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job.
WB	Causes the job to wait for the mounting of disk packs.

The local files, lfn_i, are rewound before and after the save operation.

LOAD/DUMP MEMORY COMMANDS

11

The load/dump memory commands transfer information that resides in your job field length to a local file or to transfer information from a load file into your job field length. The following commands are included in this category.

DMB	DMP	LOC	WBR
DMD	DMPECS	PBC	
DMDECS	LBC	RBR	

NOTE

For information concerning security restrictions associated with the use of these commands, refer to Security Control in section 3.

The DMB command dumps the exchange package, central memory, and extended memory in binary. The DMP and DMD commands dump the exchange package or central memory in octal representation and/or display code equivalences. Likewise, the DMDECS and DMPECS commands dump extended memory. These commands are particularly helpful in creating dumps for debugging purposes (refer to section 14, Debugging Aids). Other transfers of data from central memory use the PBC command, which dumps a binary record to PUNCHB, and the WBR command, which writes a binary record on a specified file.

Data is loaded to central memory by the LBC, LOC, and RBR commands. The LBC command is useful in loading binary data in an unknown format.

All numeric parameters for these commands may be expressed in octal (postradix is B) or decimal (postradix is D) notation. If no radix is specified, octal is assumed.

DMB COMMAND

The DMB command generates a binary dump of the exchange package, central memory, and extended memory associated with the current job. File ZZZZDMB receives the dump and is not rewound before or after the dump. If ZZZZDMB is assigned to the terminal before the dump, the system returns it and creates a mass storage file by that name. Each dump creates one logical record on ZZZZDMB.

Format:

DMB,ordinal,xmemory.

<u>Parameter</u>	<u>Description</u>
ordinal	Specifies the dump record name. The system appends this parameter to the letter D to form a seven-character dump record name. Allowable values of ordinal are 0 through 7777778. If the parameter ordinal is greater than 3777778, the job aborts after the dump completes. This parameter is optional.
xmemory	Determines whether the system dumps the extended memory assigned to the job. The parameter xmemory can be any whole number. If xmemory is zero or is omitted, the system dumps only the exchange package and central memory assigned to the job. If xmemory is nonzero, the system also dumps the extended memory assigned to the job.

You must not place another command (other than DMB, DMD, DMP, DMDECS, DMPECS, or EXIT) between the program to be dumped and the DMB command.

A logical record of ZZZZDMB has the following format:

59	53	17	0
D	dump record name		0
59	53	47	17
0	ID	0	5
1	version		
2	version		
3	Δ	yy/mm/dd.	
4	Δ	hh.mm.ss.	
5	machine characteristics		
59	47	17	0
0	XP	0	32g
1	exchange package		
:	(A0) register		
20	:		
21	(A7) register		
:	RA		
30	RA+1		
31			
32			
59	47	17	0
0	CM	0	CM FL/100g
1	central memory		
:			
n			
59	41	17	0
0	ECS	0	EM FL/1000g
1	extended memory		
:			
m			
59	41	0	0
	END	0	

If an extended memory error occurs, the block in error will be filled with 400g words of the following format:

59	11	0
ECSERR		nnnn

- n Octal number of central memory words assigned to the job.
- m Octal number of extended memory words assigned to the job.
- Δ Space (display code value 55g).
- nnnn Word number of the block in error (starting with 1).

DMD COMMAND

The DMD command requests a dump similar to that of the DMP command but adds the display code equivalences to the right of the octal representations. If lines are duplicated, they are suppressed and an informative message is issued to the output file.

Formats:

DMD,fwa,lwa.
DMD,lwa.
DMD.

<u>Parameter</u>	<u>Description</u>
fwa	Specifies the first word address of memory to be dumped; fwa is relative to RA. If fwa is absent, dump mode depends on the presence or absence of lwa.
lwa	Specifies the last word address plus one of memory to be dumped; lwa is relative to RA. If lwa alone is present, DMD assumes fwa is 0. If neither fwa nor lwa is present, DMD dumps the exchange package and 40g locations before and after the program address in the exchange package. Only the lower 17 bits of the program address are used.

You must not place another command (other than DMD, DMB, DMP, DMDECS, DMPECS, or EXIT) between the program to be dumped and the DMD command.

In batch jobs, the dump is written on file OUTPUT. Central memory dumps are written four words per line.

In interactive jobs, DMD is effective only within procedure files or when specified on the ENTER command. A dump from a terminal is formatted for 72-column output and written on local file ZZZDUMP. DMD displays an informative message at the terminal. ZZZDUMP is not rewound before or after the dump.

DMDECS COMMAND

The DMDECS command requests a dump of extended memory on file OUTPUT. The dump is four words per line with display code equivalences to the right of the octal representations. If lines are duplicated, they are suppressed and an informative message is issued to the output file.

Formats:

DMDECS,fwa,lwa.
DMDECS,lwa.

<u>Parameter</u>	<u>Description</u>
fwa	Specifies the first word address of extended memory to be dumped; fwa is relative to the reference address of the field in extended memory being used by the job (RAE). If fwa is absent, DMDECS assumes fwa is 0.
lwa	Specifies the last word address of extended memory to be dumped; lwa is relative to RAE.

The DMDECS command must immediately follow a program to be dumped, except that another DMDECS or a DMPECS, DMB, DMP, DMD, or EXIT command may intervene.

Dumping always stops at the field length in extended memory (FLE) if lwa is greater than FLE.

If fwa is greater than FLE, fwa is set to FLE-10. If both fwa and lwa are greater than FLE, fwa is set to FLE-10 and lwa is set to FLE.

DMP COMMAND

The DMP command can request an exchange package dump or a central memory dump.

Formats:

DMP,fwa,lwa.
DMP,lwa.
DMP.

<u>Parameter</u>	<u>Description</u>
fwa	Specifies the first word address of memory to be dumped; fwa is relative to the first word of the job's field length. If fwa is absent, the resulting dump depends on the presence or absence of lwa. If fwa is greater than the job's field length, fwa is set at the job's field length minus 10g. If fwa is greater than or equal to 400000g, the first dump address is fwa minus 400000g, memory from the first dump address through lwa is dumped, and the job is aborted.
lwa	Specifies the last word address plus one of memory to be dumped; lwa is relative to the first word of the job's field length. If lwa alone is present, DMP assumes fwa is 0. If neither fwa nor lwa is present, DMP dumps the exchange package and 40g locations before and after the address in the program address register in the exchange package. Only the lower 17 bits of the program address are used. If lwa is greater than the job's field length, the dump stops at the end of the field length.

If either fwa or lwa is nonnumeric, DMP dumps the exchange package and 40g locations before and after the program address register in the exchange package. If both fwa and lwa are greater than the job's field length, the last 10g words of the job's field length are dumped. If fwa equals lwa, the 10g words beginning at fwa are dumped. If fwa is greater than lwa, DMP issues an error message and terminates the job step.

You must not place another command (other than DMB, DMP, DMD, DMPECS, DMDECS, or EXIT) between the program to be dumped and the DMP command.

DMP suppresses duplicate lines and then issues an informative message. In batch jobs, the dump is written on file OUTPUT. Central memory dumps are written four words per line.

In interactive jobs, DMP is effective only within procedure files or when specified on the ENTER command. A dump from a terminal is formatted for 72-column output and written on local file ZZZDUMP. DMP displays an informative message at the terminal. ZZZDUMP is not rebound before or after the dump.

DMPECS COMMAND

The DMPECS command dumps the contents of extended memory of a job's field length on file OUTPUT or a user-specified file. The dump is four words per line. If lines are duplicated, they are suppressed and an informative message is issued to the output file.

A DMPECS command within an interactive job copies the contents of the extended memory of the job's field length to the local file ZZZDUMP and displays a message at the terminal informing you of the dump. ZZZDUMP is not rewound before or after the dump.

Formats:

DMPECS,fwa,lwa.
DMPECS,lwa.
DMPECS,fwa,lwa,f,lfn.

<u>Parameter</u>	<u>Description</u>
fwa	Specifies the first word address of extended memory to be dumped; fwa is relative to the reference address of the field in extended memory being used by the job (RAE).
lwa	Specifies the last word of extended memory to be dumped; lwa is relative to RAE.
f	Specifies the print format (included for compatibility with NOS/BE).
lfn	Specifies the file to dump to.

If the first format is used, the field in extended memory defined by fwa and lwa is dumped to the file OUTPUT. Display code equivalences do not appear.

If the second format is used, DMPECS assumes fwa is 0. Display code equivalences do not appear.

If the third format is used, the specified field in extended memory is dumped to lfn. The parameter f is ignored. Display code equivalences appear to the right of the octal representations, the same as the DMDECS command.

The DMPECS command must immediately follow a program to be dumped, except that another DMDECS or DMPECS, DMB, DMP, DMD, or EXIT command may intervene.

Dumping always stops at the field length in extended memory (FLE) if lwa is greater than FLE.

If fwa is greater than FLE, fwa is set to FLE-10. If both fwa and lwa are greater than FLE, fwa is set to FLE-10 and lwa is set to FLE.

LBC COMMAND

The LBC command is intended for loading binary data of unknown format.

Format:

LBC,addr.

<u>Parameter</u>	<u>Description</u>
addr	Specifies the address relative to RA at which binary load begins; if addr is omitted, 0 (RA) is assumed.

LBC reads only one record from file INPUT. You must make an LBC call for each record of data to be loaded. If addr is specified in the program call, binary data is loaded beginning at that address; otherwise, loading begins at the reference address (RA).

LOC COMMAND

The LOC command reads octal line images from file INPUT and enters them in the job's central memory field length.

Formats:

LOC,fwa,lwa.

LOC,lwa.

LOC.

<u>Parameter</u>	<u>Description</u>
fwa	Specifies the first word address of an area to clear (zero) before loading correction statements. If fwa is absent, LOC assumes 0.
lwa	Specifies the last word address plus 1 of the area to be cleared. If lwa is absent, LOC assumes 0.

To process the LOC command, the system reads correction statement images from the current INPUT record. A correction statement consists of an octal address and a data field. The address field specifies the location to be corrected, and the data field contains the data to be placed in that location. Both fields may start at any column as long as the address precedes the data. The address field consists of a one- to six-digit address. If it is five characters or less, it is separated from the data field by a nonoctal character (for example, a blank). If it is six characters, no separator is required.

The data field consists of 1 to 20 octal characters. If it is less than 20 characters, it is terminated by a nonblank, nonoctal character and is stored right-justified. If it is 20 characters, no terminator is required. Embedded blanks in the data field are ignored.

If both fwa and lwa are specified and both are nonzero, storage is cleared from fwa to lwa, and the octal line images are loaded at the specified addresses. If the current INPUT record is empty, LOC clears the indicated area of memory.

PBC COMMAND

The PBC routine writes one record from the specified area of central memory to file PUNCHB.

Formats:

PBC,fwa,lwa.
PBC,lwa.
PBC.

<u>Parameter</u>	<u>Description</u>
fwa	Specifies the address relative to RA at which the binary deck begins; if this parameter is omitted, the PBC operation depends upon the presence or absence of lwa.
lwa	Specifies the last word address of the binary deck. If lwa alone is present, PBC assumes that fwa is RA. If lwa equals fwa, and a nonzero value is specified, PBC adds 10_8 to lwa. If fwa and lwa are 0 or are omitted, lwa is set to the value in the lower 18 bits of RA. If the upper 12 bits of RA are 7700_8 , lwa is the lower 18 bits of the location following the prefix (77) table plus the length of the prefix table.

Central memory is not altered by PBC.

RBR COMMAND

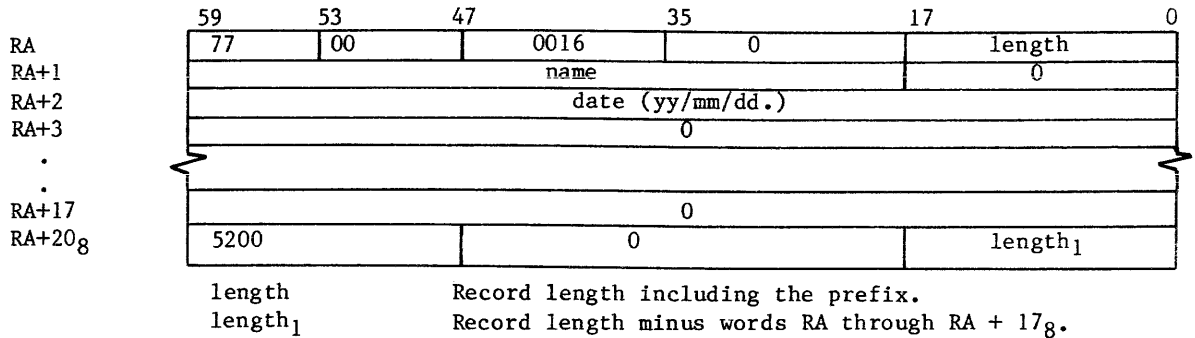
The RBR command loads one binary record from a specified file.

Format:

RBR,n,name.

<u>Parameter</u>	<u>Description</u>
n	Specifies the n is used in constructing the name of the file containing the binary record to be read. If n is less than four characters and is numeric, TAPE _n is the file name. If n contains a nonnumeric character or is four or more characters long, n itself is used as the file name. If n is absent, TAPE is the file name.
name	Specifies the one- to seven-character name used in a record prefix.

The RBR command loads one binary record from the specified file into central memory starting at RA. If the name parameter is included, a record prefix is placed in central memory starting at RA. The record itself follows. The following is the format of the record prefix.



If the record is too long for available memory, memory is filled, excess data is skipped, and an error message is issued to the job's dayfile.

WBR COMMAND

The WBR command writes a binary record from central memory to a file at its current position.

Format:

WBR,n,rl.

<u>Parameter</u>	<u>Description</u>
n	Specifies the n is used in constructing the name of the file on which the binary record is to be written. If n is less than four characters and is numeric, TAPEn is the file name. If n contains a nonnumeric character or is four or more characters long, n itself is used as the file name. If n is absent, TAPE is the file name.
rl	Specifies the record length in words. If rl is 0 or absent, the length is taken from the lower 18 bits of RA.

WBR begins writing from RA.

This section describes commands used with magnetic tape files.[†] For additional information on NOS magnetic tape files, consult the glossary for definitions of terms; Magnetic Tape Files in section 2 for descriptions of tape labels and data formats; and appendix G for tape label formats. Section 7 describes the RESOURC command required in jobs that use more than one tape or removable auxiliary pack concurrently.

NOTE

The term file as used in this section may refer to a multifile file. Refer to table 2-1 for the EOR and EOF marks for tape files.

The commands described in this section are:

ASSIGN	Assigns a local file to a tape unit [†] (system origin jobs or jobs with system origin privileges only). Section 9 describes the ASSIGN command for nontape files.
BLANK	Blank labels a tape and may restrict access to the labeled tape.
LABEL	Assigns a local file to a magnetic tape, [†] creates and verifies tape labels, and creates and accesses multifile set tapes.
LISTLB	Lists tape labels.
REQUEST	Assigns a local file to a magnetic tape device. [†]
VSN	Associates a file name with one or more VSNs for later assignment by a LABEL or REQUEST command.

TAPE ASSIGNMENT

Whenever a tape is mounted, the system checks for labels. If the tape is labeled, the system records the volume serial number (VSN) read from the VOL1 label and the equipment on which the tape is mounted. When a tape assignment is requested by a LABEL or REQUEST command specifying an lfn and a VSN (or an lfn that has been named in a previous VSN command), the system compares the VSN with the VSNs read from mounted tapes. If a match is found, the system automatically assigns the tape to the requesting job, provided a deadlock would not occur.^{††} If the tape is not mounted, the system rolls out the job until a tape with the requested VSN is mounted.

[†]If you do not specify a VSN parameter or an MT or NT parameter on the tape assignment command, the operator can assign the file to any device.

^{††}Refer to the RESOURC command in section 7.

For a mounted, unlabeled tape, the operator enters a command specifying the requested VSN. The system can then assign the tape. A VSN which contains nonalphanumeric characters should not be specified in a request for an unlabeled tape because nonalphanumeric characters cannot be entered with the operator command.

If a VSN is not associated with the requested lfn, the system directs the operator to assign an available device.

COMMAND RULES

On the tape assignment commands (LABEL, REQUEST, and ASSIGN), you can specify the tape label contents, tape density, track type, nine-track conversion mode, data format, noise size, and processing options. If any of these specifications are omitted, the system uses a default value.

NOTE

1. For nine-track tapes, the density specification given on the tape assignment is used only when the tape is written from load point. Otherwise, the tape is read or written using the density previously used for that tape. To ensure that a labeled tape is at load point for rewriting the tape at a new density, perform one of the following before the write operation.
 - Rewind the tape.
 - Specify the W parameter on the LABEL command used to assign the tape.
 - Assign the tape using a REQUEST or ASSIGN command.
2. Do not create labeled S or L format tapes with tape marks embedded in the data. Future adherence to ANSI standards will make these tapes nonstandard as the ANSI standard allows tape marks to be used only as delimiters of label groups.

Do not specify duplicate or equivalent parameters. When you specify a parameter, do not use = as a separator unless the parameter requires the keyword=value format.

You can use a continuation line for an ASSIGN, BLANK, LABEL, REQUEST, and VSN command when any one of these requires more than 80 characters. If, in processing one of these commands, the system does not encounter a termination character prior to the end of the line, it assumes the next line is a continuation line. A continuation line should be terminated with a valid terminator. The terminator for a continuation line must appear in or before column 80.

NOTE

The system accepts continuation lines from an interactive terminal only if they are within a procedure file.

You can use literals for parameters that contain nonalphanumeric characters. These parameters are FI/L, FA, SI/M, VA, FA, OFA, and VSN. Nonalphanumeric characters are characters other than letters, numbers, and asterisks.

A literal is a character string delimited by dollar signs. Blanks within literals are retained. If the literal is to contain a dollar sign, two consecutive dollar signs must be included. Thus, the literal

\$A B\$\$41\$

is interpreted as

A B\$41

When continuation lines are used, a literal cannot extend from one line to another.

PROCESSING OPTIONS

The PO= parameter on the LABEL, ASSIGN, and REQUEST tape assignment commands specify one or more processing options that are to apply to that tape file. The characters representing the processing options and their meaning are listed below.

PO=S gives the default end-of-tape conditions. Default error recovery attempts to recover blocks having errors by repeatedly rereading the block. If the A, E, or N processing options are not specified, the program determines whether an error aborts the job or the program performs error processing (refer to the FET ep bit description in Volume 4, Program Interface).

<u>P_i</u>	<u>Description</u>
A	Automatically aborts job on an irrecoverable read or write parity error (refer to the N option).
E	Specifies error inhibit. All hardware read/write errors are ignored and processing continues. The system does not attempt error recovery, issue error messages, or return error status. During a read operation, blocks less than noise size (refer to the NS parameter) are unconditionally bypassed. Use this option guardedly. Use it to recover portions of data from a bad tape, to check out hardware, and to write on tape without skipping bad spots; in the latter case, you are responsible for verifying that the data is written correctly.

<u>Pi</u>	<u>Description</u>
F	Forces unload. Unload at end of usage. (Refer to the U option.)
G	Disables all hardware error correction activity in GE (6250 cpi) write mode. An on-the-fly error while writing a GE tape results in standard error recovery processing. The system erases the defective portion of tape, thereby reducing the amount of data that can be stored on the tape. The default is installation-defined (refer to the H option).
H	Enables hardware error correction activity in GE (6250 cpi) write mode. The system allows certain types of single track errors to be written that can be corrected when the tape is read (on-the-fly correction). This is the recommended mode because it provides efficient throughput, error recovery, and tape usage when writing GE tapes on a medium that is suitable for use at 3200 fci or 6250 cpi. The default option (G or H) is installation-defined.
I	Rewrites the block on which the end-of-tape occurred as the first block on the next volume, if the system senses the EOT during a write operation. During a read operation, the block on which the EOT occurred is ignored and reading continues on the next volume. If a tape mark and the EOT are sensed at the same time, the EOT is ignored. This option cannot be specified for I or SI format tapes. Refer to the P and S options.
L	Issues only the first and last error messages for each bad tape block. Numerous attempts are made to read each bad block, but only the messages for the first and last attempts are issued to the dayfile. The default is installation-defined (refer to the M option).
M	Issues an error message for each attempt to read a bad tape block. The default is installation-defined (refer to the L option).
N	Specifies that job is not automatically aborted on an irrecoverable read or write parity error (refer to the A option); data is passed to the job on a read operation without error status set even if the program requested error processing. Use this option guardedly.
P	Writes a trailer sequence following the block on which the EOT was sensed, if the system senses the EOT during a write operation. Any data that occurs following the block on which EOT was sensed, yet before the tape mark, is ignored. During a read operation, the system transfers the block on which the EOT was sensed to the user job. The read operation resumes on the next reel. If a tape mark and the EOT are sensed at the same time, the EOT is ignored. Refer to the I and S options.
R	Enforces ring out. If the tape is mounted with the write ring in, job processing is suspended until the operator remounts the tape correctly.
S	Specifies where the system is to stop on an exit condition. For unlabeled tape, it directs the system to stop at the first tape mark after the EOT is sensed. For labeled tape, it directs the system to stop at the tape mark plus EOF1 or the tape mark plus EOV1 when the EOT is encountered.

<u>Pi</u>	<u>Description</u>
	If, during a write operation, the system senses the end-of-tape, the system writes a trailer sequence following the block on which the EOT was sensed. This trailer sequence consists of a tape mark followed by an EOVI label for labeled tapes and four tape marks for unlabeled tapes. The next block is written on the next volume. During a read operation, the EOT is noted and the system transfers to the user job the block on which the EOT was sensed plus all following blocks until a trailer sequence (as described previously) is recognized. Reading resumes on the next volume.
U	Inhibits unload. Do not unload at the end of usage. For system origin jobs, the inhibit unload option is selected by default; for all other jobs, omission of the U option causes the tape to be unloaded at end of usage.
W	Enforces ring in. If the tape is mounted without the write ring in, job processing is suspended until the operator remounts the tape correctly.

If both ring enforcement options (R and W) are specified or more than one EOT option (I, P, or S) is specified, the system issues a dayfile message and terminates the job step.

For further information on end-of-tape/end-of-reel conditions, refer to the CLOSER, REWIND, UNLOAD, and LABEL macros in Volume 4, Program Interface.

ASSIGN COMMAND

The ASSIGN command names a tape unit and the local file to be assigned to that unit. It can create an unlabeled tape file or access an existing labeled or unlabeled tape. It cannot create or verify tape labels.

NOTE

Only system origin jobs or users validated for system origin privileges (debug mode) and for use of magnetic tapes can use the ASSIGN command to assign a tape unit.

Jobs that use this command without proper validation are aborted, and a dayfile message is issued.

Before performing the assignment, the system unloads the local file (refer to the UNLOAD command in section 9).

The following description applies only to magnetic tape files; for use of the ASSIGN command with devices other than magnetic tape, refer to section 9.

Format:

ASSIGN, nn, lfn, VSN=vs₁/vs₂=...=vs_{n-1}/vs_n, $\left\{ \begin{array}{l} \text{MT} \\ \text{NT} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{D=den} \\ \text{den} \end{array} \right\}$, F=format, LB=lb,
 $\left\{ \begin{array}{l} \text{FC=fcount} \\ \text{C=ccount} \end{array} \right\}$, CV=cv, NS=ns, PO=p₁p₂...p_n, $\left\{ \begin{array}{l} \text{CK} \\ \text{CB} \end{array} \right\}$.

Required parameters:

- nn Device or device type to which the file lfn is assigned. nn can be the EST ordinal† of a magnetic tape unit or one of the device types MT or NT. Specifying MT informs the operator to assign the file to a seven-track magnetic tape drive; NT informs the operator to assign the file to a nine-track magnetic tape drive. Omission of this parameter results in an error.
- lfn Name of the file to be assigned to the device nn. Omission of this parameter results in an error.

Optional parameters:

<u>Parameter</u>	<u>Description</u>
VSN=vs ₁ /vs ₂ =...=vs _{n-1} /vs _n	One- to six-character volume serial number that uniquely identifies a reel of tape. ASSIGN does not use the VSN parameter to assign the tape. The nn parameter determines the tape assignment.
MT or NT	Specifies seven-track (MT) or nine-track (NT) tape drive. It must not conflict with the nn specification.
D=den or den	Tape density; must not conflict with the MT or NT specification. The default is installation-defined. The parameter is ignored for nine-track tapes not positioned at load point. Can be one of the following:

<u>Seven-track (MT)</u>		<u>Nine-track (NT)</u>	
<u>den</u>	<u>Density</u>	<u>den</u>	<u>Density</u>
LO	200 bpi	HD	800 cpi
HI	556 bpi	PE	1600 cpi
HY	800 bpi	GE	6250 cpi
200	200 bpi	800	800 cpi
556	556 bpi	1600	1600 cpi
800	800 bpi	6250	6250 cpi

†Contact installation personnel for a list of EST ordinals.

<u>Parameter</u>	<u>Description</u>												
F=format	Data format. Default is I. Refer to Magnetic Tape Files in section 2 for descriptions of the data formats.												
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;"><u>Format</u></th> <th style="text-align: center;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>I</td> <td>Internal.</td> </tr> <tr> <td>SI</td> <td>System internal.†</td> </tr> <tr> <td>L</td> <td>Long block stranger.</td> </tr> <tr> <td>S</td> <td>Stranger.</td> </tr> <tr> <td>F</td> <td>Foreign.</td> </tr> </tbody> </table>	<u>Format</u>	<u>Description</u>	I	Internal.	SI	System internal.†	L	Long block stranger.	S	Stranger.	F	Foreign.
<u>Format</u>	<u>Description</u>												
I	Internal.												
SI	System internal.†												
L	Long block stranger.												
S	Stranger.												
F	Foreign.												
LB=lb	Labeled or unlabeled tape. Default is KU if VSN is omitted or KL if VSN is specified.												
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;"><u>lb</u></th> <th style="text-align: center;"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>KL</td> <td>ANSI-labeled. If the tape is a NOS tape, volume and header label access restrictions are enforced (refer to appendix G).</td> </tr> <tr> <td>KU</td> <td>Unlabeled.</td> </tr> <tr> <td>NS</td> <td>Nonstandard-labeled. Assumes data begins immediately after the first tape mark.</td> </tr> </tbody> </table>	<u>lb</u>	<u>Meaning</u>	KL	ANSI-labeled. If the tape is a NOS tape, volume and header label access restrictions are enforced (refer to appendix G).	KU	Unlabeled.	NS	Nonstandard-labeled. Assumes data begins immediately after the first tape mark.				
<u>lb</u>	<u>Meaning</u>												
KL	ANSI-labeled. If the tape is a NOS tape, volume and header label access restrictions are enforced (refer to appendix G).												
KU	Unlabeled.												
NS	Nonstandard-labeled. Assumes data begins immediately after the first tape mark.												
FC=fcount or C=ccount	Whenever F format is specified, this parameter must be specified. It specifies maximum block size in frames. No default value. Illegal for other tape formats.												
CV=cv or N=cv	Conversion mode†† for nine-track tapes; applies to both labels and data on coded tapes; applies only to labels on binary tapes. Default is installation-defined. Parameter is ignored for unlabeled I or SI format binary tapes whose trailer labels are always ASCII. Must not be specified with MT or seven-track density specification.												
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;"><u>cv</u></th> <th style="text-align: center;"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>AS</td> <td>ASCII/display code conversion.</td> </tr> <tr> <td>US</td> <td>Same as AS.</td> </tr> <tr> <td>EB</td> <td>EBCDIC/display code conversion.</td> </tr> </tbody> </table>	<u>cv</u>	<u>Meaning</u>	AS	ASCII/display code conversion.	US	Same as AS.	EB	EBCDIC/display code conversion.				
<u>cv</u>	<u>Meaning</u>												
AS	ASCII/display code conversion.												
US	Same as AS.												
EB	EBCDIC/display code conversion.												

† NOS/BE system default tape format (binary mode only); used for tape interchange with NOS/BE systems.
†† Refer to Magnetic Tape Users in appendix A.

<u>Parameter</u>	<u>Description</u>
NS=ns	Noise size. Ignored for I and SI format tapes. Default is 18 frames for other formats. Maximum value is 31 frames. If NS=0 is specified, the default is used.
PO=p ₁ p ₂ ...p _n	A string of characters (not separated by commas) that specify processing options (refer to Processing Options in this section).
CK or CB	Specifies that lfn is to be used as a checkpoint file (refer to section 13).
	CK Each dump is written at the previous EOI of lfn.
	CB Each dump is written at the BOI of lfn.

Example:

ASSIGN,51,TAPE1,D=PE,F=SI.

This command assigns the file TAPE1 to the nine-track magnetic tape unit identified by EST ordinal 51.

BLANK COMMAND

The BLANK command writes the ANSI standard labels VOL1, HDR1, and EOF1 following the load point of a tape. The labels are written as follows (asterisks represent tape marks):

VOL1	HDR1	*	*	EOF1	*	*					
------	------	---	---	------	---	---	--	--	--	--	--

If the value of a labeled field is specified by a BLANK command parameter, that value is written; otherwise, the default value is used. Refer to appendix G for the tape label formats and default values.

NOTE

A BLANK command issued in a nonsystem origin job cannot overwrite a label containing an unexpired expiration date or a nonblank VA field.

If the FA field within the label is nonblank, a nonsystem origin job must specify the FA character using the OFA parameter. If the FA character is A, only the owner or a system origin job can overwrite the label.

Format:

BLANK, VSN=vsn, $\left\{ \begin{array}{l} \text{MT} \\ \text{NT} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{D=den} \\ \text{den} \end{array} \right\}$, CV=cv, FA=fa, OFA=ofa, VA=va,

OWNER=username/familyname, LSL=ls1, U.

<u>Parameter</u>	<u>Description</u>																																
VSN=vsn	One- to six-character volume serial number that uniquely identifies the reel of tape. It is entered in the VOL1 label. It need not match the VSN previously recorded on the tape.																																
MT or NT	Specifies seven-track (MT) or nine-track (NT) tape drive. Installation-defined default. Must not conflict with D=den specification.																																
D=den or den	Tape density; must not conflict with the MT or NT specification. The default is installation-defined. den can be one of the following. <table><thead><tr><th colspan="2"><u>Seven-track (MT)</u></th><th colspan="2"><u>Nine-track (NT)</u></th></tr><tr><th><u>den</u></th><th><u>Density</u></th><th><u>den</u></th><th><u>Density</u></th></tr></thead><tbody><tr><td>LO</td><td>200 bpi</td><td>HD</td><td>800 cpi</td></tr><tr><td>HI</td><td>556 bpi</td><td>PE</td><td>1600 cpi</td></tr><tr><td>HY</td><td>800 bpi</td><td>GE</td><td>6250 cpi</td></tr><tr><td>200</td><td>200 bpi</td><td>800</td><td>800 cpi</td></tr><tr><td>556</td><td>556 bpi</td><td>1600</td><td>1600 cpi</td></tr><tr><td>800</td><td>800 bpi</td><td>6250</td><td>6250 cpi</td></tr></tbody></table>	<u>Seven-track (MT)</u>		<u>Nine-track (NT)</u>		<u>den</u>	<u>Density</u>	<u>den</u>	<u>Density</u>	LO	200 bpi	HD	800 cpi	HI	556 bpi	PE	1600 cpi	HY	800 bpi	GE	6250 cpi	200	200 bpi	800	800 cpi	556	556 bpi	1600	1600 cpi	800	800 bpi	6250	6250 cpi
<u>Seven-track (MT)</u>		<u>Nine-track (NT)</u>																															
<u>den</u>	<u>Density</u>	<u>den</u>	<u>Density</u>																														
LO	200 bpi	HD	800 cpi																														
HI	556 bpi	PE	1600 cpi																														
HY	800 bpi	GE	6250 cpi																														
200	200 bpi	800	800 cpi																														
556	556 bpi	1600	1600 cpi																														
800	800 bpi	6250	6250 cpi																														
CV=cv or N=cv	Conversion mode† for nine-track tape labels. Installation-defined default. Must not be specified with MT or seven-track density specification. <table><thead><tr><th><u>cv</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>AS</td><td>ASCII/display code conversion.</td></tr><tr><td>US</td><td>Same as AS.</td></tr><tr><td>EB</td><td>EBCDIC/display code conversion.</td></tr></tbody></table>	<u>cv</u>	<u>Meaning</u>	AS	ASCII/display code conversion.	US	Same as AS.	EB	EBCDIC/display code conversion.																								
<u>cv</u>	<u>Meaning</u>																																
AS	ASCII/display code conversion.																																
US	Same as AS.																																
EB	EBCDIC/display code conversion.																																
FA=fa	File accessibility character indicating who has access to the labeled tape. Value entered in HDR1 and EOF1 labels. <table><tbody><tr><td>Blank</td><td>Unlimited access (default).</td></tr><tr><td>A</td><td>Only the owner of this NOS written tape can access it.</td></tr><tr><td>Other</td><td>In all future accesses of this tape, you must specify this character.</td></tr></tbody></table>	Blank	Unlimited access (default).	A	Only the owner of this NOS written tape can access it.	Other	In all future accesses of this tape, you must specify this character.																										
Blank	Unlimited access (default).																																
A	Only the owner of this NOS written tape can access it.																																
Other	In all future accesses of this tape, you must specify this character.																																

†Refer to Magnetic Tape Users in appendix A.

<u>Parameter</u>	<u>Description</u>
OFA=ofa	Old file accessibility character on a labeled tape that is to be relabeled. This parameter must be specified if the FA field is currently other than A or blank. Future accesses of the tape must specify the character specified with the FA parameter.
VA=va	Volume accessibility character indicating that the volume must be accessed as an ANSI-labeled tape (LB=KL). If VA is nonblank, only a system origin job can destroy VOL1 (for example, assign tape as unlabeled). Default is unrestricted access. Refer to the VOL1 format in appendix G.
OWNER=username/familyname	Owner identification entered in VOL1 label. Determines the owner for file accessibility (FA) parameter.
LSL=ls1	Label standard level entered in VOL1 label. Default is the number 1.

<u>ls1</u>	<u>Description</u>
1	Tape labels and data format for this volume conform to the ANSI standard.
Blank	Tape labels and data format for this volume may or may not conform to the ANSI standard.

U If U is specified, the tape is physically unloaded when returned after blank labeling. If U is omitted, physical unloading is inhibited. This parameter does not apply to system origin jobs.

An installation can use the BLANK command to restrict use of its labeled tapes. Once a tape has been blank labeled, you can modify the labels as follows:

1. If the volume accessibility field of VOL1 indicates unlimited access (that is, VA is blank), you can:
 - Include another BLANK command to change VOL1, HDR1, or EOF1 values.
 - Request the tape as unlabeled (with the parameter LB=KU) and write it in whatever format you specify.
 - Include a LABEL command to change HDR1 by specifying one or more of the parameters associated with that label and specifying the W parameter.
2. If the volume accessibility field is nonblank, you can:
 - Include a LABEL command to change HDR1. However, in requesting a tape in which VA is nonblank, you must specify an ANSI-labeled tape (with the parameter LB=KL), and therefore, cannot change or destroy the VOL1 label.
 - If validated, submit a system origin job to change VOL1.

LABEL COMMAND

Like ASSIGN and REQUEST commands, the LABEL command associates a file name lfn with a magnetic tape, usually identified by its VSN. Unlike the ASSIGN and REQUEST commands, the LABEL command can create and verify tape labels. It can also position a multifile set for access to any of its existing files or for appending a new file. The LABEL command can create and access unlabeled as well as labeled tapes.

NOTE

A LABEL command cannot overwrite a label with an unexpired expiration date (refer to appendix G).

To write the labels that begin a labeled tape (refer to Magnetic Tape Files in section 2), specify a write label (W) parameter. The W parameter always rewinds the tape to load point and rewrites the first label group. Overwriting the header labels for a tape file effectively destroys the data on the file. The label contents remain the same when a LABEL command with the W parameter names an lfn already assigned to a tape file.

If the tape was not previously part of a multifile set (the SI field in the first HDR1 label is blank), then specification of the SI and QN=9999 parameters rewrites the initial tape labels.

To position the tape after any HDR1 label other than the first HDR1 label (multifile set only), the SI parameter must be specified. When SI is specified, the R and W parameters are ignored unless QN=1 and the first file on the tape is to be written. The system determines where to position the tape by matching the SI, FI, and QN parameter values (if specified) to the corresponding values in the HDR1 label. (The HDR1 label format is given in appendix G.)

To write the EOF1 and HDR1 labels between two files in a multifile set (refer to figure 2-2), you specify the SI and QN=9999 parameters. The W parameter is ignored if specified when appending the file (QN=9999). To ensure that all files in a file set have the same set identifier, an appended file is given the same file set identifier as the previous file in the file set regardless of the SI=setid specification.

If neither the MT nor NT parameter is specified and no VSN is named, the operator can assign the file to any equipment. You must be validated for the assigned equipment or the job is terminated.

Format:

LABEL, lfn, VSN=vs₁/vs₂=...=vs_{n-1}/vs_n, { MT / NT }, D=den, F=format, LB=lb,
 { FC=fcount / C=ccount }, CV=cv, NS=ns, PO=p₁p₂...p_n, { CK / CB } { SI=setid / M=setid },
 { SN=secno / V=secno }, { QN=seqno / P=seqno }, { FI=fileid / L=fileid }, FA=fa, G=genno, E=gvn,
 { CR=cdate / C=cdate }, { RT=yyddd / T=ddd }, { W / R }.

Required parameter:

lfn Name of the file that resides or is to reside on magnetic tape. If lfn is already assigned to a mass storage file, processing continues with the next command. To assign a previously assigned lfn, you must return lfn before its reassignment. If lfn is already assigned to a tape and R parameter is specified, the contents of the tape labels are compared to the command parameter specifications. If the label verification fails, the job aborts. Omission of lfn results in an error.

Optional parameters:

<u>Parameter</u>	<u>Description</u>
VSN=vs ₁ /vs ₂ =...=vs _{n-1} /vs _n	One- to six-character volume serial number that uniquely identifies a reel of tape. If VSN is omitted, the operator assigns an available unit to lfn. Multiple VSNs can be specified if separated by / or = characters. If the VSNs are separated by the = character, LABEL assigns lfn to the first available VSN in the list. If the VSNs are separated by the / character, lfn is a multivolume file set, and LABEL assigns the volumes in the sequence given. If VSN=, VSN=0, or VSN=SCRATCH is specified, a scratch tape is assigned. If a scratch tape is unavailable, the job is suspended until a tape is available.
MT or NT	Requests seven-track (MT) or nine-track (NT) tape drive. Installation-defined default. Must not conflict with D=den specification.

Parameter

Description

D=den

Tape density; must not conflict with MT or NT specification. The default is installation-defined. The parameter is ignored for nine-track tape not positioned at load point. Can be one of the following:

<u>Seven-track (MT)</u>		<u>Nine-track (NT)</u>	
<u>den</u>	<u>Density</u>	<u>den</u>	<u>Density</u>
LO	200 bpi	HD	800 cpi
HI	556 bpi	PE	1600 cpi
HY	800 bpi	GE	6250 cpi
200	200 bpi	800	800 cpi
556	556 bpi	1600	1600 cpi
800	800 bpi	6250	6250 cpi

F=format

Data format. Default is I. Refer to Magnetic Tape Files in section 2 for descriptions of the data formats.

<u>format</u>	<u>Description</u>
I	Internal.
SI	System internal.†
S	Stranger.
L	Long block stranger.
F	Foreign.

LB=lb

Labeled or unlabeled tape. Default is KL.

<u>lb</u>	<u>Description</u>
KL	ANSI-labeled.
KU	Unlabeled.
NS	Nonstandard-labeled. Assumes data begins immediately after the first tape mark.

FC=fcount or C=ccount

Whenever F format is specified, this parameter must be specified. It specifies the maximum block size in frames (no default value). Illegal for other tape formats.

†NOS/BE system default tape format (binary mode only); used for tape interchange with NOS/BE systems.

<u>Parameter</u>	<u>Description</u>								
CV=cv or N=cv	Conversion mode† for nine-track tapes; applies to both labels and data on coded tapes; applies only to labels on binary tapes. Installation-defined default. Ignored for unlabeled I or SI format binary tapes whose trailer labels are always ASCII. Must not be specified with MT or seven-track density specification.								
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>cv</u></th> <th style="text-align: left;"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>AS</td> <td>ASCII/display code conversion.</td> </tr> <tr> <td>US</td> <td>Same as AS.</td> </tr> <tr> <td>EB</td> <td>EBCDIC/display code conversion.</td> </tr> </tbody> </table>	<u>cv</u>	<u>Meaning</u>	AS	ASCII/display code conversion.	US	Same as AS.	EB	EBCDIC/display code conversion.
<u>cv</u>	<u>Meaning</u>								
AS	ASCII/display code conversion.								
US	Same as AS.								
EB	EBCDIC/display code conversion.								
NS=ns	Noise size; any block containing fewer than ns frames is considered noise and discarded. Ignored for I and SI format tapes. Default is 18 frames for other formats. Maximum value is 31 frames. If NS=0 is specified, the default is used.								
PO=P ₁ P ₂ ...P _n	A string of characters (not separated by commas) that specifies processing options. Refer to Processing Options in this section.								
CK or CB	Specifies that lfn is to be used as a checkpoint file (refer to section 13).								
	<table border="0" style="margin-left: 40px;"> <tbody> <tr> <td>CK</td> <td>Each dump is written at the previous EOI of lfn.</td> </tr> <tr> <td>CB</td> <td>Each dump is written at the BOI of lfn.</td> </tr> </tbody> </table>	CK	Each dump is written at the previous EOI of lfn.	CB	Each dump is written at the BOI of lfn.				
CK	Each dump is written at the previous EOI of lfn.								
CB	Each dump is written at the BOI of lfn.								

†Refer to Magnetic Tape Users in appendix A.

Optional tape label parameters (refer to appendix G):

<u>Parameter</u>	<u>Description</u>								
SI=setid or M=setid	<p>One-to-six-character file set identifier; must be specified for file positioning within a multifile set.</p> <p>A file set identifier should be specified when the first file of a file set is written. When appending a file to a file set, the SI=setid parameter must be specified to position the multifile set, but the specified set identifier is not written in the HDR1 label. The appended file is given the same set identifier as the previous file in the file set.</p> <p>If the SI=setid parameter is omitted when the first file of a file set is written, the set identifier field in the HDR1 label is left blank. A blank set identifier field is then written in the HDR1 labels of all files in the file set.</p>								
SN=secno	<p>One-to-four-digit file section number specifying the position of the volume within a multivolume file set (numbered consecutively from 0001). The default is 1.</p>								
QN=seqno	<p>One-to-four-digit file sequence number specifying the position of the file within the multifile set (numbered consecutively from 0001). The default is 1. QN must be set to 9999 to append a new file to a multifile set.</p>								
FI=fileid or L=fileid	<p>A 1- to 17-character file identifier recorded in the HDR1 label (refer to appendix G). The default is blank.</p>								
FA=fa	<p>File accessibility character indicating who has access to the labeled tape.</p> <table border="1"> <thead> <tr> <th><u>fa</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>Blank</td> <td>Unlimited access (default).</td> </tr> <tr> <td>A</td> <td>Only the owner of the tape can access it.</td> </tr> <tr> <td>Other</td> <td>To access the tape, you must specify the character in the FA field of the HDR1 label.</td> </tr> </tbody> </table>	<u>fa</u>	<u>Description</u>	Blank	Unlimited access (default).	A	Only the owner of the tape can access it.	Other	To access the tape, you must specify the character in the FA field of the HDR1 label.
<u>fa</u>	<u>Description</u>								
Blank	Unlimited access (default).								
A	Only the owner of the tape can access it.								
Other	To access the tape, you must specify the character in the FA field of the HDR1 label.								
G=genno	<p>One- to four-digit generation number. The number zero cannot be used. The default is 1.</p>								

<u>Parameter</u>	<u>Description</u>
E=gvn	One- to two-digit generation version number. The default is 0.
CR=cdate	Creation date in the form yyddd where $1 \leq ddd \leq 366$. Used only on read operations; write operations always use the current date.
RT=yyddd or T=ddd	RT=yyddd specifies the expiration date where yy is the last two digits of the year, and ddd is the day of the year ($1 \leq ddd \leq 366$). T=ddd specifies the number of days the file is to be retained ($0 \leq ddd \leq 999$). The expiration date is entered in the HDR1 label. On or after this date the label and the file can be overwritten.
R or W	<p>If R is specified, the system compares the values recorded on the file labels with the LABEL command parameter values. If the comparison fails, it terminates the job. R is the default.</p> <p>If W is specified, the system writes ANSI standard labels on the tape. The labels contain the values specified with the LABEL command parameters or their default values. If the tape is mounted without the write ring, job processing is suspended until the operator remounts the tape correctly. If both the W and the PO=R parameters are specified, the job step aborts.</p> <p>W and R are ignored when SI is specified and $QN > 1$. When $QN = 1$ (default value) and W are specified, the initial header label is rewritten.</p>

Example 1 - Reading and Writing a Single-File File Set:

In the following job, the user reads data from one tape and writes data on another tape. Program input data previously written at 1600 cpi on a nine-track tape unit is read from tape TP01. Output data is written on tape TP02.

```
FTNJOB.
USER,USERNAM,PASSWRD.
CHARGE,*
RESOURC,PE=2.
FTN5.
LABEL,TAPE1,VSN=TP01,D=PE,PO=R.
LABEL,TAPE2,VSN=TP02,D=PE,PO=W,W.
LGO.
-EOR-
    PROGRAM SORT
    .
    .
    .
    END
-EOR-
```

The RESOURC command schedules two nine-track, 1600-cpi tape units for concurrent use in the job. The tapes are not requested until after compilation of the program in case compilation errors occur.

Assuming the compilation completes without fatal error, the LABEL commands request that two tapes, TP01 and TP02, be mounted and assigned to the job. The tape files are called TAPE1 and TAPE2 within this job. PO=R is specified for the input tape, ensuring that the tape does not have a write ring. PO=W is specified for the output tape which requires a write ring. The W parameter on the second LABEL command specifies the writing of ANSI standard labels. Default values are used in the label fields.

Following completion of the job, the tapes are rewound and unloaded.

Example 2 - Reading and Writing a Multifile Set:

The following job writes the object programs produced by three compilations as three files of a multifile set. It then copies one of the files to mass storage and executes it.

```
SJOB.
USER,USERNAM,PASSWRD.
CHARGE,*
GET,BSORT,QSORT,LSORT.
LABEL,STAPE,VSN=TP03,D=PE,SI=BINSET,FI=BSORT,PO=W,W.
FTN5,I=BSORT,B=STAPE.
LABEL,STAPE,VSN=TP03,D=PE,SI=BINSET,FI=QSORT,QN=9999.
FTN5,I=LSORT,B=STAPE.
LABEL,STAPE,VSN=TP03,D=PE,SI=BINSET,QN=2.
COPIEI,STAPE,QSORTB.
QSORTB.
```

The GET command retrieves three indirect access files - BSORT, QSORT, and LSORT - containing FORTRAN Extended source programs. The first LABEL command requests the mounting and assignment of tape TPO3 to the job. The W parameter specifies the writing of ANSI standard labels. The set identifier and file identifier fields are written using the values specified on the SI and FI parameters. The B=STAPE parameter on the FTN5 command specifies that the object program is to be written on the tape file.

The QN=9999 parameter on the second and third LABEL commands specifies that a file is to be appended to the tape. Although the same set identifier is used for all files within a file set, the SI parameter must be specified when positioning a multifile set. The second and third compilations write the second and third files of the file set.

The QN=2 parameter on the fourth LABEL command positions the tape at the second file of the file set. The tape could also have been positioned according to the file identifier (FI=QSORT). The second file is copied to a local mass storage file, QSORTB, and executed.

Example 3 - Replacing a File within a Multifile Set:

To replace a file within a multifile set, you must first copy to temporary storage the files that follow the file to be replaced, then write the replacement file, and last rewrite the succeeding files in the file set.

```
RWJOB.  
USER,USRNAME,PASWRD.  
CHARGE,*.  
GET, SORT2.  
LABEL, STAPE, VSN=TPO3, D=PE, SI=BINSET, QN=2, PO=W  
COPYEI, STAPE, SCRATCH.  
LABEL, STAPE, VSN=TPO3, D=PE, SI=BINSET, QN=1.  
COPYEI, SORT2, STAPE.  
LABEL, STAPE, VSN=TPO3, D=PE, SI=BINSET, FI=QSORT, QN=9999.  
COPYEI, SCRATCH, STAPE.
```

The first LABEL command requests tape TPO3 containing the multifile set created in example 2. The tape is mounted with a write ring inserted (PO=W) and is positioned at the second file of the file set (QN=2). The COPYEI command copies the second file to the mass storage file, SCRATCH.

The second LABEL command positions the tape at the first file. Although the contents of the first file are rewritten, the file labels remain unaltered.

The third LABEL command appends the second file to the file set (QN=9999). The labels are rewritten so you can change the label contents. The third COPYEI command writes the second file as stored on file SCRATCH.

The third file of the multifile set is lost, because it was not saved before the tape was rewritten.

LISTLB COMMAND

The LISTLB command lists the labels of an ANSI-labeled tape file. For a correct listing of labels, the data format specified on the tape assignment command must correspond to the format in which the tape was written.

Format:

$$\text{LISTLB, lfn, } \left\{ \begin{array}{l} \text{SI=setid} \\ \text{M=setid} \end{array} \right\}, \left\{ \begin{array}{l} \text{QN=seqno} \\ \text{P=seqno} \end{array} \right\}, \text{LO=ltype, L=out.}$$

<u>Parameter</u>	<u>Description</u>
lfn	File name assigned to tape file whose labels are to be listed. Default is file name TAPE. If you use the default name, assign TAPE to the labeled tape being read.
SI=setid or M=setid	One- to six-character file set identifier. If specified, only label groups whose HDRl label contains this value are listed. You must specify this parameter to list the labels of a multifile set.
QN=seqno	One- to four-digit file sequence number. If seqno is specified, only the label group whose HDRl label contains this value is listed. If seqno is specified, SI must be specified; otherwise, LISTLB terminates.
LO=ltype	Label type(s) to be listed. The default is R. Required and optional labels are listed in appendix G. Combinations of ltype mnemonics can be specified, such as LO=VH to list only the VOLn and HDRn labels. <ul style="list-style-type: none"> A List all labels. R List required labels. O List optional labels. V List VOLn labels. H List HDRn labels. F List EOFn labels. E List EOVn labels. U List UVLn, UHLn, and UTLn labels.
L=out	File on which the labels are to be listed. Default is OUTPUT.

To list labels for a multifile set (lfn contains more than one HDR1/EOF1 label pair), you must specify the SI=setid parameter and the tape must be positioned at load point. LISTLB then positions the tape for reading the requested labels. It searches for labels through all volumes associated with lfn. At the end of the multifile set or if an expected label group is not found, the following dayfile message is issued. n is the sequence number of the last file found. (nmn should be ignored.)

```
MULTI-FILE NOT FOUND, lfn AT nnn.  
REQUEST SECTION n+1.  
FOUND SECTION n.
```

After issuing this dayfile message, LISTLB leaves the tape positioned after the last listed label. The next command processed for the tape file must be either RETURN, EVICT, UNLOAD, or LABEL.

For a tape file that is not a multifile set, there are two cases. If you specify trailer labels, LISTLB leaves the tape positioned before the trailer labels last listed. Otherwise, LISTLB leaves the tape positioned after the label last listed.

Example 1:

The following commands list the second label group of file set ABCDEF.

```
LABEL,T,VSN=EXAMP1,MT,D=HY,SI=ABCDEF.  
LISTLB,T,SI=ABCDEF,QN=2.
```

Example 2:

To list only the volume and header labels (trailer labels omitted) of a multivolume file set, you must request a volume of the file set, list its labels, and return the file set, repeating the procedure for each volume of the file set.

```
LABEL,T,VSN=REEL1,MT,D=HY.  
LISTLB,T,LO=VH.  
RETURN,T.  
LABEL,T,MT,D=HY,VSN=REEL2.  
LISTLB,T,LO=VH.
```

Example 3:

To list all labels of the following file set, only one LISTLB command is required.

```
VSN,T=REEL1/REEL2.  
LABEL,T,VSN=REEL1,D=HY.  
LISTLB,T.
```

The LISTLB(T) command lists all labels on the volumes associated with T, that is, REEL1 and REEL2.

REQUEST COMMAND

The REQUEST command associates a file name, lfn, with a magnetic tape device,† which you can describe in a comment following the command terminator. This comment appears at the system console so that the operator can make the requested assignment. However, if the tape is labeled and you previously specified a VSN via the VSN control command or included the VSN parameter on the REQUEST command, the system can automatically assign the tape. The REQUEST command can create unlabeled tape files and access existing labeled and unlabeled tape files. It cannot create or verify tape labels.

The LABEL command performs the same function as the REQUEST command. Unless you have reason to do otherwise, use the LABEL command.

Format:

$$\text{REQUEST, lfn, VSN=vsn}_1/\text{vsn}_2=\dots=\text{vsn}_{n-1}/\text{vsn}_n, \left\{ \begin{array}{l} \text{MT} \\ \text{NT} \end{array} \right\}, \left\{ \begin{array}{l} \text{D=den} \\ \text{den} \end{array} \right\}, \text{F=format,} \\ \text{LB=lb, } \left\{ \begin{array}{l} \text{FC=fcount} \\ \text{C=ccount} \end{array} \right\}, \text{CV=cv, NS=ns, PO=p}_1\text{p}_2\cdots\text{p}_n, \left\{ \begin{array}{l} \text{CK} \\ \text{CB} \end{array} \right\} \text{.comment}$$

Required parameter:

lfn

Name of the file that resides or is to reside on magnetic tape. If lfn is already assigned to a mass storage file, processing continues with the next command. To assign a previously assigned lfn, you must return lfn before its reassignment. Omission of this parameter results in an error.

Optional parameters:

<u>Parameter</u>	<u>Description</u>
VSN=vsn ₁ /vsn ₂ =...=vsn _{n-1} /vsn _n	One- to six-character volume serial number uniquely identifying a reel of tape. You should specify a VSN for labeled and unlabeled tapes. If VSN is omitted, the operator must assign an available device to lfn. If VSN=, VSN=0, or VSN=SCRATCH is specified, a scratch tape is assigned. If a scratch tape is unavailable, the job is suspended until a tape is available. Multiple VSNs can be specified if separated by a / or = character. If the VSNs are separated by the = character, the system assigns lfn to the first available VSN in the list. If the VSNs are separated by the / character, lfn is a multivolume file set, and LABEL assigns the volumes in the sequence given.

†If you do not specify a VSN parameter or an MT or NT parameter on the command, the operator can assign any device to the file. If you are not authorized to use the assigned device, the job aborts.

<u>Parameter</u>	<u>Description</u>
MT or NT	Requests seven-track (MT) or nine-track (NT) tape drive. Installation-defined default. Must not conflict with D=den specification.
D=den or den	Tape density; must not conflict with MT or NT specification. The default is installation-defined. The parameter is ignored for nine-track tape not positioned at load point. Can be one of the following.

<u>Seven-track (MT)</u>		<u>Nine-track (NT)</u>	
<u>den</u>	<u>Density</u>	<u>den</u>	<u>Density</u>
LO	200 bpi	HD	800 cpi
HI	556 bpi	PE	1600 cpi
HY	800 bpi	GE	6250 cpi
200	200 bpi	800	800 cpi
556	556 bpi	1600	1600 cpi
800	800 bpi	6250	6250 cpi

F=format Data format. Default is I. Refer to Magnetic Tape Files in section 2 for descriptions of the data formats.

<u>format</u>	<u>Description</u>
I	Internal.
SI†	System internal.
S	Stranger.
L	Long block stranger.
F	Foreign.

LB=lb Labeled or unlabeled tape. Default is KL if a volume serial number is specified by the VSN parameter or by a VSN command; otherwise, the default is KU.

<u>lb</u>	<u>Description</u>
KL	ANSI-labeled.
KU	Unlabeled.
NS	Nonstandard-labeled. Assumes that data begins immediately after the first tape mark.

† NOS/BE system default tape format (binary mode only); used for tape interchange with NOS/BE systems.

<u>Parameter</u>	<u>Description</u>								
FC=fcount or C=ccount	Whenever F format is specified, this parameter must be specified. It specifies the maximum block size in frames (no default value). Illegal for other tape formats.								
CV=cv or N=cv	Conversion mode [†] for nine-track tapes; applies to both labels and data on coded tapes; applies only to labels on binary tapes. Installation-defined default. Ignored for unlabeled I or SI format binary tapes whose trailer labels are always ASCII. Must not be specified with MT or seven-track density specification.								
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>cv</u></th> <th style="text-align: left;"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>AS</td> <td>ASCII/display code conversion.</td> </tr> <tr> <td>US</td> <td>Same as AS.</td> </tr> <tr> <td>EB</td> <td>EBCDIC/display code conversion.</td> </tr> </tbody> </table>	<u>cv</u>	<u>Meaning</u>	AS	ASCII/display code conversion.	US	Same as AS.	EB	EBCDIC/display code conversion.
<u>cv</u>	<u>Meaning</u>								
AS	ASCII/display code conversion.								
US	Same as AS.								
EB	EBCDIC/display code conversion.								
NS=ns	Noise size. Ignored for I and SI format tapes. Default is 18 frames for other formats. Maximum value is 31 frames. If NS=0 is specified, the default is used.								
PO=p ₁ p ₂ ...p _n	A string of characters (not separated by commas) that specifies processing options. (Refer to Processing Options in this section.)								
CK or CB	File lfn is to be used as a checkpoint file (refer to section 9). <table border="0" style="margin-left: 40px;"> <tr> <td>CK</td> <td>Each dump is written at the previous EOI of lfn.</td> </tr> <tr> <td>CB</td> <td>Each dump is written at the BOI of lfn.</td> </tr> </table>	CK	Each dump is written at the previous EOI of lfn.	CB	Each dump is written at the BOI of lfn.				
CK	Each dump is written at the previous EOI of lfn.								
CB	Each dump is written at the BOI of lfn.								
comment	The comment appears at the system console. In the comment field, you can direct the operator to make the requested assignment.								

Example:

To send a message to the operator requesting that volume XYZ be mounted on tape unit NT62 and assigned to lfn TAPE1, you could issue the following command.

```
REQUEST,TAPE1. NEED VSN=XYZ ON NT62.
```

[†]Refer to Magnetic Tape Users in appendix A.

VSN COMMAND

The VSN command associates a file name lfn with one or more volumes of tape.† An lfn/VSN association allows the system to assign the specified VSN to lfn without reference to a VSN parameter on the LABEL or REQUEST command or to an operator command. Once declared, an lfn/VSN association remains until the file is returned by an operation such as an EVICT, RETURN, or UNLOAD command.

Format:

VSN, lfn₁=vsn₁, lfn₂=vsn₂, ..., lfn_n=vsn_n.

<u>Parameter</u>	<u>Description</u>
lfn _i	File name to be associated with vsn _i . This parameter is required if parameters are specified.
vsn _i	One or more one- to six-character volume serial numbers to be associated with lfn _i . If vsn _i contains nonalphanumeric characters, it must be a literal delimited by dollar signs (\$).

<u>vsn_i</u>	<u>Meaning</u>
Omitted	An available scratch tape is automatically assigned to lfn _i .
0	Same as omitted.
SCRATCH	Same as omitted.
vsn _a =vsn _b =...=vsn _z	Names duplicate volumes, any of which may be used with lfn _i .
vsn _a /vsn _b /.../vsn _z	Successive volumes to be assigned to lfn _i .†† The system assigns volumes in the order listed.

With a VSN command you can:

- Omit the VSN keyword from your LABEL or REQUEST commands and specify lfn/VSN associations on the VSN command instead. This allows you to specify new VSNs without changing LABEL or REQUEST command.
- Override the VSN specified on subsequent ASSIGN, LABEL, REQUEST, or VSN commands. For example, the sequence

```
VSN, FILEA=123.
VSN, FILEA=124.
LABEL, FILEA.
```

† Up to 60 VSNs can be specified for a single file name in any condition of duplicate reel and/or multireel specifications.

†† All subsequent volumes must have the same characteristics as the first volume in the sequence. (Characteristics include labels, track type, density, and conversion mode.) It is recommended that all volumes be blank labeled (refer to the BLANK command) before use in a multivolume sequence.

directs the system to assign FILEA to the tape with VSN 123. However, by returning file lfn, you can specify another lfn/VSN association. Thus, the following sequence directs the system to assign FILEA to the tape with VSN 124.

```
VSN,FILEA=123.  
RETURN,FILEA.  
VSN,FILEA=124.  
LABEL,FILEA.
```

- Associate the VSNs of two or more duplicate volumes with one file name. For example, the following command indicates that either the tape with VSN VOL100 or the tape with VSN VOL101 can be assigned to FILE1.

```
VSN,FILE1=VOL100=VOL101.
```

- Specify the VSNs of a multivolume file set. For example, the following command indicates that FILE2 may extend through the three volumes identified by VSN23, VSN24, and VSN25.

```
VSN,FILE2=VSN23/VSN24/VSN25.
```

- Specify alternate volumes within a multivolume file set.

```
VSN,FILE3=VSNA=VSN1/VSN2/VSNB=VSN3=VSN4.
```

The first volume of the set can be either VSNA or VSN1. The second volume is VSN2. The third volume can be either VSNB, VSN3, or VSN4, depending on which is available.

A job may terminate as the result of system, operator, or programmer error. For some jobs, it becomes more advantageous to accept the overhead of checkpoint procedures than to run the risk of losing the entire job output. The checkpoint/restart feature is implemented through the CKP command and the RESTART command.

NOTE

For information concerning security restrictions associated with the use of these commands, refer to Security Control in section 3.

CKP COMMAND

The CKP command causes a checkpoint dump to be taken.

Format:

CKP, lfn₁, lfn₂, ..., lfn_n.

lfn_i Specifies a file to be included in the checkpoint dump. If no files are specified, all files local to the job at the time the CKP command is processed are checkpointed.

Each time a CKP command is processed, the system takes a checkpoint dump. The dump is written on the tape or mass storage checkpoint file specified on a REQUEST, ASSIGN, or LABEL command with the CK or CB parameter. The dump consists of a copy of the job's field length in central memory, the system information used for job control, and the names and contents of all assigned files explicitly or implicitly identified by the CKP command. These files are:

- INPUT, OUTPUT, PUNCH, PUNCHB, P8, CCCCCO, and LGO. These files are always included in the checkpoint dump. Procedure scratch files ZZZZxx are also included if present.
- Common files, library type files, working copies of indirect access files, and some direct access files. If one of these types of files is specified on the CKP command, it is included in the checkpoint dump, and all other files of that type are excluded. If no files are specified, all files of these types assigned to the job are included in the dump.

The system does not include global library set files (ZZZZLD and your library files) unless you specify them on the command.

Each checkpointed file is copied according to the last operation performed on it. If the last operation was a write, the file is copied from the BOI to its position at checkpoint time; only that portion is available at restart time. The file is positioned at the latter point.

If the last operation was a read and the EOI was not detected, the file is copied from its position at checkpoint time to the EOI; only that portion is available at restart time. The file is positioned at the former point. If the last operation was a read and the EOI was detected, no copy is performed.

The exception to this rule is the type of operation performed on execute-only direct access files. If a dump is specified for this type of file, its name and associated system information are copied but the contents of the file itself is not copied. Thus, if you attempt to resume from such a dump, RESTART is unable to retrieve that file and aborts. You can avoid this by selecting the NA and FC options of the RESTART command and retrieving the file yourself.

If the checkpoint file is to reside on mass storage, you must include a SAVE or DEFINE command in the checkpoint job and a GET or ATTACH command in the restart job.

If the checkpoint file is to reside on magnetic tape, care should be taken to use a labeled or nonblank tape. An unlabeled blank tape (one which has never been used) cannot be specified as the checkpoint file since the checkpoint program attempts to read the tape to determine the number of the last checkpoint. The tape subsystem then aborts the job with a blank tape read message.

The system numbers checkpoints starting at 1 and increases by 1 to a limit of 4095. At this point, a second cycle of numbering begins, again starting at 1. An example showing how to restart from a specific checkpoint is given in the RESTART command section.

RESTART COMMAND

The RESTART command directs the system to restart a previously terminated job from a specified checkpoint.

Format:

RESTART, lfn, nnnn, x_i.

<u>Parameter</u>	<u>Description</u>
lfn	Identifies the checkpoint file; you must have write permission to lfn.
nnnn	Number of the checkpoint from which to restart; if nnnn is *, the last available checkpoint on lfn is used; if nnnn is omitted, the first checkpoint is used. The nnnn parameter can be obtained from the CHECKPOINT nnnn COMPLETE messages issued to the user's dayfile in response to CKP command.
x _i	Any of the following in any order: RI If this parameter is included, the command file on lfn is not restored. The command file of this restart job at its current position is used instead. If this parameter is not included, the entire command file of the checkpointed job is restored and set to its position at checkpoint time; any command following RESTART is not processed.

Parameter

Description

- NA If this parameter is included, RESTART does not abort if a required file is not available. Also, if NA is included and a read parity error occurs in an attempt to obtain a file from checkpoint nnnn, RESTART selects checkpoint nnnn-1 if it is available.
- FC Normally RESTART restores all files included in the specified checkpoint. However, if this option is selected, RESTART first checks if a file is already local to the restart job. If it is, RESTART does not replace it with the file on the checkpoint dump.

You must assign lfn to your job before the RESTART command is processed. You must include a REQUEST, ASSIGN, or LABEL command if lfn resides on magnetic tape or a GET or ATTACH command if lfn resides on mass storage.

Checkpoint dumps are numbered in ascending order from 1 to 4095. When nnnn equals 4095, the numbering sequence begins again at nnnn equal to 1. The value of nnnn depends on the structure of the checkpoint file, as defined by the CK and CB parameters of the REQUEST, ASSIGN, or LABEL commands.

If CK was specified when the checkpoints occurred, each dump is appended to the checkpoint file, and therefore, all dumps up to the time the job aborted are available for restart. You may specify a particular checkpoint dump in the following manner.

Assume a CK file of the name CHKFILE is being used and checkpoint number 4095 has been passed. The job is terminated at checkpoint number 10 in the second cycle of numbering. To restart the job from checkpoint 4 of the second numbering cycle, the following commands can be used.

- SKIPR,CHKFILE,8196. There are two records for every checkpoint, and 4098 checkpoints must be skipped to reach checkpoint 4 of the second numbering cycle.
- COPYBR,CHKFILE,AA,2. The fourth checkpoint is copied to file AA. At this point, file CHKFILE is not positioned correctly for subsequent checkpoints. If you intend to continue checkpointing on this file, a
- BKSP,CHKFILE.
- command should be included.
- RESTART,AA... . The job is restarted from file AA using the fourth checkpoint.

If the CB parameter was specified on the ASSIGN, LABEL, or REQUEST command naming the checkpoint file, each dump is written over the preceding dump, and therefore, only the last dump is available. If two REQUEST, ASSIGN, or LABEL commands with CB specified are submitted, successive dumps are alternated between two files; therefore, the last two dumps are available.†

†If alternate checkpoint files are used and a read parity error occurs in an attempt to read the last checkpoint, RESTART aborts even if the NA option was selected.

All files copied by RESTART are made local to the restart job. Therefore, you must make sure that any direct access files are not lost. For example, assume that direct access files X, Y, and Z are attached to a job. The job is then checkpointed and X, Y, and Z are copied to the checkpoint file lfn. To retain these files as direct access files during restart, you should include the following sequence of commands.

PURGE,X,Y,Z.

DEFINE,X,Y,Z.

RESTART,lfn,mnnn,x₁.

If the information table associated with a file was included on the checkpoint file, but the file itself was not copied, RESTART issues the appropriate commands to retrieve the file.

Some program errors prevent compilation or assembly of the source program; other errors prevent execution of the object program. You determine the cause of a compilation error using the compiler diagnostics, a source listing, and the compiler reference manual. The cause of an execution error is often more difficult to determine. If you cannot determine the cause of the error from the execution error message, you can use the interactive debug utility or interpret memory dumps to locate the cause. CYBER Interactive Debug is described in its reference manual (listed in the preface). This section describes central memory dumps and their use as a debugging aid.

CYBER 170, CYBER 70 and 6000 Computer Systems that support NOS have the central exchange jump/monitor exchange jump (CEJ/MEJ) feature. This feature enables a program to directly switch CPU control to the system monitor. The information transferred from the CPU to central memory by an exchange jump operation is called an exchange package.

You can dump the job exchange package and locations within the job field length using the DMP, DMD, and DMB commands described in section 11. (Dump restrictions are described under Security Control in section 3.) Most CPU mode errors result in an exchange package dump.

You interpret a memory dump using the load map and the compiler-generated symbolic reference map. Dump interpretation may also require knowledge of display code equivalences (appendix A), machine codes, and internal integer and floating point number representations (refer to the COMPASS Reference Manual).

EXCHANGE PACKAGE DUMPS

You can dump a job's exchange package using a DMP, DMD, or DMB command within the job (refer to section 11). Figures 14-1 and 14-2 show actual exchange package dumps. The format of the first dump is produced by all CYBER 170 Computer Systems except model 176; all CYBER 70 Computer Systems; and all 6000 Computer Systems. The second dump format is produced only by the model 176.

The following are the exchange package fields and their contents.

<u>Label</u>	<u>Contents</u>
P	Program address at which execution stopped.
RA	Reference address; starting address of central memory field length.
FL	Field length in central memory.

EXCHANGE PACKAGE.

P	242	A0	51760	B0	0	(A0)	0000	0000	0000	0000	0000
RA	622400	A1	1	B1	1	(A1)	0516	0420	0000	0000	0000
FL	52000	A2	114	B2	30	(A2)	0400	0005	0300	0007	7775
EM	7007	A3	574	B3	6	(A3)	5555	5555	5555	5555	5555
RAE	0	A4	557	B4	22	(A4)	7777	7777	7777	7777	7776
FLE	0	A5	573	B5	1	(A5)	1717	0631	4631	4631	4632
MA	3600	A6	1	B6	7776	(A6)	0516	0420	0000	0000	0000
		A7	277	B7	14657	(A7)	3232	3232	3206	0300	0171

X0	7777	7777	7777	0000	0000
X1	0000	0000	0000	0000	0000
X2	0000	0000	0000	0000	0000
X3	0000	0000	0040	0000	0000
X4	2000	0000	0000	0000	0012
X5	1717	0631	4631	4631	4632
X6	0516	0420	0000	0000	0000
X7	0000	0000	0000	0000	0000

(RA)	0000	0000	0000	0000	0000
(RA+1)	0516	0420	0000	0000	0000

Figure 14-1. Exchange Package Dump

EXCHANGE PACKAGE.

P	10435	A0	2165	B0	0	(A0)	1725	2420	2524	0000	0131
RA	136100	A1	1	B1	1	(A1)	0516	0420	0000	0000	0000
FL	15000	A2	6251	B2	777755	(A2)	1717	0631	4631	4640	3615
PSD	70000	A3	2	B3	6032	(A3)	0000	0000	0000	0000	0000
RAE	0	A4	6207	B4	11437	(A4)	0400	0062	4600	0000	0000
FLE	0	A5	4324	B5	12711	(A5)	2000	0000	0000	0000	0065
MA	1200	A6	1	B6	776677	(A6)	0516	0420	0000	0000	0000
EEA	1200	A7	12557	B7	30	(A7)	6000	0000	0000	0001	5000

X0	0000	0000	0000	0000	0000
X1	0000	0000	0000	0000	0000
X2	1717	0631	4631	4640	3615
X3	2000	0000	0000	0000	0012
X4	2000	0000	0000	0000	0000
X5	0000	0000	0000	0000	0000
X6	0516	0420	0000	0000	0000
X7	2000	0000	0000	0000	0001

(RA)	0000	0000	0000	0000	0000
(RA+1)	0516	0420	0000	0000	0000

Figure 14-2. Exchange Package Dump for Model 176

LabelContents

EM† Exit mode. This field contains control bits and status bits. Each control bit set indicates that if the hardware-detected error occurs, the program aborts. A status bit(s) is set when the specified condition occurs, except for bit 4 on models 825, 835, and 855. On models 825, 835, and 855, bit 4 is the instruction stack purging status bit and indicates whether stack purging is extended (1) or normal (0). The bit positions are numbered with 0 as the rightmost bit (each digit shown represents 3 bits). For more information, refer to the MODE macro in Volume 4, Program Interface.

<u>Bit Position</u>	<u>Error</u>
11	CM data error.††
10	Central memory control (CMC) input error.
9	Extended memory flag register operation parity error.††
8	Central memory copy flag.
7	Reserved.
6	Software flag.
5	CMU interruption flag.
4	Instruction stack purge flag.†††
3-4	Hardware error exit status bits.††††
2	Indefinite operand.
1	Operand out of range.
0	Address out of range.

The EM field in figure 14-1 has bit positions 11, 10, 9, 2, 1, and 0 set.

PSD†††††

Program status designator (PSD) register. Each bit set indicates the setting of a mode flag or an error condition. The bit positions are numbered with 0 as the rightmost bit (each digit shown represents 3 bits).

<u>Bit Position</u>	<u>Error</u>
14	Indefinite mode.
13	Overflow mode.
12	Underflow mode.

†Does not apply to model 176.

††Applies to all CYBER 170 Computer Systems except models 176, 825, 835, and 855.

†††Applies to models 825, 835, and 855 only.

††††Applies to model 74 only.

†††††Applies to model 176 only.

<u>Label</u>	<u>Bit Position</u>	<u>Contents</u> <u>Error</u>
	11	LCME error.
	10	CM error.
	9	LCME block range error.
	8	CM block range error.
	7	LCME direct range error.
	6	CM direct range error.
	5	Program range error.
	4	Not used.
	3	Step condition.
	2	Indefinite condition.
	1	Overflow condition.
	0	Underflow condition.

The PSD field in figure 14-2 has bit positions 14, 13, and 12 set.

RAE	Extended memory reference address; starting address of extended memory field length.
FLE	Extended memory field length.
MA	Monitor address (normal exit address for the model 176).
EEA	Error exit address (model 176).†
Ai	Contents of the address registers.
(Ai)	Contents of the central memory word addressed by the named address register.
Bi	Contents of the increment registers.
Xi	Contents of the operand registers.
(RA)	Contents of the reference address word.
(RA+1)	Contents of the request word following the reference address word.

†Normally this does not apply to you if you are writing applications programs.

USING DUMPS

You receive an exchange package and partial CM dump when a hardware-detected error occurs. You can also obtain dumps of your job's exchange package and field length by including a DMP, DMB, or DMD command in your job. If you are using FORTRAN, you can generate a CM dump within a program using the DMP subroutine (refer to the FORTRAN 5 Reference Manual). If you are using COMPASS, you can specify the REPRIEVE macro to control error processing and the SYSTEM macro to generate dumps (refer to Volume 4, Program Interface).

When the system hardware detects one of the error conditions listed in the MODE command description (section 6), NOS dumps the job exchange package and the contents of the 32 words preceding and the 32 words succeeding the address where the job step terminated.

You can specify commands after an EXIT command; the commands then are processed only if the job step terminates abnormally (refer to Error Control in section 3). If you specify a dump command to be executed and if the job step aborts, you can analyze the resulting dump to determine the cause of the job step abort.

Example 1 - Finding the Source Program Location Where the Program Terminated:

This example uses a dump to find the cause of an execution error.

You submit the following job to compile and execute a FORTRAN 5 program called FPROG.

```
FJOB.  
USER,username,password.  
CHARGE,*.  
GET,FPROG.  
FTN5,I=FPROG.  
MAP,PART.  
LDSET,PRESET=ZERO.  
LGO.
```

The FTN5 command compiles the source program in the retrieved file FPROG. The resulting program listing and symbolic reference map are shown in figure 14-3. The fields within the reference map are defined in the FORTRAN 5 Reference Manual.

```

PROGRAM FPROG      73/74  OPT=1                      FTN                      11.03.07      PAGE      1
1      PROGRAM FPROG
      DIMENSION N(10)
      DATA (N(I),I=1,10)/1,2,3,4,5,6,7,8,9,10/
5      NSUM=0
      DO 10 I=1,111111
10     NSUM=NSUM+N(I)
      STOP
      END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
4137 FPROG

VARIABLES      SN TYPE      RELOCATION
4153 I          INTEGER
4152 NSUM      INTEGER      4154 N          INTEGER  ARRAY

FILE NAMES      MODE
0 INPUT        2054 OUTPUT

STATEMENT LABELS
0 10

LOOPS LABEL     INDEX     FROM-TO     LENGTH     PROPERTIES
4144 10         I         5 6        3B        INSTACK

STATISTICS
PROGRAM LENGTH      160B      112
BUFFER LENGTH      4006B      2054
52000B CM USED

```

Figure 14-3. Example 1: Program Listing and Symbolic Reference Map

The MAP(PART) command instructs the loader to generate a partial load map when it loads the program. The LDSET(PRESET=ZERO) command tells the loader to set uninitialized memory words to zero during the next load. (Refer to the CYBER Loader Reference Manual for loader command descriptions.)

The LGO command loads and executes the object program that the compiler wrote on file LGO. The partial load map is shown in figure 14-4. The fields within the load map are defined in the CYBER Loader Reference Manual.

The job dayfile is shown in figure 14-5. Program execution terminates abnormally, resulting in the following error message.

CM OUT OF RANGE.

As explained in appendix B, this message indicates that the program references an address outside the job field length.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCSSR	VER	LEVEL	HARDWARE	COMMENTS
FPROG	111	4166	LGO	79/08/29	FTN	4.8	505	666X I	PROGRAM OPT=1
/STP.END/	4277	1							
/FCL.C./	4300	26							
/Q8.IO./	4326	144							
Q2NTRY=	4472	1	SL-FORTRAN	79/07/09	COMPASS	3.6	505		FCL INITIALIZATION ROUTINE.
/FCL=ENT/	4473	40							
FCL=FDL	4533	40	SL-FORTRAN	79/07/09	COMPASS	3.6	505		FCL CAPSULE LOADING
FEIFST=	4573	3	SL-FORTRAN	79/07/09	COMPASS	3.6	505		CONVERTED DATA STORAGE
FORSYS=	4576	515	SL-FORTRAN	79/07/09	COMPASS	3.6	505		FORTRAN OBJECT LIBRARY UTILITIES.
FORUTL=	5313	25	SL-FORTRAN	79/07/09	COMPASS	3.6	505		FCL MISC. UTILITIES.
GETFIT=	5340	64	SL-FORTRAN	79/07/09	COMPASS	3.6	505		LOCATE AN FIT GIVEN A FILE NAME.
SYS AID=	5424	1	SL-FORTRAN	79/07/09	COMPASS	3.6	505		LINK BETWEEN SYS=AID AND INITIALIZATION CO
CPUCPM	5425	5	SL-SYSLIB	79/06/09	COMPASS	3.6	498		79/05/10. CONTROL POINT MANAGER
CPU.SYS	5432	40	SL-SYSLIB	79/07/09	COMPASS	3.6	505		.PROCESS SYSTEM REQUEST.
CMF.ALF	5472	162	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - ALLOCATE FIXED.
CMF.CSF	5654	6	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA	5662	14	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF	5676	36	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - FREE FIXED.
CMF.GSS	5734	22	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - GET SUMMARY STATISTICS.
CMM.KIL	5756	12	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - DEACTIVATE CMM.
CMM.MEM	5770	7	SL-SYSLIB	79/07/09	COMPASS	3.6	505		
CMM.R	5777	206	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF	6205	22	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CMM V1.1 - SHRINK AT LWA FIXED.
/FDL.COM/	6227	14							
FDL.RES	6243	211	SL-SYSLIB	79/07/09	COMPASS	3.6	505		FAST DYNAMIC LOADER RESIDENT.
FDL.MMI	6454	222	SL-SYSLIB	79/07/09	COMPASS	3.6	505		FDL MEMORY MANAGER INTERFACE.
CTL\$RM	6676	427	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CRM CONTROLLING ROUTINE.
ERR\$RM	7325	25	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CRM ERROR PROCESSOR ENTRY.
LIST\$RM	7352	67	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CRM - ALLOCATE SPACE FOR LIST OF FILES
RM\$SYS=	7441	5	SL-SYSLIB	79/07/09	COMPASS	3.6	505		CRM - POST RA+1 REQUEST

.285 CP SECONDS 24000B CM STORAGE USED 2 TABLE MOVES

Figure 14-4. Example 1: Partial Load Map

```

11.03.06.FJOB.
11.03.06.USER(USRNAME,,FAMNAME)
11.03.06.CHARGE(CHRGNUM,PROJNUM)
11.03.07.GET,FPROG.
11.03.07.FTN,I=FPROG.
11.03.08. .093 CP SECONDS COMPILATION TIME
11.03.08.MAP(PART)
11.03.08.LDSET(PRESET=ZERO)
11.03.08.LGO.
11.03.09.CPU ERROR EXIT AT 004256.
11.03.09.CM OUT OF RANGE.

```

Figure 14-5. Example 1: Dayfile from a Job Run

The exchange package dump that results from the job step abort is shown in figure 14-6. The address in register A4 is 7500, the same value as that given for the job field length (FL). An A register containing an address greater than or equal to the job field length indicates that the job step attempts to reference data outside its field length. A bad address in registers A1 through A5 indicates the error occurs on a read request. A bad address in register A6 or A7 indicates an attempt to write outside the job field length.

The address where the program terminated and the type of error that occurred are given in the dayfile.

```
CPU ERROR EXIT AT 004256.  
CM OUT OF RANGE.
```

This address can also be found in bits 47 through 30 in the RA field of the exchange package. Usually, the program actually terminates at the instruction word preceding the address given, in this case, word 4255. Bits 59 through 48 of the RA contain the value 1, indicating that a mode 1 error occurred (CM OUT OF RANGE).

To find the approximate object program address to which memory address 4256g corresponds, you refer to the load map (figure 14-4). The load map lists each of the blocks loaded, its length, and where it was loaded. If overlays or capsules are used in the program, an area within the field length may contain different program blocks at different times during program execution. In that case, you must also determine which overlay or capsule was in memory at program termination.

Program FPROG has only one program block, FPROG. The remainder of the load is from various system libraries. It is 4166g words long and was loaded at address 111g according to the load map. The relative address where the program terminated is therefore 4255g-111g, or 4144g.

Referring to the symbolic reference map (figure 14-3), you find that relative address 4144g is the address of loop label 10. Loop label 10 is on line 6 of the source program. By examining that loop, you can determine that the execution error occurred when loop index I was set too large.

The symbolic reference map generated by the loader does not provide relative addresses for each line of source code. In many cases, you are not given the exact source code location corresponding to the termination address. You must approximate the termination location as being between two relative addresses given in the reference map.

Example 2 - Finding the Contents of a Program Variable:

You can also use a CM dump to determine the contents of a program variable when program execution terminated. For example, assume that you submit FJOB after correcting the source code and adding the command DMP(5000). The resulting symbolic reference map and load map are identical to those shown in figures 14-3 and 14-4. To determine the contents of the variable NSUM following normal program termination, you must determine its absolute address.

The relative address of NSUM as given in the reference map in figure 14-3 is 4152g. The block containing the object program, FPROG, was loaded at absolute address 111g so the absolute address of NSUM is 111g+4152g, or 4263g.

You then find address 4263g in the memory dump. (Portions of the dump are shown in figure 14-7.) The content of address 4263g is

```
00000 00000 00000 00067
```

Because NSUM is an integer variable, the number is stored in integer format. Converted to base 10, this value is 55 (the sum of the numbers 1 through 10).

EXCHANGE PACKAGE.													
P	0	A0	0	B0	0	(A0)	0001	0042	5600	0000	0000		
RA	375500	A1	0	B1	0	(A1)	0001	0042	5600	0000	0000		
FL	7500	A2	0	B2	0	(A2)	0001	0042	5600	0000	0000		
EM	7007	A3	0	B3	0	(A3)	0001	0042	5600	0000	0000		
RAE	0	A4	7500	B4	0	(A4)	0000	0000	0000	0000	0000		
FLE	0	A5	4263	B5	0	(A5)	0631	6514	3474	6057	1525		
MA	2600	A6	4264	B6	331007	(A6)	0000	0000	0000	0000	0000	0001	
		A7	4263	B7	3214	(A7)	0631	6514	3474	6057	1525		
x0	0000	0000	0000	0000	0000								
x1	0000	0000	0000	0000	0000								
x2	0000	0000	0000	0000	0000								
x3	0000	0000	0000	0000	0000								
x4	0000	0000	0000	0000	0000								
x5	0631	6514	3474	6057	1525								
x6	0000	0000	0000	0000	0001								
x7	0631	6514	3474	6057	1525								
(RA)	0001	0042	5600	0000	0000								
(RA+1)	0000	0000	0000	0000	0000								
CM DUMP FROM 4216 TO 4316.													
4214	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
DUPLICATED LINES.													
4240	00000	00000	00000	00000	00000	20020	00000	00000	04244	00000	00000	00000	11610
4244	11162	02524	00000	00111	00000	17252	42025	24000	02165	00000	00000	00000	00000
4250	04000	04251	00000	00001	00000	43700	71600	00001	46000	51700	04263	51600	04264
4254	63750	46000	46000	46000	00000	51500	04263	51470	04264	61770	00001	36745	54750
4260	51700	04264	51100	04262	00000	04000	04700	46000	46000	00000	00000	00000	00000
4264	00000	00000	00000	00001	00000	00000	00000	00000	00001	00000	00000	00000	00002
4270	00000	00000	00000	00004	00000	00000	00000	00000	00005	00000	00000	00000	00006
4274	00000	00000	00000	00010	00000	00000	00000	00000	00011	00000	00000	00000	00012
4300	55555	55555	55555	55555	00000	40404	04040	04040	04040	11162	02524	00000	00000
4304	17171	27432	14774	13155	00000	20001	20726	42717	30565	00000	00000	00000	00000
4310	00000	00000	00000	11610	00000	00000	00000	00000	00000	00000	00000	00000	00000
4314	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	000024	00000

Figure 14-6. Example 1: Exchange Package Dump

CM DUMP FROM 0 TO 5000.													
0	00000	00000	00000	00000	00000	05160	42000	00000	00000	00000	00000	00000	00000
4	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
DUPLICATED LINES.													
54	56110	03110	00054	54710	00000	51100	00001	03110	00055	64550	02550	00000	46000
60	15051	52000	00000	00061	00000	00000	07500	00000	00001	00000	00000	00000	00000
64	14071	70000	00000	00000	00000	40000	00000	00000	07446	40000	00000	01000	00111
70	14071	75700	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
74	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
100	54000	00000	01000	00001	00000	00733	50000	00000	07446	00000	00000	00000	00000
104	00000	00000	00000	07446	00000	00000	00000	00000	00000	00000	00000	00000	00000
110	06202	21707	00000	04250	00000	11162	02524	00000	00001	00000	00000	00000	00162
114	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
120	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
124	00000	00022	00000	00000	00000	00000	00000	00000	00000	14000	00000	00000	00000
130	00000	00006	00000	00000	00000	00000	00000	00000	00000	00000	00000	02003	00000
134	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
DUPLICATED LINES.													
2164	00000	00000	00000	00000	00000	17252	42025	24000	00001	00000	00000	00000	02236
2170	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
2174	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
2200	00000	00022	00000	00000	00000	00000	00000	00000	00000	14000	00000	00000	00000
2204	00000	00006	00000	00000	00000	00000	00000	00000	00000	00000	00000	02003	00000
2210	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
DUPLICATED LINES.													
4240	00000	00000	00000	00000	00000	20020	00000	00000	04244	00000	00000	00000	11610
4244	11162	02524	00000	00111	00000	17252	42025	24000	02165	00000	00000	00000	00000
4250	04000	04251	00000	00001	00000	43700	71600	00001	46000	51700	04263	51600	04264
4254	63750	46000	46000	46000	00000	51500	04263	51470	04264	61770	00001	36745	54750
4260	51700	04264	51100	04262	00000	04000	04700	46000	46000	00000	00000	00000	00000
4264	00000	00000	00000	00013	00000	00000	00000	00000	00001	00000	00000	00000	00002
4270	00000	00000	00000	00004	00000	00000	00000	00000	00005	00000	00000	00000	00006
4274	00000	00000	00000	00010	00000	00000	00000	00000	00011	00000	00000	00000	00012
4300	55555	55555	55555	55555	00000	40404	04040	04040	04040	11162	02524	00000	00000

Figure 14-7. Example 2: Central Memory Dump

NOS provides the following utilities for file maintenance.

- EDIT Edits a text file.
- KRONREF Generates a cross-reference listing of system symbols.
- MODIFY Edits a Modify-formatted program library file.
- OPLEDIT Removes modification decks and identifiers from a Modify-formatted program library file.
- PROFILE Enables a master user to update and inquire about a profile file.
- UPDATE Edits an Update-formatted program library file.
- XEDIT Edits a text file.

EDIT COMMAND

The EDIT command calls the Text Editor utility. The Text Editor manipulates data on a specified mass storage file through use of special input directives called edit commands. For a detailed description of the Text Editor and an explanation of these commands, refer to the Text Editor Reference Manual.

Format:

EDIT, lfn₁, m, lfn₂, lfn₃.

or

EDIT, FN=lfn₁, M=m, I=lfn₂, L=lfn₃.

<u>parameter</u>	<u>Description</u>
lfn ₁	Name of file to be edited (referred to as edit file). This specification is required for batch origin jobs.
m	Mode of file processing: <ul style="list-style-type: none"> ASCII or AS ASCII mode edit file. NORMAL or N NORMAL mode edit file.
lfn ₂	File from which directives (edit commands) are to be read. If omitted, INPUT is assumed.
lfn ₃	File to which output is to be written. If omitted, OUTPUT is assumed.

KRONREF COMMAND

The KRONREF command generates a cross-reference listing of system symbols used by decks on a Modify OPL.

Format:

KRONREF,P=lf_{n1},L=lf_{n2},S=lf_{n3},G=lf_{n4}.

<u>Parameter</u>	<u>Description</u>
P=lf _{n1}	OPL input from file lf _{n1} . If the P option is omitted or P alone is specified, file OPL is assumed.
L=lf _{n2}	List output on file lf _{n2} . If the L option is omitted or L alone is specified, file OUTPUT is assumed.
S=lf _{n3}	System text from overlay lf _{n3} . The system text must contain symbol definitions. If the S option is omitted or S alone is specified, file NOSTEXT is assumed. If S=0 is specified, only the common deck references and statistics are listed.
G=lf _{n4}	System text from local file lf _{n4} . The system text must contain symbol definitions. If G is omitted, system text is acquired as specified or defaulted by the S option. If G alone is specified, local file TEXT is used. Use of the G option overrides any S specification.

The names of programs on the OPL are listed for those decks that reference the following.

- PP direct cell locations defined in lf_{n3} or lf_{n4}.
- PP resident entry points defined in lf_{n3}.
- Monitor functions.
- Central memory pointers (in low central memory) defined in lf_{n3} or lf_{n4}.
- Central memory locations (in low central memory) defined in lf_{n3} or lf_{n4}.
- Control point area words defined in lf_{n3} or lf_{n4}.
- Dayfile message options.
- File types and mass storage constants.
- Job origin types, service classes, file types, and priorities.
- Error flags referenced.
- Common deck calls.
- PP packages called.
- Special entry points.

MODIFY COMMAND

The MODIFY command edits a Modify-formatted program library file.

Format:

MODIFY,P₁,P₂,...,P_n.

The following optional parameters can be specified in any order.

<u>P_i</u>	<u>Description</u>
A	Write the compile file in compressed format.
C=lf _{n1}	Write compile output on lf _{n1} . If you specify only C or omit this parameter, lf _{n2} is COMPILER. C=0 results in no compile output.
CB=lf _{n2}	Write binary output on lf _{n2} . This parameter is equivalent to the B=lf _n parameter on a compiler or assembler call command. If you specify only CB or omit this parameter, LGO is the binary output file. CB=0 produces no binary output. This parameter is meaningful only if you specify the Q=processor or X=processor parameter.
CG=lf _{n3}	Load system text from lf _{n3} . This parameter is equivalent to the G=lf _n parameter on a compiler or assembler call command. Load system text from SYSTEXT. CG=0 loads no system text. By default, the system uses the CS=lf _{n5} parameter. You must specify the Q=processor or X=processor parameter if the CL=lf _{n3} is to be meaningful.
CL=lf _{n4}	Write full listing on lf _{n4} . CL=lf _{n5} is equivalent to the L=lf _{n2} parameter on a compiler or assembler call command. If you specify only CL, OUTPUT is the list file. If you omit this parameter or specify CL=0, a short listing is written on OUTPUT. This parameter is meaningful only if you specify the Q=processor or X=processor parameter.
CS=lf _{n5}	Take system text from lf _{n5} . This parameter is equivalent to the S=lf _n parameter on a compiler or assembler call command. If you omit this parameter or specify only CS, the system text is taken from SYSTEXT overlay. CS=0 produces no system text. This parameter is meaningful only if you specify the Q=processor or X=processor parameter.
CV=cv	Convert the library to the character set specified by cv (63 or 64). By default, no conversion occurs.

NOTE

When the CV=63 or CV=64 parameter is specified, Modify prevents compile file generation (C=0).

<u>P_i</u>	<u>Description</u>
D	Do not abort the job on a directive error. This parameter does not affect fatal error processing. By default, the job aborts on a directive or fatal error.
F	Edit all decks on the library and write them on new program library, compile file, and source file if the respective options are selected. By default, the U parameter or EDIT directives determine the decks to be edited.
I=lf _n ₆	Take directives from next record on lf _n ₆ . INPUT is the default. The system reads no directives if you specify I=0.
L=lf _n ₇	Write list output on lf _n ₇ . OUTPUT is the default. L=0 generates no list output file.
LO=c ₁ c ₂ ...c _n	Select list options c _i . In an interactive job, the default is LO=E if the list output file is assigned to the terminal. Otherwise, the default is LO=ECTMWD. You can specify a maximum of seven list options c _i .

<u>c_i</u>	<u>Significance</u>
A	Active lines in deck.
C	Directives other than INSERT, DELETE, RESTORE, MODNAME, I, or D.
D	Deck status.
E	Errors.
I	Inactive lines in deck.
M	Modifications performed.
S	Statistics.
T	Text input.
W	Compile file directives.

Example: LO=EMS

N=lf _n _g	Write new program library on lf _n _g . If you specify only the keyword N, the system uses file NPL. If you omit this parameter or specify N=0, the system generates no new program library.
--------------------------------	--

NOTE

If a new program library is being generated, and the system performs an evict operation upon the file specified by the N=lf_n_g parameter.

<u>Pi</u>	<u>Description</u>
NR	Do not rewind compile file. By default, the system rewinds compile file at beginning and end of Modify run.
P=lfng	Take program library input from lfng. If you do not specify a file, the system takes the input from OPL. P=0 results in no input file for the program library.
Q=processor	<p>Set LO=E and the A parameter at the beginning of the Modify run and call the assembler or compiler specified by processor at the end of the run. The system does not rewind the directives file or the list output file. If you do not specify this parameter or specify Q=0, no call is made to an assembler or compiler.</p> <p>If you specify only the keyword Q, the system sets LO=E and the A parameter at the beginning of the Modify run and calls the COMPASS assembler at the end of the run. When this option is selected, the CB=lfng₂, CL=lfng₄, CS=lfng₅, and CG=lfng₃ parameters are meaningful. Compiler input is assumed to be COMPILE. All other parameters are set by default. If CL is not specified with Q, lines beginning with an asterisk in column 1 are not written to the compile file (compile file directives are processed).</p>
S=lfng ₁₀	<p>Write source output on file lfng₁₀. If you specify only the keyword S, the file SOURCE is used. If you omit this parameter or specify S=0, the system generates no source output file.</p> <p>Do not specify this parameter if you specify the A, Q=processor, or X=processor parameter.</p>
U	Edit only decks for which the directives file contains deck directives and write them on the compile file, new program library, and source file (if their respective parameters are selected). F, if specified, takes precedence. By default, decks to be edited are determined by EDIT directives or by the F parameter.
X	Process the same as Q=processor parameter, except that modify directives input (I parameter) and list output (L parameter) files are rewound before processing.
Z	Read the input directives following the terminator of the MODIFY command; the input file is not read. This eliminates the need for a separate input file for the directives when only a few directives are needed. The first character following the command terminator is the separator character for all directives on the command. Any display code character that is not used in any of the directives, including a space, can be used as the separator character. The directives can extend to column 72 on the command. Continuation lines are not permitted. If Z is omitted, the command does not contain the input directives.

NOTE

Do not place another terminator after the directives.

For a detailed description of the Modify utility, refer to the Modify Reference Manual.

OPLEDIT COMMAND

The OPLEDIT command removes modification decks and identifiers from a Modify-formatted program library file.

Format:

OPLEDIT, P₁, P₂, ..., P_n.

<u>P_i</u>	<u>Description</u>
I=lf _{n1}	Use directive input from file lf _{n1} . Take input directives from file lf _{n1} . INPUT is the default. I=0 indicates that there are no input directives.
P=lf _{n2}	Use file lf _{n2} for the old program library. Use file lf _{n2} for the old program library. OPL is the default. P=0 indicates that there is no old program library.
N=lf _{n3}	Write new program library on file lf _{n3} . Write the new program library on lf _{n3} . If you specify only the keyword N, NPL is used. If you omit this parameter or specify N=0, the system writes no new program library.
L=lf _{n4}	List output on file lf _{n4} . Place list output on file lf _{n4} . OUTPUT is the default. L=0 generates no list output.
M=lf _{n5}	Write output from *PULLMOD directives on file lf _{n5} . If this option is omitted, M=MODSETS is assumed.
LO=c ₁ c ₂ ...c _n	Selects up to five list options c _i .

<u>c_i</u>	<u>Description</u>
E	Errors.
C	Input directives.
M	Modifications made.

Pi

Description

ci Description

D Deck status.

S Statistics.

If this parameter is omitted or just L0 is specified, it selects all list options. If the output file is assigned to the terminal in a interactive job and this parameter is omitted or just L0 is specified, list option E is selected.

F Modify all decks.

D Debug; ignore errors.

U Generate *EDIT directives for all decks; meaningful only for *PULLMOD executions.

U=0 Generate no *EDIT directives. If the U option is omitted, generate *EDIT directives for common decks. This is meaningful only for *PULLMOD execution.

Z Read the input directives that follow the terminator. The input file is not read. This eliminates the need to use a separate input file for the directives when only a few directives are needed. The first character following the command terminator is the separator character. If Z is omitted, the command does not contain the input directives.

NOTE

Do not place another terminator after the directives.

For a complete description of the OPLEDIT utility, refer to the Modify Reference Manual.

PROFILE COMMAND

The PROFILE command enables the master user to update and inquire about a project profile file for user profile control. Other capabilities of PROFILE (available only to system origin jobs) are described in the NOS 2 System Maintenance Reference Manual.

Format:

PROFILE, P₁, P₂, ..., P_n.

The following optional parameters can be specified in any order.

<u>Pi</u>	<u>Meaning</u>
I=lf _{n1}	File lf _{n1} contains input directives for an update (OP=U). If omitted, file INPUT is assumed.
L=lf _{n2}	File lf _{n2} receives output listings. If omitted, OUTPUT is assumed.
P=lf _{n3}	File lf _{n3} is the project profile file. If omitted, file PROFILB is assumed.
CN=cn	Charge number inquiry. All project numbers valid for charge number cn are written to output file. Valid only if OP=I option specified.
PN=pn	Project number inquiry. The control values and all valid user names for project number pn are written to the output file. The OP=I and CN=cn options must also be specified to use the PN option.
CV	Convert option. Specifies that directives on the input file are in NOS 1.0 or 1.1 format and are to be converted to update a profile file for a later version of NOS. Obsolete directives are ignored. OP=U or OP=T option must also be specified.
OP=opt	PROFILE processing option. One of the following:

<u>opt</u>	<u>Description</u>
L	Lists portions of the profile file as specified by the LO=l _p parameter.
U	Updates the project profile file with directives supplied by the input file. U is the default value for the master user if the OP option is omitted.
T	Interactive update. Processing is the same as OP=U, but preliminary instructions are suppressed.
I	Inquire option. Output is dependent upon CN and/or PN=pn parameters specified.

<u>Pi</u>	<u>Meaning</u>
LO=op	PROFILE list option (OP=L must be specified). One of the following:

<u>op</u>	<u>Description</u>
FM	Full list of everything accessible on the profile file by the master user. This is the default for a nonsystem origin job if the LO=op parameter is omitted.
CM	Charge number list of all charge numbers accessible on the profile file by the master user.
PM	Project number list of all project numbers accessible on the profile file by the master user.

Directives are available to the master user as input to PROFILE to add or update information concerning each charge number. The input file for a PROFILE update (OP=U) is divided into groups of entries that each begin with a charge number directive. Each directive following a particular charge number entry applies only to that charge number, until another charge number entry occurs.

Each line of the input file can contain one or more directives (up to 72 characters per line) in the following format.

dir₁,dir₂,...,dir_n

Each directive is separated by a delimiter which can be any special character (display code greater than 44g) except the following:

/ + - * :

An end-of-line or end-of-card also delimits directives. The following directives are available to the master user for PROFILE input.

<u>dir_i</u>	<u>Description</u>
/cn	Specifies the charge number cn to which the following directives apply. If this form of charge number specification is used (refer to CN=cn directive), it must begin in column 1 with the slant (/).
APN=pn	Adds or activates project number pn.
AUN=un	Adds user names un (must be preceded by PN directive).
CN=cn	Same as /cn except that it can begin in any column.
DPN=pn	Deactivates project number pn. This directive does not delete the specified project number entry but sets its status such that it cannot be specified by users.
DUN=un	Deletes user name un from the list of those who may access the project number (must be preceded by PN directive).
ISV=x	Sets x as the maximum SRU accumulation for any job using the charge number and project number specified by preceding CN and PN directives.

<u>dir_i</u>	<u>Description</u>
PEX=yymmdd.	Specifies expiration date for project number of preceding PN directive. If PEX=0 is entered, the project number is not limited by an expiration date.
PN=pn	Project number for which the following directives (until the next PN directive) apply.
SMA=acc	Sets the current number of accumulated SRUs the project number has used (PN directive required). This accumulator is updated at the end of a job or terminal session and each time a CHARGE command is entered. When the SMA value surpasses the SML value, the project is not available to users until either the limit or accumulator is respecified.
SML=lim	Specifies the maximum number of accumulated SRUs the project (PN directive) may use. SML=0 implies no restriction.
TI=ti	Specifies the time of day before which the project number specified by the PN directive cannot be used. The time is specified in 24-hour clock notation. For example, a ti specification of 1315 indicates the project number cannot be used before 1:15 in the afternoon.
TO=to	Specifies the time of day after which the project number specified by the PN directive cannot be used. The time is specified in 24-hour clock notation. For example, a to specification of 1315 indicates that the project number cannot be used after 1:15 in the afternoon.

UPDATE COMMAND

The UPDATE command creates, edits, or copies an Update-formatted program library file. For complete information on Update directives and processing, refer to the Update Reference Manual.

Format:

UPDATE,p₁,p₂,...,p_n.

The parameters are optional and can be specified in any order.

For file-specifying parameters (those associated with the keywords C, G, I, K, M, N, O, P, S, or T), you can append either the digit 6 or 8 (but not both) to the keyword. Each file-specifying parameter has its own default for the character code set of the file (refer to Character Sets in appendix A). Except for M and P, the digit 6 or 8 overrides this default. The digit 6 directs the system to represent the file in the display code set. The digit 8 directs the system to represent the file in the ASCII code set.† However, the system uses the ASCII code set only if the digit 8 is appended to the keyword and if the input file is also represented in the ASCII code set. For example, C8=FILE directs the system to write the decks on the compile file FILE in the ASCII code set.

†The digit 8 directs the system to represent the file in the 6/12 display code set if the job is interactive and the file is an input or output file.

<u>Pi</u>	<u>Description</u>
A	Copies the old sequential access program library to a new random access program library.
B	Copies the old random access program library to a new sequential access program library.
C=lf _{n1}	Specifies the compile file. Unless otherwise specified, the compile file has the same character code set as the old program library. C=PUNCH implies the D and 8 parameters. C=0 implies no compile file. If you omit this parameter or specify the keyword only, COMPILE becomes the compile file. If you specify the K=lf _{n1} parameter, the C=lf _{n1} parameter is ignored.
D	Sets the width of the compile file to 80 characters. The default is 72.
E	Edits the old program library. To completely edit the library, specify the E parameter on two UPDATE commands. The first occurrence of the E parameter causes the system to rearrange the directory to reflect the actual order of the decks on the library and to remove previously purged identifiers. The second occurrence of the E parameter causes the system to remove the identifiers that exist only as directory entries.
F	Specifies full Update mode.
G=lf _{n2}	Specifies the output file for the PULLMOD directive. Unless otherwise specified, the character code set of lf _{n2} is the same as that of the old program library. If you do not specify lf _{n2} , the output from the PULLMOD directive is appended to the source file (specified by the S=lfng parameter).
H=n	Specifies whether a 63- or 64-character character set is to be used in the Update run.

<u>n</u>	<u>Description</u>
3	A 63-character character set.
4	A 64-character character set.
omitted	Character set that was used in the old program library.
I=lf _{n3}	Specifies the primary input file. The default is INPUT. The default character code set is ASCII code if INPUT is associated with the terminal. Otherwise, it is display code.

<u>P_i</u>	<u>Description</u>
K=lf _n ₁	Specifies the compile file and specifies that the decks are written to that file in their order of appearance in the COMPILER directives. The default is file COMPILER. The default character code set is display code.

L=c₁c₂...c_n Specifies one or more of the following list options.

<u>c_i</u>	<u>List Options</u>
A	Lists deck names and correction set identifiers, COMDECK directives, definitions, and decks written on the compile file.
F	Uses all list options except 0.
0	Suppresses all listing.
1	Lists lines in error.
2	Lists active Update directives.
3	Notes on each line that changed status during current execution.
4	Lists text lines.
5	Lists active compile file directives.
6	Lists active and inactive lines.
7	Lists all active lines.
8	Lists all inactive lines.
9	Lists correction history of all lines selected by list options 5, 7, and 8.

The default depends on the nature of the Update processing. For a creation run, the default is options A, 1, and 2. For a correction run, the default is A, 1, 2, 3, and 4. For a copy run, the default is A and 1.

M=lf_n₄ Merges lf_n₄ and the old program library. If you specify only the keyword M, the system merges the file MERGE and the old program library. The two libraries must have the same character code set. If the parameter is omitted, no merging occurs.

N=lf_n₅ Specifies the new program library. If you do specify a new program library on a correction run, no new program library is generated. Otherwise, the default is the file NEWPL. The default character code of the new program library is the same as that of the old program library unless the primary input file is in ASCII code and the old program library is in display code. In this case, the default is the ASCII code set.

<u>P_i</u>	<u>Description</u>										
O=lf _n ₆	Specifies the output file. The default is OUTPUT. The default character code is the same as that of the old program library.										
P=lf _n ₇ /s ₁ /s ₂ /.../s ₇	Specifies file lf _n ₇ as the old program library. If just the keyword P is specified, file OLDPL is assumed. If s ₁ /s ₂ /.../s ₇ is specified, secondary old program libraries will reside on files s ₁ ,s ₂ ,...,s ₇ .										
Q	Processes only decks on COMPILE directives (quick mode).										
R=c ₁ c ₂ ...c ₄	Rewind specified files before and after the Update run.										
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>c_i</u></th> <th style="text-align: left;"><u>Files</u></th> </tr> </thead> <tbody> <tr> <td>C</td> <td>Compile file.</td> </tr> <tr> <td>N</td> <td>New program library.</td> </tr> <tr> <td>P</td> <td>Old program library and merge library.</td> </tr> <tr> <td>S</td> <td>Source and PULLMOD files.</td> </tr> </tbody> </table>	<u>c_i</u>	<u>Files</u>	C	Compile file.	N	New program library.	P	Old program library and merge library.	S	Source and PULLMOD files.
<u>c_i</u>	<u>Files</u>										
C	Compile file.										
N	New program library.										
P	Old program library and merge library.										
S	Source and PULLMOD files.										
	If c _i is omitted, no files are rewound. If R=c ₁ c ₂ ...c ₄ is omitted, the old program library, new program library, compile file, source file, and PULLMOD file are rewound.										
S=lf _n _g	Specifies the name of the source file. If you specify the keyword only, the file SOURCE becomes the source file. If you omit the parameter, the system does not write a source file unless you specify the T=lf _n _g parameter. The default character code set of the source file is the same as that of the old program library.										
T=lf _n _g	Functions the same as the S=lf _n _g parameter, except that it excludes common decks from the source file. This parameter takes precedence over the S=lf _n _g parameter.										
U	Specifies that Update processing is to continue even if normally fatal errors occur.										
W	The new program library is written in sequential access format. If W is omitted, the new program library is written in random access format unless the file is on magnetic tape.										
X	Writes the compile file in compressed format. If X is omitted, the compile file is not compressed.										
8	Writes compile file output as 80-character line images. If 8 is omitted, the compile file is written as 90-character line images.										

<u>Pi</u>	<u>Description</u>
*=char	The master control character (first character of each directive) for this Update run is char, which can be any character having a display code octal value in the range 01 through 54 except for 51 and 52 (the open and close parentheses). If this option is omitted, the master control character is *.
/=char	The comment control character for this Update run is char, which can be A through Z, 0 through 9, or +-*/\$=. The character should not be changed to one of the abbreviated forms of the directives unless NOABBREV is in effect. If this option is omitted, the comment control character is a slant bar.

XEDIT COMMAND

The XEDIT command calls the text editor, XEDIT. For a complete description of XEDIT parameters and directives, refer to the XEDIT Reference Manual.

Format:

XEDIT, lfn₁, p₁, p₂, ..., p_n.dds

All parameters are optional. The following are brief parameter descriptions.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Use lfn ₁ as the name of the file to be edited or created. If lfn ₁ is omitted (indicated by two separators before other parameters), the primary file is edited.
<u>pi</u>	<u>Optional Parameters</u>
	<u>pi</u>
	<u>Description</u>
AS	Edit lfn ₁ in ASCII interactive mode. After the XEDIT job step, processing returns to the original mode.
B	Process the job as a batch origin job.
C	Create a new file lfn ₁ .
FR	Take the first editing directive from the first line of lfn ₁ .
I=lfn ₂	Take editing directives from lfn ₂ . By default, directives are taken from file INPUT. If you specify I=0, the system takes all directives from the trailing delimited directive sequence. (If the FR parameter is also specified, the system processes the delimited directive sequence after processing directives from the first line of lfn ₁ .)
L=lfn ₃	Put all XEDIT output on the specified file. OUTPUT is the default. If you specify L=0, the system suppresses all output.
NH	Suppress printing of the XEDIT header.

Parameter

Description

pi

Description

P Retrieve and edit permanent file lfn₁. Direct access files are attached in write mode. If P is omitted, the file lfn₁ is assumed to be a local file.

dds

Process the directives dds that follow the command terminator. The first character that follows the terminator delimits the directives. The value dds has the following form (the semicolon serve as the delimiter here):

;directive₁;directive₂;...;directive_n

Do not place a terminator after the last directive.

A library is a file containing records that are accessed individually. Library records can be of several types and can be accessed randomly or sequentially.

This section describes file access methods, library record types, and the following commands and their functions.

<u>Command</u>	<u>Function</u>
CATALOG	Describes the records on a file.
COPYL and COPYLM	Copies an old file to a new file, updating the records from a replacement file.
GTR	Appends records selected from one file to another file.
ITEMIZE	Describes the records on a file.
LIBEDIT	Generates a file containing records from one or more other files. The records may be of several types. LIBEDIT handles a LIBGEN-generated library as a single record.
LIBGEN	Generates a user library; that is, a library of relocatable, capsule, absolute, CP overlay, or procedure file records that can be accessed by CYBER Loader.
VFYLIB	Compares the records in two files and lists their differences.

If you use these commands on files structured by CYBER Record Manager (CRM), the output may not reflect the CRM file structure. In the case of S- or U- type records, the CRM file structure is preserved. Otherwise, the resultant NOS records will not correspond to CRM records (refer to File Structure in section 2).

LIBRARY FILE ACCESS METHODS

The methods used to access records within NOS files are sequential access and random access.

To access a record sequentially, NOS rewinds the file to BOI and then reads records until it finds the requested one. To replace, insert, or delete records from a sequential access file, NOS must rewrite the entire file. (Records can be appended at EOI without rewriting the file.)

To access a record randomly, the file must be on mass storage and must have a directory containing the address of each record (figure 16-1). The GTR, LIBGEN, and LIBEDIT commands can generate random access directories. Records within a file can then be replaced, inserted, or deleted by rewriting the directory instead of rewriting the entire file. Records within a random access file can also be accessed sequentially.

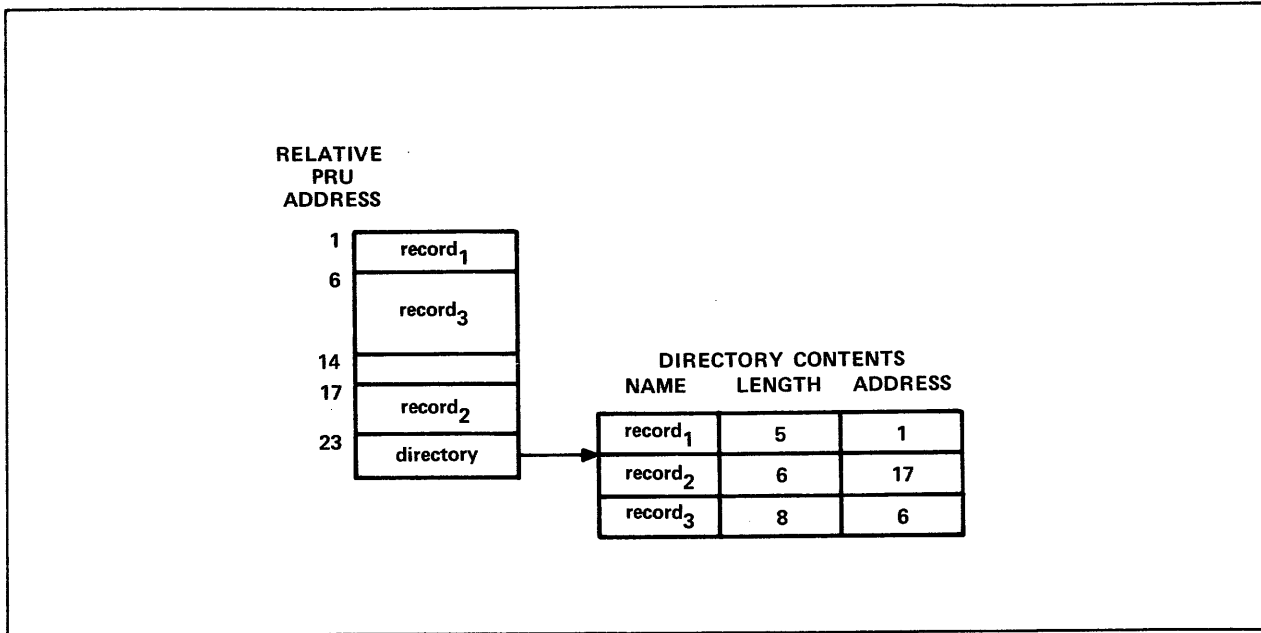


Figure 16-1. Random Access File Structure

LIBRARY RECORD TYPES

NOS determines the type and the name of a record from information contained within the record. If the record begins with a prefix table, the record name (if any) is obtained from that table and the type of the record is determined from the first word following the prefix table. Otherwise, the first word of the record determines the type and name (if any) of the record.

Prefix tables exist, unless they have been specifically suppressed, for:

- Assembled or compiled programs.
- System text overlays.
- User library header records and directory records.
- Modify program library decks, common decks, or directory records.

The prefix table consists of a header word with an octal table type identifier followed by varying amounts of control information.

The prefix table is identified by octal digits 7700 in bits 59 through 48 of its first word; consequently, it is often referred to as a 77 or 7700 table. Information in the prefix table, which originates with the assembler, compiler, or other program that creates the table, can specify items such as the date created and the system on which the job was processed (refer to CYBER Loader Version 1 Reference Manual).

If a prefix table is not present, the first word in a record is examined. If a record meets the criteria for a given type of record, the utilities identify it as such. For instance, a load file beginning with a Job command may be identified as type TEXT, depending on the particular characters in the Job command.

Although some of the records may contain display coded data (loader directives, for instance, are coded), they are considered binary records.

The following record types can be specified on the GTR and LIBEDIT commands and are recognized by the CATALOG and VFYLIB commands. You determine the record types contained in a file by issuing a CATALOG command naming the file. VFYLIB lists the differences between the record types of two files. LIBGEN generates a user library from relocatable (REL) capsule (CAP), absolute (ABS), CP overlay (OVL), or procedure file (PROC) records.

<u>Mnemonic</u>	<u>Meaning</u>
ABS	Multiple entry point overlay.
CAP	CYBER Loader capsule.
OPL	Old program library deck created by Modify.
OPLC	Old program library common deck created by Modify.
OPLD	Old program library directory created by Modify.
OVL	Central processor overlay.
PP	Peripheral processor program.
PPU	Peripheral processor unit program.
PROC	Procedure file.
REL	Relocatable central processor program.
TEXT	Unrecognizable as a program.
ULIB	User library directory.

The following record types are recognized by COPYL, COPYLM, and ITEMIZE commands. Record types ACF, OPLD, UCF, ULIB, and UPLx are recognized by the ITEMIZE command and are not recognized by COPYL or COPYLM.

<u>Mnemonic</u>	<u>Meaning</u>
ABS	Multiple entry point overlay.
ACF	Modify compressed compile file. It has no record name. This record type is determined by a nonzero value in bits 17 through 0 in the second word of the 77 table.
CAP	CYBER Loader capsule.
OPL	Old program library deck created by Modify.
OPLC	Old program library common deck created by Modify.

<u>Mnemonic</u>	<u>Meaning</u>
OPLD	Old program library directory created by Modify.
OVL	Central processor overlay.
PROC	Procedure file.
REL	Relocatable user program.
TEXT	Unrecognizable as a program.
UCF	Update compressed compile file. It has no record name. This record type is determined by a 77 table with a zero word count.
ULIB	User library directory.
UPLx	Update sequential program with x master control character. It has no record name. This record type is determined by no 77 table and the characters CHECK in bits 59 through 30; control characters are obtained from bits 5 through 0.
6PP	Peripheral processor program.
7PP	Peripheral processor unit program.

Appendix G in Volume 4, Program Interface, contains the formats of PP (6PP), OPL, OPLD, ULIB, and TEXT records. (The OPLC record format is identical to that of the OPL record.) The Modify Reference Manual describes how to create ACF, OPL, OPLC, and OPLD records. The CYBER Loader Reference Manual contains the formats of ABS, CAP, OVL, PP, and REL records. The CYBER Loader Instant contains the PPU (7PP) record format. The Update Reference Manual contains the formats of UCF and UPLx records. Section 4 describes the creation of procedures (PROC).

CATALOG COMMAND

The CATALOG command lists information about each record in a file assigned to the job. The record types recognized by CATALOG are listed in Library Record Types in this section.

NOTE

CATALOG produces unpredictable results when attempting to catalog an S, L, or F format tape. Use the COPY command to convert the S, L, or F format tape to a mass storage file or to an I or SI binary format tape before attempting to catalog the file.

Format:

CATALOG, lfn, p1, p2, ..., pn.

lfn Specifies the file to be cataloged.

p_i may be any of the following:

<u>P_i</u>	<u>Description</u>
N=n	Catalogs n files. If n is 0, the system catalogs the file until an empty file is encountered (two consecutive EOFs). If you specify only the keyword N, the system catalogs until it encounters the EOI. The default is N=1.
L=fname	Specifies the name of the output file; if L=fname is omitted, CATALOG assumes L=OUTPUT.
T	Lists the entire text record if the record name begins with any of the following characters: APRO IPRDC CMRDC IPRDECK CMRDECK LIBDC DDSDC LIBDECK DDSDECK If the text record name begins with OVERLAY, the system always lists the first line of the record. If you omit T, the system does not list the contents of the text record.
U	Lists records within a user library; if U is omitted, only the ULIB record within the user library is listed.
D	Suppresses the comments field; suppresses all page headings after the initial page heading for each file.
CS	Suppresses character set indicator (63 or 64) for OPL and OPLC records.
R	Rewinds lfn before and after cataloging; if R is omitted, lfn is not rewound.

The listing for each file of a multifile file begins on a new page with a page heading for that file. If the D option has been specified, the page heading appears only once, at the beginning of the file. The information listed under each heading is as follows:

<u>Heading</u>	<u>Information</u>
REC	Record number (zero-length records and EOF marks are counted).
NAME	Record name [the contents of the name field from the second word of the prefix (77) table, if present; otherwise, the first seven characters of the record]. For ABS and REL records, the entry points are listed beneath the record name.

<u>Heading</u>	<u>Information</u>
TYPE	Record type (refer to Library Record Types in this section).
LENGTH	Record length in words (less prefix table length) printed as an octal number.
CKSUM	A checksum [a value used to verify that the contents of a record (excluding the prefix table) were copied correctly].
DATE	Record creation date (taken from the third word of the prefix table, if present).
COMMENTS	Comments are taken from the comments field of the prefix table. A zero byte or a COPYRIGHT comment in the comments field of the table terminates the listing. This field is not shown when CATALOG Output is assigned to a terminal file.)

CATALOG lists additional information depending on the record type. Entry points are listed for REL and ABS records. The character set used, correction identifiers, and their YANK status (refer to the Modify Reference Manual) are listed for OPL and OPLC records.

A ULIB record suppresses listing of the other records in the user library unless the U parameter is specified on the command.

If an OPLD record in the user library is not encountered before an EOF or EOI within the cataloged file, the following message is output before the *EOF* or *EOI* line.

OPLD MISSING

When a zero-length record is encountered, the length since the last zero-length record is given. If an EOR does not precede an EOF or EOI within the cataloged file, the following message is output before the *EOF* or *EOI* line.

EOR MISSING

The ITEMIZE command is similar to the CATALOG command, but ITEMIZE recognizes additional record types (refer to Library Record Types, earlier in this section).

Example:

Compilation of the FORTRAN program SUBROUT and its subroutines SUB1, SUB2, and SUB3 wrote relocatable object code on file LGO. The following is a catalog of file LGO (refer to the heading definitions given earlier).

REC	CATALOG OF LGO NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
1	SUBROUT SUBROUT	REL	44	6215	yy/mm/dd.
2	A	REL	41	7424	yy/mm/dd.
3	B	REL	41	0465	yy/mm/dd.
4	C	REL	41	5525	yy/mm/dd.
5	* EOF *	SUM =	207		

COPYL AND COPYLM COMMANDS

The COPYL and COPYLM commands copy an old file to a new file, substituting records from a replacement file for the matching records on the old file. Records on the replacement file which do not match records on the old file are either ignored or appended to the new file according to your parameter specifications. Records are considered to match if they have the same type and same name. However, you may specify that the record type be ignored. COPYL and COPYLM are commonly used to maintain files of procedures or relocatable records.

COPYL and COPYLM differ only in the handling of multiple occurrences of a record on the old file. COPYL uses each record on the replacement file only once, replacing the first matching record from the old file. COPYLM uses the first matching record encountered on the replacement file to replace each matching record from the old file. COPYL can be used to replace multiple occurrences of the same record if multiple occurrences of the record are in the replacement file.

The old file and the replacement file must reside on mass storage or an I or SI format tape. Only a single file terminated by an end-of-file marker is processed by a single call to COPYL or COPYLM unless you request processing to the end-of-information by using the E processing option. When working with multifile files, you must be sure to position the multifile file to the file that is to be processed.

The order of the records on the replacement file is not significant. The system copies the records to the new file in the same order as they are on the old file.

COPYL and COPYLM issue dayfile messages during processing; no other printed output is produced. The dayfile messages list which replacement records were copied and which replacement records were not copied to the new file.

COPYL and COPYLM replace only the types of records listed in Library Record Types, earlier in this section. Any record on the old file that is not recognized as one of the listed types is copied to the new file without further processing. Any replacement file record type that is not listed in Library Record Type, is ignored without comment.

Format:

COPYL,oldlfn,replfn,newlfn,last,flag. Single replacement.

or

COPYLM,oldlfn,replfn,newlfn,last,flag. Multiple replacement.

All parameters are optional and order-dependent. You denote an omitted parameter by consecutive commas.

<u>Parameter</u>	<u>Description</u>
oldlfn	File name of the old file; default name is OLD.
replfn	File name of the replacement file; default name is LGO.
newlfn	File name of the updated file; default name is NEW.
last	Name of the last record on oldlfn to be processed. If last is not specified, all records on oldlfn are processed from its current position to the next EOF (or EOI if the E processing option is used).
flag	Processing options.

<u>flag</u>	<u>Description</u>
R	Rewind oldlfn and newlfn files before processing. (The replfn file is always rewound before and after processing. oldlfn and newlfn are not necessarily rewound to beginning-of-information in multifile files. Refer to explanation below.)
A	Append to the end of newlfn all replfn records that do not match any records on the oldlfn. If A is not selected, records on the replacement file that do not match any records on the oldlfn are ignored and a dayfile message is issued.
T	Check for matching name of record, but omit check for matching type of record. If T is not selected, records match only if both the type and name of the records are the same.
E	Process oldlfn until the end-of-information.

These options can be specified by combining one or more letters in any order, such as TRA, AR, ERTA, or TR.

COPYL and COPYLM check only the first four flag options; if more than four are specified, the remaining characters are ignored.

The R option affects file positioning of the old and new files before processing. If R is specified, the old and new files are rewound before processing. In a multifile file, if there is one or more end-of-file markers between the current position of the file and the beginning-of-information, the R option rewinds the file to the first preceding EOF. In the absence of R, you are responsible for positioning the oldlfn and newlfn files. The R option does not affect the file of replacement records, since the current file of the replacement file always is rewound to the beginning-of-information before and after processing.

The E option causes the old file to be processed to the end-of-information. Each end-of-file encountered on the old file causes a matching end-of-file to be written on the new file. Records that are added to the new file as a result of combining the A and E options are appended with an end-of-file prior to the end-of-information. In this case, such appended records will follow an end-of-file if both end-of-file and end-of-information existed at the end of the old file.

Processing stops after an end-of-file, end-of-record, or end-of-information is reached, depending on the structure of the old file and the processing options selected. If processing stopped because end-of-file or end-of-record was reached, the old file is positioned after that EOF or EOR. If processing stopped because end-of-information was reached, the old file is positioned just prior to the end-of-information.

COPYL and COPYLM add an end-of-file to the new file even if no end-of-file is encountered on the old file. No further positioning of the new file takes place.

Example:

The following COPYL command updates OLDFILE with replacement records from REPFIL and writes them on NEWFILE. The A option appends to the end of NEWFILE any records in REPFIL that do not match a record on OLDFILE.

```
COPYL,OLDFILE,REPFIL,NEWFILE,,A.
```

The following dayfile segment shows that COPYL updated the record named B and appended the record named C.

```
08.53.23.COPYL,OLDFILE,REPFIL,NEWFILE,,A.
08.53.24. UPDATED -- PROC / B
08.53.24.APPENDED -- PROC / C
08.53.24. COPYL COMPLETE.
```

The ITEMIZE command show the record information for all records in OLDFILE, REPFIL, and NEWFILE.

```
/itemize,oldfile
```

1	REC	ITEMIZE OF OLDFILE NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
	1	A	PROC	4	3310	
	2	B	PROC	4	5306	
	3	(00)	SUM =	10		
		* EOI *	SUM =	10		

```
ITEMIZE COMPLETE.
```

```
/itemize,repfile
```

1	REC	ITEMIZE OF REPFIL NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
	1	B	PROC	4		
	2	C	PROC	4		
		* EOI *	SUM =	10		

```
ITEMIZE COMPLETE.
```

```
/itemize,newfile
```

1	REC	ITEMIZE OF NEWFILE NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
	1	A	PROC	4	3310	
	2	B	PROC	4	1535	
	3		SUM =	10		
	4	C	PROC	4	7304	
	5	* EOF *	SUM =	14		

```
ITEMIZE COMPLETE.
```


GTR COMMAND

The GTR command appends records selected from one file to the end of another file. The records are selected according to directives specifying their type and name. (Refer to Library Record Types in this section for the list of valid record types.) Records can be accessed randomly (default if a directory exists) or sequentially. If specified, a random access directory is appended to the changed file. The system writes an EOF after the directory..

Format:

GTR,lfn₁,lfn₂,d,NR,S,NA.directive₁,directive₂,...,directive_n

NOTE

The parameters must be in the order shown. GTR identifies its parameters by their position, not by keywords. An omitted parameter is denoted by consecutive commas. Do not place blanks between the terminator (right parenthesis or period) and directive₁. Also, do not place a terminator after directive_n.

The parameters are defined as follows:

<u>Parameter</u>	<u>Description</u>
lfn ₁	File which is searched for the requested records; if lfn ₁ is omitted, file OLD is assumed. lfn ₁ is always rewound before the GTR operation.
lfn ₂	File on which the selected records are written; if lfn ₂ is omitted, file LGO is assumed. GTR always positions lfn ₂ at EOI before copying the selected records.
d	Random access directory parameter. If d is specified, lfn ₂ must be a mass storage file. GTR cannot append records after a directory.

If lfn₁ has a random access directory and if sequential access (the S parameter) is not specified, the lfn₂ directory record is given the same name as the lfn₁ directory record. Otherwise, the lfn₂ directory record is named lfn₂.

<u>d</u>	<u>Description</u>
omitted	No new random access directory (OPLD) is added to lfn ₂ . If the directives specify the user library record type (ULIB), the first record of the user library (ULIB) is not copied to lfn ₂ , the relocatable records are copied, and the last record (OPLD) is copied without alteration. The system does not write an EOF at the end of the file.

Parameter

Description

<u>d</u>	<u>Description</u>
U	No new random access directory (OPLD) is added to lfn ₂ . If the directives specify the user library record type (ULIB), the first record of the user library (ULIB) is copied without alteration to lfn ₂ along with the relocatable records and the old random access directory (OPLD). The system does not write an EOF at the end of the file.
D or other	Writes a new random access directory (OPLD) at the end of lfn ₂ . If the directives specify the user library record type (ULIB), the first record of the user library (ULIB) is copied without alteration to lfn ₂ along with the relocatable records and the old random access directory (OPLD). The system writes an EOF after the new directory.

NOTE

The new random access directory contains entries only for records specified on the current GTR command. To build a new random access directory which contains entries for all records on lfn₂, use LIBEDIT.

NR	No rewind option. lfn ₁ is not rewound after the operation; lfn ₂ is not rewound before or after the operation. If lfn ₁ has a directory, the directory is copied to lfn ₂ . If nr is omitted, both files are rewound before and after the operation.
S	lfn ₁ is searched sequentially; no attempt is made to read a directory.
NA	No abort option. If specified, GTR does not search for an EXIT command when an error occurs. It issues a dayfile message for the error and continues GTR processing at the next directive.
directive _i	Specifies a record or group of records to be retrieved. One or more of the following formats can be used. Valid record types are listed under Library Record Types in this section. The default type is the last type specified on a directive, or if none specified, TEXT. The record name is the first seven characters of the record, or if a prefix table is present, the contents of the name field in its second word.

<u>Directive</u>	<u>Meaning</u>
type/name	Record with the specified type and name.
name	Record with the specified name and the default type.

<u>Parameter</u>	<u>Directive</u>	<u>Description</u>	<u>Meaning</u>
	type ₁ /name ₁ -type ₂ /name ₂	Group of records beginning with name ₁ of type ₁ and ending with name ₂ of type ₂ .	
	type ₁ /name ₁ -name ₂	Group of records beginning with name ₁ of type ₁ and ending with name ₂ of type ₁ .	
	name ₁ -name ₂	Group of records beginning with name ₁ of the default type and ending with name ₂ of the default type.	
	type/name-*	All records of the specified type beginning with the named record.	
	name-*	All records of the default type beginning with the named record.	
	type/*	All records of the specified type.	
	*	All records of the default type.	
	0	A zero-length record is inserted.	

GTR searches file lfn₁ for the records specified by the selection directives. If GTR cannot find a record specified by type and name, it issues the following dayfile message.

GTR ERRORS.

If also issues this message when the record specified is within a user library and when the GTR command syntax is incorrect.

If lfn₁ has a directory (OPLD) record, GTR writes the selected records on lfn₂ in the order specified on the GTR command. If lfn₂ does not have a directory record, GTR writes the selected records in the order that it finds them on lfn₁, rather than in the order specified on the command.

If lfn₂ is on tape, the selected records are copied from the current file position; if lfn₂ is on mass storage, the copy starts at the current EOI of the file. If an EOF exists before the EOI, GTR appends the records as a file following the existing EOF.

Examples:

- GTR,SYSTEM,BIN,D.PP/*

GTR copies all PP records from file SYSTEM to file BIN. It then builds a random access directory and writes it as the last record on BIN.

- GTR,OPL,NEW,,NR.OPLC/COMCARG,0,COMCCIO

GTR retrieves common decks COMCARG and COMCCIO from file OPL. It then writes COMCARG, a zero-length record, and COMCCIO at the current position of file NEW. NEW is not rewound before the operation; OPL and NEW are not rewound following the operation.

- GTR,SYSTEM,SYSLIB,D.ULIB/SYSLIB

GTR copies the user library SYSLIB from file SYSTEM to the end of file SYSLIB.

- GTR.REL/A

GTR retrieves the relocatable record A from file OLD and copies it to file LGO.

ITEMIZE COMMAND

ITEMIZE lists pertinent information about each record of a binary file in a format suitable for printing. Earlier in this section, Library Record Type describes the types of records processed by ITEMIZE.

ITEMIZE processes mass storage files or I or SI format tape files. A file can be processed from beginning-of-information through end-of-information.

Format:

ITEMIZE, lfn, L=listlfn, BL, PW=n, PD, NR, N=n, E.

The first parameter is order-dependent; if lfn is omitted, its position must be indicated by a comma. All the other parameters are optional and order-independent.

<u>Parameter</u>	<u>Description</u>
lfn	Name of file to be itemized; default name is LGO.
L=listlfn	List output on file listlfn; default is L=OUTPUT.
BL	Burstable listing; each file output starts at the top of a page. Default is a compact listing in which a page eject occurs only when the current page is nearly full.
PW=n	Print either 136-character lines or 72-character lines, depending on the value of the decimal integer n. If $n \geq 136$, print 136-character line. If $n < 136$, print 72-character lines. If just PW is specified, print 72-character lines regardless of the listing file device. If PW=n is omitted, the default value is 72-character lines if the listing file is a terminal; otherwise, the default value is 136-character lines.
PD	Print density at eight lines per inch; default is six lines per inch. If this parameter is to produce the desired result, you must ensure that output appears at a printer capable of eight lines per inch.
NR	No rewind of lfn before or after processing; default is rewind before and after processing.
N=n	Itemize n files, where n is a decimal integer; default is N=1. If just N is specified, itemize until end-of-information. If N=0, itemize until an empty file is processed.

<u>Parameter</u>	<u>Description</u>
E	Expand output to list further information; default is no expansion. For record types CAP and REL, list entry points. For types OPL and OPLC, list modification set names and their YANK status. For record types UPL, list correction identifier names.
U	Itemize all records within ULIB type records; default is list only the user library directory.

If both E and U are selected for ULIB type records, all records in the library are itemized; and for REL type records, their entry points are listed.

Output from ITEMIZE is affected by the type of record and parameters selected. A header appears for each file terminated by an end-of-file marker within the file specified by the file name. The first line of the header identifies the file name, file position within that file, and the date and time of the run. It also includes the FI field if the file is a multifile set tape file (refer to the LABEL command). The second line of the header has the following fields:

<u>Field</u>	<u>Description</u>
REC	Position of the record within the file.
NAME	Record name obtained from the second word of the prefix table or from the first word of the record.
TYPE	Type of record. Refer to Library Record Type, earlier in this section, for a description of the record types.
LENGTH	Number of words (octal) in the record, excluding the prefix table.
CKSUM	Logical checksum (octal), excluding the prefix table.
DATE	Date record was created as stored in the prefix table.
COMMENTS	Contents of the comments field in the prefix table.

If no prefix table is present, the associated fields are blank.

Additional information listed depends on the type of record:

<u>Record Type</u>	<u>Description</u>
ABS	Entry point names are listed.
OPL, OPLC, UPL	Deck names are listed.
OVL	Overlay level is listed (octal).
TEXT	Entire record is listed if the name of the record is CMRDC, IPRDECK, IPRDC, LIBDECK, LIBDC, or COMMENT.
6PP	Lists the octal equivalent of the load address.
7PP	Lists the octal equivalent of the load address.

The E parameter can select further details about several types of records (refer to the example that ends the ITEMIZE command description).

The last record in each file is the end-of-file marker, which appears on the listing as the characters *EOF*. The SUM= identification is the total length, in words, for all records in the file, including the prefix table lengths.

Any zero-length record in the file appears with the record name (00). When a zero-length record is encountered, a sum of the lengths of the records encountered since the beginning of the file, or since the last sum was taken, is listed on the output. The length includes prefix tables. Record numbering is not restarted until a new file is encountered.

If a record of type UPL has more correction identifier names and/or deck names than can be accommodated within ITEMIZE, the following message appears on the listing in place of the excess names:

```
TRUNCATED--IDENT OR DECK LIST TOO LONG
```

In this instance, the Update utility must be used to obtain a complete list of identifiers and deck names.

A dayfile message is issued when ITEMIZE completes execution.

Example:

A FORTRAN program named SUBROUT has three subroutines. The following ITEMIZE command lists the records of the binary object file LGO. The E option lists the entry point for each REL type record listed.

```
/itemize,lgo,e
```

1	REC	ITEMIZE OF LGO NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
	1	SUBROUT	REL SUBROUT	105	5355	yy/mm/dd.
	2	SUB1	REL SUB1	45	0263	yy/mm/dd.
	3	SUB2	REL SUB2	45	3276	yy/mm/dd.
	4	SUB3	REL SUB3	45	1542	yy/mm/dd.

```
* EOI *          SUM =          264
```

```
ITEMIZE COMPLETE.
```

LIBEDIT COMMAND

LIBEDIT is a general-purpose utility that generates a file containing records copied from one or more other files (figure 16-2). LIBEDIT can build a random access directory for the new file. It recognizes the record types listed in Library Record Types in this section. LIBEDIT can edit a library according to directives requesting addition, deletion, or replacement of specified records from one or more replacement files.

LIBEDIT executes in two phases. During the first phase, it reads directives and replacement records. It groups directives by type and file and groups changes when several insertions take place relative to the same record.

During the second phase, LIBEDIT writes the new file. If LIBEDIT cannot process the specified combination of directives, and the D option (refer to description under Command Format) was not specified, LIBEDIT lists its interpretation of the conflicting directives, issues an error message, and aborts the job step. If the D option was specified, LIBEDIT continues processing the directives.

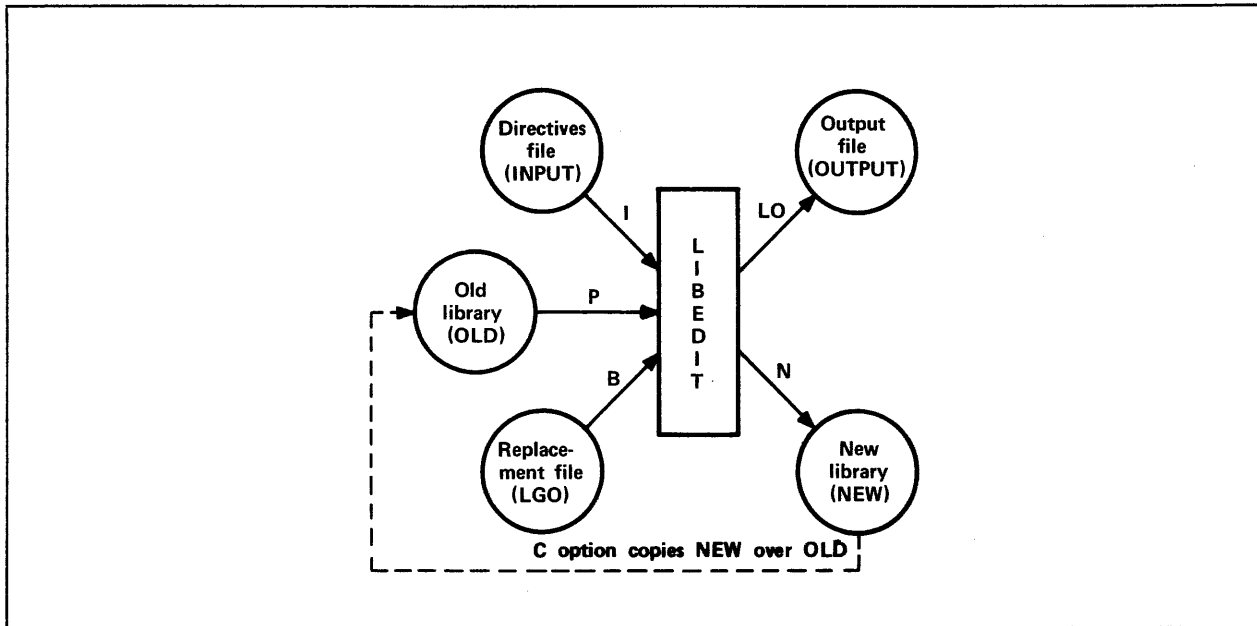


Figure 16-2. LIBEDIT Input and Output

COMMAND FORMAT

The following command calls LIBEDIT. Its parameters specify options and files used for the call as illustrated in figure 16-2.

Format:

LIBEDIT, p_1,p_2,\dots,p_n .

Optional parameters p_i can be in any order. Each parameter cannot be specified more than once.

<u>P_i</u>	<u>Description</u>
$P=lf_n_1$	Edit old file lf_n_1 . The default is OLD. $P=0$ directs the system to create the new file from replacement file(s).
$N=lf_n_2$	Write the new file on lf_n_2 . The default is NEW.
$I=lf_n_3$	Take the new directives from the next record of file lf_n_3 . The default is INPUT (INPUT is connected to the terminal in interactive jobs). $I=0$ indicates that there are no directives.

<u>Pi</u>	<u>Description</u>												
Z	Take the directives from the command line. The directives immediately follow the command terminator. The first character following the terminator is the separator character used to separate the directives in the sequence. It can be any character not used in the directives. For example: LIBEDIT,Z.#*DELETE REL/PROG1#*ADD PROG2 The Z parameter overrides the I=lf _{n3} .												
B=lf _{n4}	Use records from file lf _{n4} for insertions and replacements. The default is LGO. B=0 indicates that there is no replacement file.												
L=lf _{n5}	List output on file lf _{n5} . The default is OUTPUT. L=0 specifies no output.												
LO=listopt	Use list option listopt.												
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>listopt</u></th> <th style="text-align: left;"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>C</td> <td>List directives.</td> </tr> <tr> <td>E</td> <td>List errors.</td> </tr> <tr> <td>M</td> <td>List modifications.</td> </tr> <tr> <td>N</td> <td>List records written to new file.</td> </tr> <tr> <td>F</td> <td>Give a full listing (enable all list options).</td> </tr> </tbody> </table> <p>You can combine the list options; for example, LO=CM or LO=CE. The default is LO=EM for interactive jobs, and LO=F for all other jobs.</p>	<u>listopt</u>	<u>Meaning</u>	C	List directives.	E	List errors.	M	List modifications.	N	List records written to new file.	F	Give a full listing (enable all list options).
<u>listopt</u>	<u>Meaning</u>												
C	List directives.												
E	List errors.												
M	List modifications.												
N	List records written to new file.												
F	Give a full listing (enable all list options).												
V	Verify the new file against the old file after LIBEDIT processing by calling VFYLIB. The default is no call to VFYLIB. The U parameter overrides this parameter.												
C	Copy the new file over the old file after LIBEDIT processing.												
D	Do not abort LIBEDIT processing because of errors.												
U	Require the old file to be a user library, add any new binaries from the replacement file to the new file, and make the new file a user library by calling LIBGEN. This parameter overrides the V parameter.												
NA	Do not abort on directive errors.												
NI	Do not insert unreplaceable records at EOF of the new file.												
NR	Do not rewind the old or new file before or after LIBEDIT processing. By default, the system rewinds the old file and new file.												

<u>Pi</u>	<u>Description</u>
NX=n	Include cross-references in the library directory of the new user library if n is 0. If n is nonzero, do not include cross-references. This parameter is meaningful only if you specify the U parameter or the *LIBGEN directive. The default is NX=0.

LIBEDIT DIRECTIVES

You can specify directives to control LIBEDIT processing. These directives can be in a record on file INPUT (terminal input for interactive jobs), can be on the file specified by I=lf₃ parameter of the command, or can follow the terminator of the command when the Z parameter is specified.

Directives are not required. If I=0 is specified, LIBEDIT compares the name and type of each record on the old file with those of the records on the replacement file (specified by the B parameter). If a record with the same name and type appears on the replacement file, LIBEDIT writes that record on the new file and skips the record on the old file; otherwise, it copies the record from the old file to the new file unless you specified the NI parameter or *NOINS directive. If I=0 and B=0 are specified, LIBEDIT copies the old file to the new file until it encounters an EOF mark or an OPLD directory on the old file.

LIBEDIT directives take precedence over the parameters of the command when conflicting specifications occur.

LIBEDIT recognizes the following directives.

<u>Directive</u>	<u>Function</u>
*ADD	Inserts records before a zero-length record within the file.
*BEFORE or *B	Inserts record before the named record.
*BUILD	Builds a directory at the end of the new file.
*COMMENT	Adds a comment to the prefix table.
*COPY	Copies the new file to the old at the end of editing.
*DATE	Adds the date and a comment to the prefix table.
*DELETE or *D	Does not copy specified records to the new file.
*FILE	Declares a file to be a replacement file.
*IGNORE	Ignores records when reading the replacement file.
*INSERT or *I or *AFTER or *A	Copies record from the replacement file after copying the specified old file record.
*LIBGEN	Generates a user library by calling LIBGEN after LIBEDIT processing.
*LIST	Specifies a list file and list options.
*NEW	Specifies a new file.

<u>Directive</u>	<u>Function</u>
*NOINS	Prevents the insertion of unreplaceable records at EOF of the new file.
*NOREP	Does not automatically replace old file records with records from the specified file.
*NOREW	Prevents the old file and the new file from being rewound before and after LIBEDIT processing.
*OLD	Specifies an old file.
*RENAME	Renames record.
*REPLACE	Replaces the named records from the old file with records from the replacement file.
*REWIND	Names file to be rewound before and after editing.
*TYPE or *NAME	Sets default record type.
*VFYLIB	Verifies the new file against the old file by calling VFYLIB after LIBEDIT processing (overridden by *LIBGEN).

A directive begins with an asterisk in column 1 followed immediately by the directive identifier. The directive identifier and the first parameter are both delimited by a comma and/or one or more spaces. If a directive does not begin with an asterisk and a directive identifier, LIBEDIT assumes that the operation is a continuation of the previous directive operation. If an asterisk and directive identifier do not begin the first line of the directives record, LIBEDIT prefixes the following to the first line.

BEFORE,,

Parameters other than the first parameter are delimited by a space, an end-of-line, or a comma. A hyphen (-) indicates a record group. Record group identifiers (gid entries) cannot be split between lines. For example, the lines

```
*B,OVL/P1,OVL/P2,OVL/P
3
```

do not constitute a valid directive. The last entry would not be processed as OVL/P3. On the other hand, the lines

```
*B,OVL/P1,OVL/P2,
OVL/P3
```

do constitute a valid directive and would be processed as

```
*B,OVL/P1,OVL/P2,OVL/P3
```

Parameters common to many directives are the reference record identifier (rid) and the record group identifier (gid). Valid record types for these parameters are listed in Library Record Types in this section. The default type is the last type specified in a directive; if none are specified, TEXT is the default. The record name is the first seven characters of the record, or if a prefix table is present, the name in its second word. The first character of a record name specified in a directive must not be an asterisk.

rid Reference record identifier specifying the reference point for the requested change. It can have the following formats.

<u>Parameter</u>	<u>Description</u>
type/name	Reference record has the specified type and name.
name	Reference record has the specified name and is of the default type.
*	Reference point is an end-of-file mark (used with *BEFORE directive only).

gid Record group identifier indicating a record or group of records to be inserted, deleted, or replaced. It can have the following formats.

<u>Parameter</u>	<u>Description</u>
type/name	Record with the specified type and name.
name	Record with the specified name of the default type.
type ₁ /name ₁ -type ₂ /name ₂	Group of records beginning with name ₁ of type ₁ and ending with name ₂ of type ₂ .
type ₁ /name ₁ -name ₂	Group of records beginning with name ₁ of type ₁ and ending with name ₂ of type ₁ .
name ₁ -name ₂	Group of records beginning with name ₁ of the default type and ending with name ₂ of the default type.
type/name-*	All records of the specified type beginning with the named record.
name-*	All records of the default type beginning with the named record.
type/*	All records of the specified type.
*	All records of the default type.
0	A zero-length record is inserted.

LIBEDIT checks the syntax of directives during an interactive session and reprompts you for a directive in error.

ADD Directive

The ADD directive inserts records before a zero-length record. A CATALOG listing of the old file numbers each group of records ending with a zero-length record (called a library on the listing). This number on the ADD directive identifies the record group.

NOTE

Adding a zero-length record does not change the directory.

Format:

```
*ADD LIBn,gid1,gid2,...,gidn
```

<u>Parameter</u>	<u>Description</u>
LIBn	Specifies the record group to which the records are appended. Values for n are 1 to 63 and can be determined from a CATALOG listing of the old file.
gid _i	Identifies the records or groups of records from the current replacement file that are to be inserted before the zero-length record.

Example:

The following is a CATALOG listing of file Q.

REC	CATALOG OF Q		FILE	CKSUM	DATE
	NAME	TYPE	LENGTH		
1	REC1	TEXT	1	5302	
2	REC2	TEXT	1	5304	
3	(00)	SUM =	2	LIBRARY =	1
4	REC4	TEXT	1	5310	
	* EOI *	SUM =	3		

The following output results when a record was added to file Q, producing file Y.

LIBEDIT DIRECTIVE CARDS.		yy/mm/dd.	15.12.51.	PAGE 1
*ADD LIB1,REC3				
RECORDS WRITTEN ON FILE Y		yy/mm/dd.	15.12.51.	PAGE 2
	RECORD	TYPE	FILE	DATE
	REC1	TEXT	Q	
	REC2	TEXT	Q	
INSERTED	REC3	TEXT	X	
	00		Q	
	REC4	TEXT	Q	
	EOF		Q	

BEFORE Directive

A BEFORE directive inserts records or groups of records before a specified reference record on the old file. An old file record with the same name and type as an inserted record is not copied to the new file.

Formats:

*BEFORE rid,gid₁,gid₂,...,gid_n

or

*B rid,gid₁,gid₂,...,gid_n

<u>Parameter</u>	<u>Description</u>
rid	Names the old file record before which the specified replacement file records are to be inserted.
gid _i	Identifies records or groups of records from the current replacement file that are to be inserted before the reference record (rid).

If the first line of the LIBEDIT directives record does not begin with an asterisk and directive name, LIBEDIT assumes that the line is the gid parameters following a *BEFORE* directive.

BUILD Directive

A BUILD directive constructs and appends a random access directory to the new file. The directory is in Modify format (an OPLD record). If the old file has an OPLD directory, LIBEDIT constructs a directory for the new file with or without a BUILD directive. BUILD can also be used to change the directory name.

Format:

*BUILD dname

dname	Specifies the name for the directory record. The name must be one to seven alphanumeric characters. No default.
-------	---

COMMENT Directive

The COMMENT directive adds a comment to the prefix (77) table of a record written on the new file.

Format:

*COMMENT rid comment

<u>Parameter</u>	<u>Description</u>
rid	Name of a record to be written on the new file.
comment	A string of up to 70 characters that appears in the comment field of the prefix table. Additional characters are truncated.

COPY Directive

The COPY directive directs LIBEDIT to copy the new file over the old file after it has processed all directives. It performs the same function as the C parameter on the LIBEDIT command. This directive has no parameters.

DATE Directive

The DATE directive adds the current date and the specified comment to the prefix (77) table of a record written on the new file.

Format:

*DATE rid comment

<u>Parameter</u>	<u>Description</u>
rid	Record to be written on the new file.
comment	A string of up to 70 characters to be written in the comment field of the prefix table. Additional characters are truncated.

DELETE Directive

The DELETE directive suppresses copying of the specified records from the old file to the new file.

Formats:

```
*DELETE gid1,gid2,...,gidn
```

or

```
*D gid1,gid2,...,gidn
```

gid_i Identifies records or groups of records that are not to be copied from the old file to the new file.

Example:

```
*DELETE OVL/LAD-REL/RUN
```

This directive requests LIBEDIT not to copy the sequence of records starting with overlay LAD through relocatable CPU program RUN.

FILE Directive

The FILE directive names a file assigned to the job that contains replacement records. LIBEDIT directives following the FILE directive refer to records on the declared replacement file.

Format:

```
*FILE lfn
```

lfn Specifies the name of a replacement file. If lfn is an asterisk (*), LIBEDIT uses the replacement file specified by the LIBEDIT command. If the B parameter was omitted from the command, LGO is used.

IGNORE Directive

The IGNORE directive requests LIBEDIT to ignore a record or group of records on the current replacement file.

Format:

```
*IGNORE gid1,gid2,...,gidn
```

gid_i Identifies records or groups of records on the replacement file that are to be ignored.

Example:

```
*FILE ALPHA  
*IGNORE C-*
```

LIBEDIT ignores the sequence of records on file ALPHA starting with record C of the default type and including all records of the default type from C to the EOF mark.

INSERT or AFTER Directive

The INSERT or AFTER directive requests LIBEDIT to copy the specified records or groups of records from the current replacement file after it has copied the specified old file record onto the new file. Any record on the old file that has the same name and type as an inserted record is not copied to the new file.

Formats:

```
*INSERT rid,gid1,gid2,...,gidn
```

or

```
*I rid,gid1,gid2,...,gidn
```

```
*AFTER rid,gid1,gid2,...,gidn
```

or

```
*A rid,gid1,gid2,...,gidn
```

Example:

```
*INSERT OPL/K,TEXT/L
```

This directive requests LIBEDIT to copy the replacement file text record L to the new file after it has copied the old file OPL record K.

LIBGEN Directive

The LIBGEN directive specifies that the new file will be a user library. The U parameter of the LIBEDIT command performs the same function. The LIBGEN directive will override the VFYLIB directive.

Format:

*LIBGEN libname

libname Specifies the name of the new user library.

LIST Directive

The LIST directive changes the list file and the list option (refer to the L=lf_n and LO=listopt parameters of the LIBEDIT command described earlier).

Format:

*LIST listfile, listopt

<u>Parameter</u>	<u>Description</u>
listfile	Specifies the list file.
listopt	Specifies the list option.

NEW Directive

The NEW directive specifies the name of the new file. The N=lf_n parameter of the LIBEDIT command performs the same function.

Format:

*NEW newfile

newfile Specifies the name of the new file.

NOINS Directive

The NOINS directive prevents the insertion of unreplaceable records at EOF of the new file. The NI parameter of the LIBEDIT command performs the same function. This directive has no parameters.

NOREP Directive

The NOREP directive declares the specified files to be no-replace files. A no-replace file is a replacement file whose records do not automatically replace old file records having the same name and type. The user selects records to be written on the new file from no-replace files by specifying the file on a FILE directive and then naming the records on *AFTER, *BEFORE, *INSERT, and *REPLACE directives.

Format:

```
*NOREP lfn1,lfn2,...,lfnn
```

lfn₁ Specifies the no-replace files.

NOREW Directive

The NOREW directive prevents the old file and new file from being rewound before and after LIBEDIT processing. The NR parameter of the LIBEDIT command performs the same function. This directive has no parameters.

OLD Directive

The OLD directive specifies the name of the old file. The P=lfn₁ parameter of the LIBEDIT command performs the same function.

Format:

```
*OLD oldfile
```

oldfile Specifies the name of the old file.

RENAME Directive

The RENAME directive assigns a new name to a record written on the new file. If the renamed record is referenced by another directive in the directive record, the old name should be used. A RENAME is not allowed on a PROC record.

Format:

```
*RENAME rid,name
```

<u>Parameter</u>	<u>Description</u>
rid	Specifies the name of the replacement file record or old file record to be renamed.
name	Specifies the name of the record. The name can be one to seven alphanumeric characters.

REPLACE Directive

The REPLACE directive requests LIBEDIT to replace the old file records having the specified names and types with the replacement file records having matching names and types. This directive is used when the current replacement file has been declared a no-replace file (refer to the NOREP directive description). If the replacement file is not a no-replace file, LIBEDIT performs the replace operation automatically.

Format:

```
*REPLACE gid1,gid2,...,gidn
```

gid_i Specifies records or groups of records that appear on both the old file and the current replacement file.

Example:

The old file contains text records A, B, C, and D; the replacement file RF also contains text records named A, B, C, and D. Either of the following directive sequences writes records A and B from the old file and records C and D from file RF onto the new file.

Sequence 1	Sequence 2
*FILE RF	*FILE RF
*NOREP RF	*IGNORE A-B
*REPLACE C-D	

REWIND Directive

The REWIND directive tells LIBEDIT to rewind the specified file before and after processing.

Format:

```
*REWIND lfn
```

lfn Specifies the name of the file to be rewound.

TYPE or NAME Directive

A TYPE or NAME directive sets the default record type.

Formats:

```
*TYPE type
*NAME type
```

type Specifies default record types. Valid record types are listed in Library Record Types in this section.

The default record type can also be set by an explicit record type specification within a directive. In either case, the default record type setting remains in effect until another record type is explicitly named. If a default record type is not declared in the directive sequence, the default is TEXT. For example, the following two directive sequences are equivalent.

<u>Sequence 1</u>	<u>Sequence 2</u>
*TYPE REL	*INSERT REL/X,Y
*INSERT X,Y	*DELETE FILE1-FILE4
*DELETE FILE1-FILE4	

VFYLIB Directive

The VFYLIB directive verifies the new file against the old file by calling VFYLIB after LIBEDIT processing. The V parameter of the LIBEDIT command performs the same function. This directive has no parameters. The LIBGEN directive or V parameter will override the VFYLIB directive.

LIBEDIT OUTPUT

LIBEDIT interprets all directives in the directive record before beginning directive processing. If one or more errors are found, LIBEDIT issues the dayfile message

```
n DIRECTIVE ERROR(S).
```

and aborts the job step (unless the D parameter is specified on the command). The following LIBEDIT output shows the results of a directive syntax error (the FILE directive is not followed by a space or comma).

```
          DIRECTIVE ERRORS.                yy/mm/dd. 10.36.47          PAGE    1
*ERROR* *FILERF1
```

Directives which cannot be executed are listed as LIBEDIT interpreted them. The following LIBEDIT run called for a replacement file not assigned to the job.

```
          DIRECTIVE ERRORS.                yy/mm/dd. 10.37.39          PAGE    1
          *FILE RF1
          *B *,X
*ERROR* DIRECTIVE CARD CAN NOT BE PERFORMED
          *FILE      RF1
          *BEFORE   TEXT/*,TEXT/X
```

Nonfatal errors are listed in an error directory following the listing of records written to the new file. The RECORDS NOT REPLACED error shown in the following example could be corrected by including an *IGNORE directive naming the records not to be replaced.

```

DIRECTIVE ERRORS.                                yy/mm/dd. 03.37.43.  PAGE 1

  *FILER F2
  *B *,REC1
RECORDS WRITTEN ON FILE NEW                      yy/mm/dd. 03.37.43.  PAGE 2
RECORD  TYPE  FILE  DATE  COMMENT
INSERTED REC 1  TEXT  RF2
**EOF**
                                OLD

ERROR DIRECTORY - RECORDS NOT REPLACED.         yy/mm/dd. 08.37.43.  PAGE 3
RECORD  TYPE  FILE
REC3    TEXT  RF2
REC4    TEXT  RF2

```

If you are using LIBEDIT from an interactive job, the BRIEF command suppresses the printing of title lines on the output.

LIBGEN COMMAND

The LIBGEN command generates a user library of routines for use with CYBER Loader.

Format:

LIBGEN,P₁,P₂,...,P_n.

Any of the following parameters may be specified in any order (only one instance of each).

<u>P_i</u>	<u>Description</u>
F=1fn ₁	Specifies the source file containing absolute (ABS), overlay (OVL), procedure (PROC), relocatable (REL), or capsule (CAP) records for the user library. Other record types are ignored. The default for this parameter is LGO.
P=1fn ₂	Specifies the file on which the user library is to be written. The default is ULIB.
N=name	Specifies the name of the user library generated; name is entered in ULIB and OPLD records. The default is the user library name determined by the P=1fn ₂ parameter.
NX=n	Specifies whether or not to include cross-references in the ULIB directory. If n is not zero, no cross-references are included in the ULIB directory. If n is zero, cross-references are included. The later case is the default.

If the F=lf_{n1} and P=lf_{n2} parameters specify the same file, LIBGEN issues a dayfile message and does not generate a user library.

Figure 16-3 illustrates the structure of a user library. To generate a user library, the system rewinds and scans the input file. It then builds a directory of all entry points, program names, and external references in the records of the file. Adding the ULIB and OPLD records, the system then copies the input file to the user library file. The system aborts if duplicate entry point names appear in the file unless they are special entry points (refer to the NOS 2 Systems Programmer's Instant).

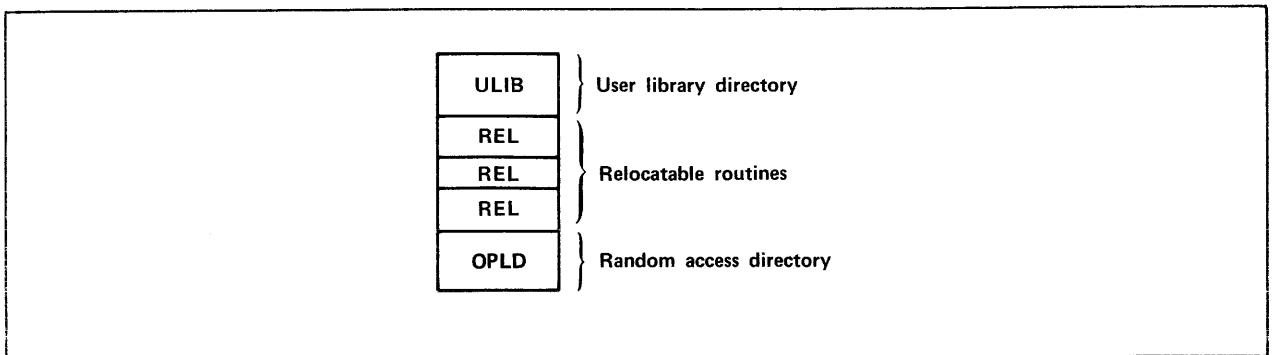


Figure 16-3. User Library Structure

Unless the NX=n parameter specifies otherwise, the ULIB directory contains the external reference/entry point linkage between routines in the user library. When CYBER Loader loads a routine from the user library, it loads (at the same time) all user library routines referenced by the requested routine. All externals for user library routines are satisfied from the user library, if possible. If desired, you can request with the NX=n parameter that the ULIB directory contains no cross-linking of records. In that case, when a routine from the user library is requested, only that routine is loaded. Loading in this manner is typically slower than using the cross-reference linkage option.

Example 1:

File RELB contains relocatable routines that are used for execution of several applications. To enable loading of these routines as needed during execution of an application program, the user generates a user library using the following command.

```
LIBGEN,F=RELB,P=MYLIB,N=APPLIB.
```

This creates user library APPLIB on file MYLIB. The following loader sequence allows use of the APPLIB routines during execution of a compiled FORTRAN 5 program on file LGO.

```
LDSET,LIB=MYLIB.  
LOAD,LGO.  
EXECUTE.
```

The program is loaded and executed with externals satisfied first from user library MYLIB and then from the system default library SYSLIB. Refer to the CYBER Loader Reference Manual for more information on library search procedures.

Example 2:

If a routine has no external references, no entry is made in the ULIB directory. To load this routine, you must include the loader command LDSET(USEP=pname) in a loader sequence.

Suppose a FORTRAN 5 contains a BLOCK DATA subroutine without external references to any of its entry points. The user has not named the block, and it has the default name BLKDAT. To load this routine, you must include the following command in the loader sequence.

```
LDSET,USEP=$BLKDAT.$.
```

VFYLIB COMMAND

The VFYLIB command rewinds two files, compares their records, and lists the differences. The record types that VFYLIB recognizes are listed in Library Record Types in this section. VFYLIB lists changes in residence (between record groups separated by zero-length records), replacements, deletions, and insertions. A record is defined as being replaced when its name and type remain the same, but its contents differ. VFYLIB does not compare prefix (77) table information such as last modification date and last assembly date. VFYLIB does not consider a difference in record order as a difference between the two files.

Format:

```
VFYLIB,lfn1,lfn2,lfn3,NR.
```

<u>Parameter</u>	<u>Description</u>
lf _{n1}	Name of the first file; if this parameter is omitted, VFYLIB assumes file OLD.
lf _{n2}	Name of the second file; if this parameter is omitted, VFYLIB assumes file NEW.
lf _{n3}	Name of the output file; if this parameter is omitted, VFYLIB assumes file OUTPUT.
NR	If specified, lf _{n1} and lf _{n2} are not rewound after verification.

Example:

The following are CATALOG listings of file OLD and file NEW.

```
1          CATALOG OF OLD          FILE      1
  REC  NAME      TYPE      LENGTH  CKSUM      DATE
    1  A          REL        41      7424  yy/mm/dd.
    2  A          (00)      SUM =    41  LIBRARY =    1
    3  B          REL        41      0465  yy/mm/dd.
    4  B          C          REL        41      5525  yy/mm/dd.
    5  * EOF *    SUM =    143

1
  CATALOG COMPLETE.
```

```
1          CATALOG OF NEW          FILE      1
  REC  NAME      TYPE      LENGTH  CKSUM      DATE
    1  A          REL        41      7424  yy/mm/dd.
    2  B          REL        41      0465  yy/mm/dd.
    3  B          (00)      SUM =   102  LIBRARY =    1
    4  D          TEXT        1      1000
    5  * EOF *    SUM =   103

1
  CATALOG COMPLETE.
```

```
1  VFYLIB.      OLD FILE = OLD  NEW FILE = NEW  yy/mm/dd. 12.18.25.      PAGE      1
0      RECORD   TYPE      ULIB      LIB  DATE      COMMENT
0  CHANGES IN RESIDENCE.
      B          REL          1  yy/mm/dd. SUBROUTINEOPT=0
0  DELETED PROGRAMS.
      C          REL          2  yy/mm/dd. SUBROUTINEOPT=0
0  INSERTED PROGRAMS.
      D          TEXT          2
```

LIBRARY PROCESSING EXAMPLES

The following examples illustrate the use of CATALOG, GTR, LIBEDIT, and LIBGEN commands. To duplicate the examples, you should execute the jobs in sequence.

Example 1:

The following job builds a program library from a replacement file of relocatable binary (REL) records.

```

/JOB
LIBTES1.
/USER
CHARGE,*.
FTN5,L=0.
DEFINE,TESTLIB.
CATALOG,LGO,R.
LIBEDIT,P=0,N=TESTLIB.
CATALOG,TESTLIB,R.
/EOR
    SUBROUTINE A
    STOP
    END
    SUBROUTINE D
    STOP
    END
    SUBROUTINE C
    STOP
    END
    SUBROUTINE B
    STOP
    END
/EOR
*BUILD LIBRARY
*B,*,REL/A,B,C,D
/EOF

```

The FORTRAN compilation produces relocatable binaries on the default file LGO. The DEFINE command creates a direct access permanent file TESTLIB on which the new program library is written. The first CATALOG command lists the LGO file as follows:

1	REC	CATALOG OF LGO NAME	TYPE	FILE LENGTH	1 CKSUM	yy/mm/dd. DATE	12.34.49. COMMENTS	PAGE	1
	1	A	REL	41	7424	yy/mm/dd.	SUBROUTINEOPT=0		
	2	D	REL	41	6566	yy/mm/dd.	SUBROUTINEOPT=0		
	3	C	REL	41	5525	yy/mm/dd.	SUBROUTINEOPT=0		
	4	B	REL	41	0465	yy/mm/dd.	SUBROUTINEOPT=0		
	5	* EOF *	SUM =	204					

The P=0 in the LIBEDIT command indicates that no old program library exists. The N parameter indicates that the new program library is written on file TESTLIB. The replacement file is the default LGO. The directives are on the default INPUT file.

LIBEDIT reads the binaries from LGO and the directives from INPUT. On the basis of the directive specifications, the binaries are inserted before the end-of-file on file TESTLIB in the order specified in the directives (A, B, C, D). The directory record created is given the name LIBRARY as a result of the *BUILD directive. It is written before the end-of-file on the new program library TESTLIB.

The directives are written to OUTPUT. The records on file TESTLIB are listed on the next page of OUTPUT. The following listing consists of these two pages.

```
LIBEDIT DIRECTIVES.                yy/mm/dd. 12.34.52.        PAGE    1

      *BUILD LIBRARY
      *B,*,REL/A,B,C,D
1
      RECORDS WRITTEN ON FILE TESTLIB                yy/mm/dd.
12.34.52.                PAGE    2

      RECORD   TYPE   FILE   DATE   COMMENT
INSERTED A     REL   LGO    yy/mm/dd. SUBROUTINEOPT=0
INSERTED B     REL   LGO    yy/mm/dd. SUBROUTINEOPT=0
INSERTED C     REL   LGO    yy/mm/dd. SUBROUTINEOPT=0
INSERTED D     REL   LGO    yy/mm/dd. SUBROUTINEOPT=0
ADDED LIBRARY  OPLD   ***** yy/mm/dd.
**EOF**
```

The second CATALOG command produces the following listing of information about the records on TESTLIB.

```
1  CATALOG OF TESTLIB  FILE  1  yy/mm/dd. 12.34.52.  PAGE  1
  REC  NAME  TYPE  LENGTH  CKSUM  DATE  COMMENTS
  1  A  REL  41  7424  yy/mm/dd. SUBROUTINEOPT=0
  2  A  REL  41  0465  yy/mm/dd. SUBROUTINEOPT=0
  3  B  REL  41  5525  yy/mm/dd. SUBROUTINEOPT=0
  4  C  REL  41  6566  yy/mm/dd. SUBROUTINEOPT=0
  5  D  REL  41  2073  yy/mm/dd.
  6  LIBRARY  OPLD  13  2073  yy/mm/dd.
  6  * EOF *  SUM =  217
```

Example 2:

This job builds a new program library from an old program library by inserting new relocatable routines into and deleting routines from the old program library created in example 1 (TESTLIB).

```

/JOB
LIBTES2.
/USER
CHARGE,*.
FTN5,L=0.
ATTACH,OLD=TESTLIB.
DEFINE,NEW=TES2LIB.
LIBEDIT.
CATALOG,NEW,R.
/EOB
    SUBROUTINE BONE
    STOP
    END
    SUBROUTINE NEWC
    STOP
    END
    SUBROUTINE D
    STOP
    END
/EOB
*TYPE REL
*I,B,BONE
*D,C
/EOF

```

Three relocatable binaries (BONE, D, and NEWC) are produced via a FORTRAN compilation. The old program library (TESTLIB) is attached in read mode and referenced as OLD. A direct access file (TES2LIB) is created for the new program library. This file is referenced as NEW.

LIBEDIT reads the binaries from the replacement file LGO and the input directives from file INPUT. It writes the modified old program library (OLD) to the new program library (NEW). BONE and NEWC are inserted after records B and C, respectively, and record C is deleted. Record D, which already existed on the old program library, is replaced by record D from the replacement file LGO. The following action is taken on file NEW.

LIBEDIT DIRECTIVES. yy/mm/dd. 12.34.58. PAGE 1

```

*TYPE REL
*I,B,BONE
*D,C

```

1 RECORDS WRITTEN ON FILE NEW yy/mm/dd. 12.34.58. PAGE 2

RECORD	TYPE	FILE	DATE	COMMENT
A	REL	OLD	yy/mm/dd.	SUBROUTINEOPT=0
B	REL	OLD	yy/mm/dd.	SUBROUTINEOPT=0
INSERTED BONE	REL	LGO	yy/mm/dd.	SUBROUTINEOPT=0
DELETED-(C)	REL	OLD		
REPLACED D	REL	LGO	yy/mm/dd.	SUBROUTINEOPT=0
ADDED NEWC	REL	LGO	yy/mm/dd.	SUBROUTINEOPT=0
ADDED LIBRARY	OPLD	*****	yy/mm/dd.	
EOF		OLD		

The CATALOG output shows the following contents of the new program library.

1	CATALOG OF NEW	FILE	1	yy/mm/dd.	12.34.58.	PAGE	1
REC	NAME	TYPE	LENGTH	CKSUM	DATE	COMMENTS	
1	A	REL	41	7424	yy/mm/dd.	SUBROUTINEOPT=0	
	A						
2	B	REL	41	0465	yy/mm/dd.	SUBROUTINEOPT=0	
	B						
3	BONE	REL	41	4312	yy/mm/dd.	SUBROUTINEOPT=0	
	BONE						
4	D	REL	41	6566	yy/mm/dd.	SUBROUTINEOPT=0	
	D						
5	NEWC	REL	41	4432	yy/mm/dd.	SUBROUTINEOPT=0	
	NEWC						
6	LIBRARY	OPLD	15	7700	yy/mm/dd.		
7	* EOF *	SUM =	262				

Example 3:

This job uses LIBGEN to generate a user library file from the program library file TES2LIB created in example 2.

```

/JOB
LIBTES3.
/USER
CHARGE,*.
ATTACH, TES2LIB.
DEFINE, LIBFILE.
LIBGEN, F=TES2LIB, P=LIBFILE, N=LOADLIB.
CATALOG, LIBFILE, R, U.
/EOF

```

The program library TES2LIB is attached to the job. A direct access file LIBFILE is defined for writing the user library file.

LIBGEN scans TES2LIB and builds a ULIB directory of entry points, program names, and external references for relocatable (REL) records in the file. ULIB is copied to the file LIBFILE, followed by the records from TES2LIB. A file index of addresses for each record in the file is added as the last record of LIBFILE. LOADLIB is the name of the ULIB and OPLD records.

The CATALOG output of the user library file LIBFILE shows the following content.

1	CATALOG OF LIBFILE			FILE	1	yy/mm/dd. 12.35.07.		PAGE	1
	REC	NAME	TYPE	LENGTH	CKSUM	DATE	COMMENTS		
	1	LOADLIB	ULIB	13	7725	yy/mm/dd.			
	2	A	REL	41	7424	yy/mm/dd.	SUBROUTINEOPT=0		
		A							
	3	B	REL	41	0465	yy/mm/dd.	SUBROUTINEOPT=0		
		B							
	4	BONE	REL	41	4312	yy/mm/dd.	SUBROUTINEOPT=0		
		BONE							
	5	D	REL	41	6566	yy/mm/dd.	SUBROUTINEOPT=0		
		D							
	6	NEWC	REL	41	4432	yy/mm/dd.	SUBROUTINEOPT=0		
		NEWC							
	7	LOADLIB	OPLD	15	0515	yy/mm/dd.			
	8	* EOF *	SUM =	275					

Example 4:

This job illustrates a method for deleting records from a user library.

```

/JOB
LIBTES4.
/USER
CHARGE,*
ATTACH,LIBFILE/M=W.
LIBEDIT,P=LIBFILE,U,Z.#*D,REL/NEWC
CATALOG,NEW,R,U.
/EOB

```

LIBEDIT references the program library LIBFILE and the directive following the command, deletes NEWC, and writes this modified file on the default NEW. The output shows the LIBEDIT processing.

```

LIBEDIT DIRECTIVES.                yy/mm/dd. 12.35.14.                PAGE      1

      *D,REL/NEWC
1  RECORDS WRITTEN ON FILE NEW      yy/mm/dd. 12.35.14.                PAGE      2
      RECORD   TYPE      FILE      DATE      COMMENT
      A        REL       LIBFILE   yy/mm/dd. SUBROUTINEOPT=0
      B        REL       LIBFILE   yy/mm/dd. SUBROUTINEOPT=0
      BONE     REL       LIBFILE   yy/mm/dd. SUBROUTINEOPT=0
      D        REL       LIBFILE   yy/mm/dd. SUBROUTINEOPT=0
DELETED-(NEWC) REL       LIBFILE
ADDED  LOADLIB OPLD     ***** yy/mm/dd.
      **EOF**      LIBFILE

```

LIBGEN generates a new user library on the file NEW. It uses LIBFILE as the source and names the new user library ULIB.

The user library is cataloged, showing the following contents.

```

1  CATALOG OF NEW          FILE 1  yy/mm/dd. 12.35.15.    PAGE      1
   REC  NAME      TYPE      LENGTH  CKSUM  DATE      COMMENTS
   1  ULIB      ULIB      11      1055  yy/mm/dd.
   2  A         REL       41      7424  yy/mm/dd. SUBROUTINEOPT=0
   3  A         REL       41      0465  yy/mm/dd. SUBROUTINEOPT=0
   4  B         REL       41      4312  yy/mm/dd. SUBROUTINEOPT=0
   5  BONE     REL       41      6566  yy/mm/dd. SUBROUTINEOPT=0
   6  D         REL       41      7632  yy/mm/dd.
   7  ULIB     OPLD     13      7632  yy/mm/dd.
   7  * EOF *      SUM =    230

```


This manual distinguishes network functions from IAF functions to aid in error checking and to allow you to carry over knowledge of the network to other products (RBF, MCS, and TAF). Even so, the network interface is largely invisible to the IAF user. In most cases, you can ignore the existence of the network and proceed as if you are communicating exclusively with IAF. However, certain commands which involve communication only with the network are available to the IAF user (appendix K). These network commands change the characteristics of a terminal (for example, changing the backspace character or initiating paper tape operations), change the character code set of a terminal, and send messages to the local operator. Unlike other system commands, these network commands can be used any time a data path exists between the terminal and the system, including prior to the completion of login.

TERMINALS

IAF and the network provide support for 15 terminal classes.† Each terminal class corresponds to an actual terminal or terminals as follows:

<u>Class</u>	<u>Terminal</u>
1	M33, M35, M37, M38 teletypewriters
2	CDC 713-10
3	Reserved for future use
4	IBM 2741
5	M40
6	Hazeltine 2000
7	CDC 751-1, 752, 756
8	Tektronix 4014
9	HASP Protocol
10	200 User Terminal
11	CDC 714-30
12	CDC 711-10
13	CDC 714-10/20
14	CDC 731-12, 732-12
15	CDC 734

†Terminals of all classes can be connected to NOS. Some terminals in terminal classes 1, 2, 5, and 8 can be connected to a packet switching network (PSN) with a logical connection established to NOS.

Due to the flexibility of the network, a computer site may support terminals other than those listed. Consult local site personnel to determine if terminal types not listed are supported and to obtain additional operating instructions for supported nonstandard terminals.

Associated with each terminal class are specific terminal operating characteristics. Default values are listed in appendix K. A site can change these default values. If it does, it should provide you with a list of the new values. You can also change terminal characteristics with the TRMDEF command (section 8) and with the terminal definition commands (appendix K). Important characteristics which may vary from terminal to terminal include:

- Abort output line character (AL).
- Backspace character (BS).
- Cancel character (CN).
- Control character (CT).
- Interruption character (B1).
- Termination character (B2).

INPUT/OUTPUT CONVENTIONS

There are many standards and conventions that apply to input and output. Some features, however, vary according to terminal class as this section indicates.

INPUT FROM TERMINAL

The system normally issues you input prompts. However, you need not wait for an input prompt before entering the next input line (refer to Entering Input Before Prompts later in this section). Any input from the terminal halts output to the terminal until you enter a **CR**.†

A question mark (?) output to the terminal is usually a request from an executing program for user input. You can enter terminal definition commands in response to any request for input; for example, when you are in text mode or are using Text Editor or XEDIT. If you enter a terminal definition command following a request for input, the system does not send a new prompt.

† If you are at a PSN terminal, however, you must wait for output to cease before entering more input.

INPUT LINE DELIMITERS

The line feed key $\text{\textcircled{LF}}$, which terminates physical lines, and carriage return key $\text{\textcircled{CR}}$, which terminates logical lines, vary with terminal class. $\text{\textcircled{LF}}$ and $\text{\textcircled{CR}}$ are generic symbols, which represent characters that terminate a line of text.

LENGTH OF OUTPUT/INPUT LINES

A logical input line can consist of a maximum of 160 characters, but the system does not impose a limit on the length of a logical output line. The current page width[†] determines the maximum number of characters in a physical output line. If a logical output line exceeds the current page width, the system places the additional characters on the next physical line. If you enter a logical input line (terminated by a carriage return) longer than 160 characters, the system discards the entire logical line and issues the *OVL* message.

NETWORK MESSAGES DURING INPUT

The following diagnostic messages can appear during input from the terminal.^{††}

<u>Message</u>	<u>Description</u>
REPEAT..	Due to a temporary overload condition, the network has discarded the last line of input. You should reenter this information.
FROM THE LOP: message	A message has been sent by the local operator. You can respond to this message by entering the message (MS) command described in appendix K.
xxERR..	You entered either a terminal definition or an interactive status command that contained a format error or a parameter that is either invalid for your terminal type or not supported at your site. The xx is the terminal definition keyword in error.

[†]This characteristic varies with the terminal class. You can change this characteristic with the TRMDEF command (section 8) or with one of the terminal definition commands (appendix K).

^{††}A PSN exercises its own control over terminal input. Consult the documentation provided by the PSN for diagnostic messages that occur during input.

TERMINATING INPUT LINE

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

You terminate each physical line of input by pressing LF . The system responds by positioning the carriage (or cursor) at the beginning of the next line.† If the input line exceeds the maximum page width,†† the system terminates the physical input line.

If you type either CR LF or LF CR , the system ignores the second entry of each pair.††† This accommodates paper tape input and block mode input. Unlike a carriage return, a line feed does not delimit logical lines of input, not even an empty line. For example, if you attempt to input an empty line by entering a LF rather than a CR , the system ignores the empty line. You must then follow the LF with two CR s to input the empty line.

Appendix L lists the carriage return, new line, and line feed keys for the various terminal classes.

CORRECTING INPUT LINE

You can correct entry errors in the input line before pressing CR by using the backspace character.†† One character (including spaces) is deleted for each backspace character entered. If the beginning of the physical 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 (or a line of text interpreted as the word BASIC, if you are in text mode).

† A PSN does not normally reposition its terminals to the next line until it sends the next output message. You must reposition the carriage on terminals connected to a PSN (for example, by entering a carriage return in local mode).

†† This characteristic varies with terminal class. You can change the characteristic with TRMDEF command (section 8) or with one of the terminal definition commands (appendix K).

††† Not true for terminals connected to a PSN.

DELETING INPUT LINE

When you discover an error in the command or other input before pressing $\text{\textcircled{CR}}$, you can delete the current logical input line by entering the cancel line character† followed by a carriage return. For most terminals, this is CTRL/X or the (character. Default values for each terminal class are given in appendix K.

The network responds to the cancel line character by printing *DEL* on the next line and positioning the carriage or cursor to the beginning of a new line. The following example illustrates the use of the cancel line character.

```
READY.  
10 program t  
20 print 6  
30 6 format 9'this is a test  
*DEL*                                You enter the cancel line character.  
30 6 format ('this is it')  
40 end  
list  
  
10 PROGRAM T  
20 PRINT 6  
30 6 FORMAT ('THIS IS IT')  
40 END  
  
READY.
```

ENTERING INPUT BEFORE PROMPTS

Under IAF, you normally wait for an input prompt before entering data or commands. This prompt appears as either a READY message, a slant (/), a question mark (?), or a program-defined prompt. You can, however, enter input before receiving a prompt; the system processes the input in the order that you enter it. The prompts the system issues after processing individual commands still appear at the terminal in their normal manner. If you enter too many commands before receiving an input prompt, the system issues the following message;

```
REPEAT..
```

You should then reenter the last command after IAF issues its next prompt.

Even if it is currently processing another command, the system processes these three types of input immediately: terminal definition commands, interactive status commands, and user break sequences.

Terminal definition commands change the terminal's characteristics. Appendix K gives a detailed description of terminal definition commands.

Interactive status commands get an immediate status response from the system. The interactive status commands, cD, cE, and cS, use the control character c defined for your terminal (refer to table K-1). Detailed descriptions of cD, cE, and cS appear in section 8.

†This characteristic varies with the terminal class. You can change this characteristic with the TRMDEF command (section 8) or with one of the terminal definition commands (appendix K).

Entry of a user break sequence has special significance. Not only does a user break sequence interrupt or terminate the execution of the current command, but it also discards any input lines the system has not yet processed. User break sequences are described later in this section.

To momentarily suspend terminal output without clearing unprocessed input, you can type any character. To resume output, you should delete the character by entering a line delete. Otherwise, the character becomes an input line, which IAF will attempt to process.

PAGE WAIT

Each terminal class defines a certain number of lines of output as a page.† For a display terminal with page wait enabled, output stops at each page boundary. To display the next page, you can press Ⓢ or, for mode 4 terminals, the SEND key.†† For a printing terminal with page wait enabled, the system inserts six line feeds at each page boundary and output continues. If output ceases and the message

OVER..

is displayed at the end of the output, while the present page is not full, another page exists and can be viewed by pressing Ⓢ.

SUSPENDING OUTPUT

The procedure for suspending output varies with the terminal type. To suspend output on asynchronous terminals (terminal classes 1 through 8), you can press the BREAK key (the BREAK key is sometimes labeled INT, INTRPT, or ATTN). You can also suspend output by pressing any other key.††† You can then enter the cancel line character (CN) † followed by a carriage return to continue output and to avoid having the keyed entry interpreted as input.

Synchronous terminals do not have a BREAK key. You cannot suspend output on these terminals, except through the normal page wait controls.

After suspending output with the BREAK key, you can either abort the current output line, interrupt the current job step, or terminate the current job step. The next three subsections define the procedures for each of these options. If you enter input other than the following, the system resumes output.

ABORTING OUTPUT LINE

To abort the current output line, first suspend output by pressing the BREAK key and then enter the abort line character followed by a carriage return.††††

† This characteristic varies with the terminal class. It can be changed using the TRMDEF command (section 8) or one of the terminal definition commands (appendix K).

†† If you are using a terminal connected to a PSN or a mode 4 terminal, you can enter the control character (CT) followed by a carriage return to see the next page.

††† On asynchronous terminals connected to a PSN, you can only suspend output by using the BREAK key, which might result in the loss of data the PSN holds.

†††† A PSN does not support the abort output line feature for its terminals.

INTERRUPTING JOB STEP

To interrupt a job step, you enter the interruption character† followed by a carriage return. For most terminals, the interruption character is CTRL/P, CTRL/F, or a colon (:). Appendix K lists the default values for each terminal class. The interruption sequence is also called the user break 1 sequence. IAF responds with the message:

INTERRUPTED

and discards any input you may have previously entered.

If the system is transmitting output to the terminal, first suspend output by pressing the BREAK key and then enter the interruption sequence. To interrupt output on synchronous terminals, you can enter the interruption sequence only when output ceases at a page boundary. You must have already enabled page wait.

TERMINATING JOB STEP

To terminate a job step, enter the termination character† followed by a carriage return. For most terminals, the termination character is CTRL/T or a right parenthesis. Appendix K lists the default values for each terminal class. The termination sequence is also called the user break 2 sequence. IAF responds with

TERMINATED

and discards any input you may have entered previously.

If the system is transmitting output to the terminal, first suspend terminal output by pressing the BREAK key and then enter the termination sequence. To terminate output on synchronous terminals, you enter the termination sequence only when output ceases at a page boundary. You must have already enabled page wait.

NOTE

You cannot enter the interruption or termination sequence if output has been suspended with a key other than the BREAK key or the interruption or termination character.

†This characteristic varies with the terminal class. It can be changed using the TRMDEF command (section 8) or one of the terminal definition commands (appendix K).

JOB SUSPENSION

You can suspend your job during program execution by entering the interruption sequence (refer to appendix K for default interruption sequences for all terminal classes). If the program is actively transmitting output to the terminal, the BREAK (or ATTN) key must first be pressed to inhibit output. The interruption sequence can then be entered to suspend the job. The system responds

INTERRUPTED

and discards any input previously entered.

If your terminal does not have a BREAK key (sometimes labeled INT, INTRPT, or ATTN), a job cannot be suspended while it is transmitting output to the terminal.

NOTE

On asynchronous terminals (terminal classes 1 through 8), output can be suspended by pressing any key. You can enter the cancel line character to continue output. The interruption sequence will not interrupt the output if you enter the sequence after suspending output by pressing a key other than the BREAK key.

A job is also suspended automatically by IAF when:

- A successful recovery has been performed (refer to RECOVER command in section 8).
- The job has exceeded its time limit. The time limit is initially set to 64 CPU seconds at login, although you can change this value through use of the SETTL command (refer to section 7).
- 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 section 7).

If the job is suspended by you (interrupted), you can enter one of the following:

<u>Entry</u>	<u>Description</u>
Ⓞ	Continues output. A portion of the output is lost if the program was transmitting output to the terminal when suspended. When output is complete, program execution continues.
P	Proceeds. If the program was transmitting output to the terminal when suspended, the system discards the data generated by the program prior to the interruption sequence. The amount of output discarded depends on the program being executed. Program execution continues.
Other	Stops. Terminates the job step.

The following example illustrates the use of the interruption sequence to interrupt a FORTRAN 5 program.

```
list
00100 PROGRAM D0
00110 INTEGER E
00120 DO 10 E=1,300
00130 PRINT*,E
00140 10 CONTINUE
00150 READ*,I,J
00160 PRINT*,I,J
00170 STOP
00180 END
READY.
run
```

```
1
2
3
```

```
*INTERRUPTED*
```

```
Ⓢ
110
111
112
```

```
*INTERRUPTED*
```

```
Ⓢ
pⓈ
193
194
195
196
```

```
*INTERRUPTED*
```

```
pⓈ
? 4,5
4 5
```

```
SRU 0.863 UNTS.
```

```
RUN COMPLETE.
```

You interrupt with the BREAK key to stop output and enter the interruption sequence to suspend the job. You continue by pressing Ⓢ (some output is lost).

You again press the BREAK key to stop output and enter the interruption sequence to suspend the job.

You enter PⓈ, directing the system to discard data. As execution continues, more data is printed at the terminal.

You again suspend the job.

You enter PⓈ directing the system to discard data. All output has been generated. Program execution continues.

The number of times you would have to interrupt a program and enter PⓈ before all output has been generated depends on the program.

USING ATTN ON IBM 2741 TERMINALS

You must press the ATTN key before entering the terminal definition control character, cancel character, abort line character, interruption character, or termination character. For further information about these characters, refer to appendix K.

CONTROL BYTES

You can design an interactive program to control terminal activity by including control bytes in your 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. You 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 your job 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 job.
0005	Initiate ASCII input mode.
0006	Initiate transparent input mode.
0007	Initiate transparent output mode.
0011	Initiate ASCII output mode.
0013	End-of-string.
0016	Terminal redefinition.

The use of these control bytes is described in detail in Volume 4, Program Interface.

CHARACTER SETS

A

A character set is composed of graphic and/or control characters. A code set is a set of codes used to represent each character within a character set.

A graphic character may be displayed at a terminal or printed by a line printer. Examples are the characters A through Z and the digits 0 through 9. A control character initiates, modifies, or stops a control operation. An example is the backspace character that moves the terminal carriage or cursor back one space. Although a control character is not a graphic character, a terminal may produce a graphic representation when it receives a control character.

All references within this manual to the ASCII character set or the ASCII code set refer to the character set and code set defined in the American National Standard Code for Information Interchange (ASCII, ANSI Standard X3.4-1977). References in this manual to the ASCII character set do not necessarily refer to the ASCII code set.

NOS supports the following character sets.

- CDC graphic 64- (or 63-) character set.
- ASCII 128-character set.
- ASCII graphic 64- (or 63-) character set.
- ASCII graphic 95-character set.

Each installation selects either the 64-character set or the 63-character set. The differences between the two are described in Character Set Anomalies in this appendix. Any reference in this appendix to the 64-character set implies either the 63- or 64-character set, unless otherwise stated.

NOS supports the following code sets.

- Display code.
- 6/12 display code.
- 12-bit ASCII code.

Display code is a set of 6-bit codes from 00₈ to 77₈.

The 6/12 display code is a combination of 6-bit codes and 12-bit codes. The 6-bit codes are 00₈ through 77₈, excluding 74₈ and 76₈. (Refer to Character Set Anomalies for the interpretation of the 00₈ and 63₈ codes.) The 12-bit codes begin with either 74₈ or 76₈ and are followed by a 6-bit code. Thus, 74₈ and 76₈ are considered escape codes and are never used as 6-bit codes within the 6/12 display code set. The 12-bit codes are 7401₈, 7402₈, 7404₈, 7407₈, and 7601₈ through 7677₈. All other 12-bit codes (74xx₈ and 7600₈) are undefined.

The 12-bit ASCII code is the ASCII 7-bit code (as defined by ANSI Standard X3.4-1977) right-justified in a 12-bit byte. Assuming that the bits are numbered from the right starting with 0, bits 0 through 6 contain the ASCII code, bits 7 through 10 contain zeros, and bit 11 distinguishes the 12-bit ASCII 0000₈ code from the end-of-line byte. The 12-bit codes are 0001₈ through 0177₈ and 4000₈.

CHARACTER SET ANOMALIES

NOS interprets two codes differently when the installation selects the 63-character set rather than the 64-character set. In tables A-1, A-2, and A-3, the codes for the colon and percent graphic characters in the 64-character set are unshaded; the codes for the colon and percent graphic characters in the 63-character set are shaded.

If an installation uses the 63-character set, the colon graphic character is always represented by a 63₈ code. However, if the installation uses the 64-character set, output of 6/12 display code 7404₈ or 00₈ produces a colon. In ASCII mode for interactive jobs, a colon can be input only as a 7404₈ 6/12 display code.

When using either the 63- or 64-character set, the use of undefined 6/12 display codes in output files produces unpredictable results and should be avoided.

On input, NOS recognizes alternate 029 punch codes of 11-0 for the right bracket (]) and 12-0 for the left bracket ([). The alternate codes support the COBOL sign overpunch convention and are not recommended for other uses. Refer to COBOL 5 Reference Manual.

Also, two 00₈ codes may be confused with an end-of-line byte and should be avoided (refer to appendix F for further explanation).

CHARACTER SET TABLES

This appendix contains character set tables for interactive jobs, batch jobs, and jobs involving magnetic tapes. Table A-1 is for interactive jobs, and table A-2 is for batch jobs. Table A-3 is a conversion table used to cross-reference 12-bit ASCII codes and 6/12 display codes and to convert ASCII codes from octal to hexadecimal.

Tables A-4, A-5, and A-6 list the magnetic tape codes and their display code equivalents.

The character set tables are designed so that you can find the character represented by a code (such as in a dump) or find the code that represents a character. To find the character represented by a code, the user looks up the code in the column listing the appropriate code set and then finds the character on that line in the column listing the appropriate character set. To find the code that represents a character, you first look up the character and then find the code on the same line in the appropriate column.

INTERACTIVE JOBS

Table A-1 shows the character sets and code sets available to you at an ASCII code terminal. When in NORMAL mode (specified by the NORMAL command), NOS displays the ASCII graphic 64-character set and interprets all input and output as display code. When in ASCII mode (specified by the ASCII command), NOS displays the ASCII 128-character set and interprets all input and output as 6/12 display code.

To determine the octal or hexadecimal ASCII code for a character, refer to table A-3. (Certain terminal definition commands require specification of an ASCII code.)

On output, the US code is reserved for network use and defined as an end-of-line. Use of this character, except in transparent mode, causes incorrect formatting and possible loss of output characters.

BATCH JOBS

Table A-2 lists the CDC graphic 64-character set, the ASCII graphic 64-character set, and the ASCII graphic 95-character sets. It also lists the code sets and card punch codes (026 and 029) that represent the characters.

The 64-character sets use display code as their code set; the 95-character set uses 12-bit ASCII code. The 95-character set is composed of all the characters in the ASCII 128-character set that can be printed at a line printer (refer to Line Printer Usage). Only 12-bit ASCII code files can be printed using the ASCII graphic 95-character set. To print a 6/12 display code file (usually created by an interactive job in ASCII mode), you must convert the file to 12-bit ASCII code. To do this, you enter the FCOPY command (section 9). The 95-character set is represented by 12-bit ASCII codes 0040₈ through 0176₈.

JOBS USING LINE PRINTERS

The batch character set printed depends on the print train used on the line printer to which the file is sent (refer to the ROUTE command in section 9). The following are the print trains corresponding to each of the batch character sets.

<u>Character Set</u>	<u>Print Train</u>
CDC graphic 64-character set	596-1
ASCII graphic 64-character set	596-5
ASCII graphic 95-character set	596-6

The characters of the default 596-1 print train are listed in the table A-2 column labeled CDC Graphic (64 Char); the 596-5 print train characters are listed in the table A-2 column labeled ASCII Graphic (64 Char); and the 596-6 print train characters are listed in the table A-2 column labeled ASCII Graphic (95 Char).

If a transmission error occurs when printing a line, the system prints the line again. The CDC graphic print train prints a concatenation symbol (↵) in the first printable column of a line containing errors. The ASCII print trains print an underline (__) instead of the concatenation symbol.

If an unprintable character exists in a line (that is, a 12-bit ASCII code outside the range 0040₈ through 0176₈), the number sign (#) appears in the first printable column of a print line, and a space replaces the unprintable character.

To route and correctly print a 6/12 display code file on a line printer with the ASCII graphic 95-character set, you must convert the 6/12 display code file to a 12-bit ASCII code file with the FCOPY command (refer to section 9). The resulting 12-bit ASCII file can be routed to a line printer (refer to the ROUTE command in section 9) but cannot be output at an interactive terminal.

Table A-1. Character Sets for Interactive Jobs (Sheet 1 of 2)

ASCII Graphic (64 Char)	ASCII Character (128 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code	ASCII Graphic (64 Char)	ASCII Character (128 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code
: colon †		00 †			# num. sign	# num. sign	60	60	0043
Display code 00 is undefined at sites using the 63-character set.					[l. bracket	[l. bracket	61	61	0133
A	A	01	01	0101] r. bracket] r. bracket	62	62	0135
B	B	02	02	0102	% †	% †	63 †	63 †	0045
C	C	03	03	0103	: colon	: colon	64	64	0042
D	D	04	04	0104	" quote	" quote	65	65	0137
E	E	05	05	0105	_ underline	_ underline	66	66	0041
F	F	06	06	0106	! †	! †	67	67	0046
G	G	07	07	0107	& ampersand	& ampersand			
					' apostrophe	' apostrophe	70	70	0047
H	H	10	10	0110	? †	? †	71	71	0077
I	I	11	11	0111	<	<	72	72	0074
J	J	12	12	0112	>	>	73	73	0076
K	K	13	13	0113	@	@	74		
L	L	14	14	0114	\ rev. slant	\ rev. slant	75	75	0134
M	M	15	15	0115	^ circumflex	^ circumflex	76		
N	N	16	16	0116	; semicolon	; semicolon	77	77	0073
O	O	17	17	0117		@		7401	0100
						^ circumflex		7402	0136
						: colon †		7404 †	0072
P	P	20	20	0120		X		7404	0045
Q	Q	21	21	0121		grave accent		7407	0140
R	R	22	22	0122					
S	S	23	23	0123		a		7601	0141
T	T	24	24	0124		b		7602	0142
U	U	25	25	0125		c		7603	0143
V	V	26	26	0126		d		7604	0144
W	W	27	27	0127		e		7605	0145
						f		7606	0146
X	X	30	30	0130		g		7607	0147
Y	Y	31	31	0131					
Z	Z	32	32	0132		h		7610	0150
0	0	33	33	0060		i		7611	0151
1	1	34	34	0061		j		7612	0152
2	2	35	35	0062		k		7613	0153
3	3	36	36	0063		l		7614	0154
4	4	37	37	0064		m		7615	0155
						n		7616	0156
						o		7617	0157
5	5	40	40	0065					
6	6	41	41	0066		p		7620	0160
7	7	42	42	0067		q		7621	0161
8	8	43	43	0070		r		7622	0162
9	9	44	44	0071		s		7623	0163
+	+	45	45	0053		t		7624	0164
-	-	46	46	0055		u		7625	0165
*	*	47	47	0052		v		7626	0166
						w		7627	0167
/	/	50	50	0057		x		7630	0170
((51	51	0050		y		7631	0171
))	52	52	0051		z		7632	0172
\$	\$	53	53	0044		{ left brace		7633	0173
=	=	54	54	0075		vert. line		7634	0174
space	space	55	55	0040		} right brace		7635	0175
, comma	, comma	56	56	0054		~ tilde		7636	0176
. period	. period	57	57	0056		DEL		7637	0177

† The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

Table A-1. Character Sets for Interactive Jobs (Sheet 2 of 2)

ASCII Graphic (64 Char)	ASCII Character (128 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code	ASCII Graphic (64 Char)	ASCII Character (128 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code
	NUL		7640	4000		DLE		7660	0020
	SOH		7641	0001		DC1		7661	0021
	STX		7642	0002		DC2		7662	0022
	ETX		7643	0003		DC3		7663	0023
	EOT		7644	0004		DC4		7664	0024
	ENQ		7645	0005		NAK		7665	0025
	ACK		7646	0006		SYN		7666	0026
	BEL		7647	0007		ETB		7667	0027
	BS		7650	0010		CAN		7670	0030
	HT		7651	0011		EM		7671	0031
	LF		7652	0012		SUB		7672	0032
	VT		7653	0013		ESC		7673	0033
	FF		7654	0014		FS		7674	0034
	CR		7655	0015		GS		7675	0035
	SO		7656	0016		RS		7676	0036
	SI		7657	0017		US †		7677	0037

† Reserved for network use. Refer to Character Set Tables in this appendix.

Table A-2. Character Sets for Batch Jobs (Sheet 1 of 3)

CDC Graphic (64 Char)	ASCII Graphic (64 Char)	ASCII Graphic (95 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code	Punch Code	
						026	029
: colon†	: colon†		00 †		.	8-2	8-2
Display code 00 is undefined at sites using the 63-character set.							
A	A	A	01	01	0101	12-1	12-1
B	B	B	02	02	0102	12-2	12-2
C	C	C	03	03	0103	12-3	12-3
D	D	D	04	04	0104	12-4	12-4
E	E	E	05	05	0105	12-5	12-5
F	F	F	06	06	0106	12-6	12-6
G	G	G	07	07	0107	12-7	12-7
H	H	H	10	10	0110	12-8	12-8
I	I	I	11	11	0111	12-9	12-9
J	J	J	12	12	0112	11-1	11-1
K	K	K	13	13	0113	11-2	11-2
L	L	L	14	14	0114	11-3	11-3
M	M	M	15	15	0115	11-4	11-4
N	N	N	16	16	0116	11-5	11-5
O	O	O	17	17	0117	11-6	11-6
P	P	P	20	20	0120	11-7	11-7
Q	Q	Q	21	21	0121	11-8	11-8
R	R	R	22	22	0122	11-9	11-9
S	S	S	23	23	0123	0-2	0-2
T	T	T	24	24	0124	0-3	0-3
U	U	U	25	25	0125	0-4	0-4
V	V	V	26	26	0126	0-5	0-5
W	W	W	27	27	0127	0-6	0-6
X	X	X	30	30	0130	0-7	0-7
Y	Y	Y	31	31	0131	0-8	0-8
Z	Z	Z	32	32	0132	0-9	0-9
0	0	0	33	33	0060	0	0
1	1	1	34	34	0061	1	1
2	2	2	35	35	0062	2	2
3	3	3	36	36	0063	3	3
4	4	4	37	37	0064	4	4
5	5	5	40	40	0065	5	5
6	6	6	41	41	0066	6	6
7	7	7	42	42	0067	7	7
8	8	8	43	43	0070	8	8
9	9	9	44	44	0071	9	9
+	+	+	45	45	0053	12	12-8-6
-	-	-	46	46	0055	11	11
*	*	*	47	47	0052	11-8-4	11-8-4

† The interpretation of this character or code depends on its context. Refer to Character Set Anomalies in this appendix.

Table A-2. Character Sets for Batch Jobs (Sheet 2 of 3)

CDC Graphic (64 Char)	ASCII Graphic (64 Char)	ASCII Graphic (95 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code	Punch Code	
						026	029
/	/	/	50	50	0057	0-1	0-1
(((51	51	0050	0-8-4	12-8-5
)))	52	52	0051	12-8-4	11-8-5
\$	\$	\$	53	53	0044	11-8-3	11-8-3
=	=	=	54	54	0075	8-3	8-6
space	space	space	55	55	0040	no punch	no punch
, comma	, comma	, comma	56	56	0054	0-8-3	0-8-3
. period	. period	. period	57	57	0056	12-8-3	12-8-3
≡ equiv.	# num. sign	# num. sign	60	60	0043	0-8-6	8-3
[l. bracket	[l. bracket	[l. bracket	61	61	0133	8-7	12-8-2†
] r. bracket] r. bracket] r. bracket	62	62	0135	0-8-2	11-8-2†
% †	% †	% †	63 †	63 †	0045	8-6	0-8-4
: colon	: colon	: colon	63	63	0072	8-2	8-2
≠	" quote	" quote	64	64	0042	8-4	8-7
→	underline	underline	65	65	0137	0-8-5	0-8-5
√	underline	underline	66	66	0041	11-0	12-8-7
^	& ampersand	& ampersand	67	67	0046	0-8-7	12
↑	' apostrophe	' apostrophe	70	70	0047	11-8-5	8-5
↓	? question	? question	71	71	0077	11-8-6	0-8-7
<	<	<	72	72	0074	12-0	12-8-4
>	>	>	73	73	0076	11-8-7	0-8-6
≤	@	@	74	74		8-5	8-4
≥	\ rev. slant	\ rev. slant	75	75	0134	12-8-5	0-8-2
ˆ	^ circumflex	^ circumflex	76	76		12-8-6	11-8-7
; semicolon	; semicolon	; semicolon	77	77	0073	12-8-7	11-8-6
		@		7401	0100		
		^ circumflex		7402	0136		
		: colon †		7404 †	0072		
		%		7404	0045		
		` grave accent		7407	0140		
		a		7601	0141		
		b		7602	0142		
		c		7603	0143		
		d		7604	0144		
		e		7605	0145		
		f		7606	0146		
		g		7607	0147		

†The interpretation of this character or code may depend on its context.
Refer to Character Set Anomalies elsewhere in this appendix.

Table A-2. Character Sets for Batch Jobs (Sheet 3 of 3)

CDC Graphic (64 Char)	ASCII Graphic (64 Char)	ASCII Graphic (95 Char)	Display Code	6/12 Display Code	12-Bit ASCII Code	Punch Code	
						026	029
		h		7610	0150		
		i		7611	0151		
		j		7612	0152		
		k		7613	0153		
		l		7614	0154		
		m		7615	0155		
		n		7616	0156		
		o		7617	0157		
		p		7620	0160		
		q		7621	0161		
		r		7622	0162		
		s		7623	0163		
		t		7624	0164		
		u		7625	0165		
		v		7626	0166		
		w		7627	0167		
		x		7630	0170		
		y		7631	0171		
		z		7632	0172		
		{ left brace		7633	0173		
		vert. line		7634	0174		
		} right brace		7635	0175		
		~ tilde		7636	0176		

Table A-3. ASCII to 6/12 Display Code Conversion (Sheet 1 of 2)

ASCII Character (128 Char)	12-Bit ASCII Code		6/12 Display Code	ASCII Character (128 Char)	12-Bit ASCII Code		6/12 Display Code
	Octal	Hex			Octal	Hex	
NUL	0000	00	7640	0	0060	30	33
SOH	0001	01	7641	1	0061	31	34
STX	0002	02	7642	2	0062	32	35
ETX	0003	03	7643	3	0063	33	36
EOT	0004	04	7644	4	0064	34	37
ENQ	0005	05	7645	5	0065	35	40
ACK	0006	06	7646	6	0066	36	41
BEL	0007	07	7647	7	0067	37	42
BS	0010	08	7650	8	0070	38	43
HT	0011	09	7651	9	0071	39	44
LF	0012	0A	7652	: colon ††	0072	3A	7404††
VT	0013	0B	7653	: colon	0072	3A	63
FF	0014	0C	7654	; semicolon	0073	3B	77
CR	0015	0D	7655	<	0074	3C	72
SO	0016	0E	7656	=	0075	3D	54
SI	0017	0F	7657	>	0076	3E	73
				?	0077	3F	71
DLE	0020	10	7660	@	0100	40	7401
DC1	0021	11	7661	A	0101	41	01
DC2	0022	12	7662	B	0102	42	02
DC3	0023	13	7663	C	0103	43	03
DC4	0024	14	7664	D	0104	44	04
NAK	0025	15	7665	E	0105	45	05
SYN	0026	16	7666	F	0106	46	06
ETB	0027	17	7667	G	0107	47	07
CAN	0030	18	7670	H	0110	48	10
EM	0031	19	7671	I	0111	49	11
SUB	0032	1A	7672	J	0112	4A	12
ESC	0033	1B	7673	K	0113	4B	13
FS	0034	1C	7674	L	0114	4C	14
GS	0035	1D	7675	M	0115	4D	15
RS	0036	1E	7676	N	0116	4E	16
US †	0037	1F	7677 †	O	0117	4F	17
space	0040	20	55	P	0120	50	20
!	0041	21	66	Q	0121	51	21
" quote	0042	22	64	R	0122	52	22
# number sign	0043	23	60	S	0123	53	23
\$	0044	24	53	T	0124	54	24
% ††	0045	25	63 ††	U	0125	55	25
%	0045	25	7404	V	0126	56	26
& ampersand	0046	26	67	W	0127	57	27
' apostrophe	0047	27	70	X	0130	58	30
(0050	28	51	Y	0131	59	31
)	0051	29	52	Z	0132	5A	32
*	0052	2A	47	[left bracket	0133	5B	61
+	0053	2B	45	\ reverse slant	0134	5C	75
, comma	0054	2C	56] right bracket	0135	5D	62
-	0055	2D	46	^ circumflex	0136	5E	7402
. period	0056	2E	57	_ underline	0137	5F	65
/	0057	2F	50				

† The interpretation of this character or code may depend on its text.

Refer to Character Set Anomalies elsewhere in this appendix.

†† Reserved for network use. Refer to Character Set Tables in this appendix.

Table A-3. ASCII to 6/12 Display Code Conversion (Sheet 2 of 2)

ASCII Character (128 Char)	12-Bit ASCII Code		6/12 Display Code	ASCII Character (128 Char)	12-Bit ASCII Code		6/12 Display Code
	Octal	Hex			Octal	Hex	
` grave accent	0140	60	7407	p	0160	70	7620
a	0141	61	7601	q	0161	71	7621
b	0142	62	7602	r	0162	72	7622
c	0143	63	7603	s	0163	73	7623
d	0144	64	7604	t	0164	74	7624
e	0145	65	7605	u	0165	75	7625
f	0146	66	7606	v	0166	76	7626
g	0147	67	7607	w	0167	77	7627
h	0150	68	7610	x	0170	78	7630
i	0151	69	7611	y	0171	79	7631
j	0152	6A	7612	z	0172	7A	7632
k	0153	6B	7613	{ left brace	0173	7B	7633
l	0154	6C	7614	vertical line	0174	7C	7634
m	0155	6D	7615	} right brace	0175	7D	7635
n	0156	6E	7616	~ tilde	0176	7E	7636
o	0157	6F	7617	DEL	0177	7F	7637

JOBS USING MAGNETIC TAPES

Coded data to be copied from mass storage to magnetic tape is assumed to be represented in display code. NOS converts the data to external BCD code when writing a coded seven-track tape and to ASCII or EBCDIC code (as specified on the tape assignment command) when writing a coded nine-track tape.

Because only 63 characters can be represented in seven-track even parity, one of the 64 display codes is lost in conversion to and from external BCD code. The following shows the differences in conversion depending on the character set (63 or 64) which the system uses. The ASCII character for the specified character code is shown in parentheses. The output arrow shows how the display code changes when it is written on tape in external BCD. The input arrow shows how the external BCD code changes when the tape is read and converted to display code.

<u>63-Character Set</u>				
<u>Display Code</u>		<u>External BCD</u>		<u>Display Code</u>
00		16 (%)		00
33 (0)	Output	12 (0)	Input	33 (0)
63 (:)	→	12 (0)	→	33 (0)

<u>64-Character Set</u>				
<u>Display Code</u>		<u>External BCD</u>		<u>Display Code</u>
00 (:)		12 (0)		33 (0)
33 (0)	Output	12 (0)	Input	33 (0)
63 (%)	→	16 (%)	→	63 (%)

If a lowercase ASCII or EBCDIC code is read from a nine-track coded tape, it is converted to its uppercase 6-bit display code equivalent. To read and write lowercase ASCII or EBCDIC characters, you must assign the tape in binary mode and write a program to convert the binary data.

Tables A-4 and A-5 show the character set conversion for nine-track tapes. Table A-4 lists the conversions to and from the ASCII character code and display code. Table A-5 lists the conversions between the EBCDIC character code and the display code. Table A-6 shows the character set conversions between external BCD and display code for seven-track tapes.

Table A-4. Nine-Track ASCII Coded Tape Conversion

ASCII				Display Code		ASCII				Display Code	
Code Conversion†		Character and Code Conversion††				Code Conversion†		Character and Code Conversion††			
Code (Hex)	Char	Code (Hex)	Char	ASCII Char	Code (Octal)	Code (Hex)	Char	Code (Hex)	Char	ASCII Char	Code (Octal)
20	space	00	NUL	space	55	3E	>	1E	RS	>	73
21	!	7D	}	!	66	3F	?	1F	US	?	71
22	"	02	STX	"	64	40	@	60	`	@	74
23	#	03	ETX	#	60	41	A	61	a	A	01
24	\$	04	EOT	\$	53	42	B	62	b	B	02
25	%	05	ENQ	%	63	43	C	63	c	C	03
25	%	05	ENQ	space†††	55	44	D	64	d	D	04
26	&	06	ACK	&	67	45	E	65	e	E	05
27	'	07	BEL	'	70	46	F	66	f	F	06
28	(08	BS	(51	47	G	67	g	G	07
29)	09	HT)	52	48	H	68	h	H	10
2A	*	0A	LF	*	47	49	I	69	i	I	11
2B	+	0B	VT	+	45	4A	J	6A	j	J	12
2C	,	0C	FF	,	56	4B	K	6B	k	K	13
2D	-	0D	CR	-	46	4C	L	6C	l	L	14
2E	.	0E	SO	.	57	4D	M	6D	m	M	15
2F	/	0F	SI	/	50	4E	N	6E	n	N	16
30	0	10	DLE	0	33	4F	O	6F	o	O	17
31	i	11	DC1	i	34	50	P	70	p	P	20
32	2	12	DC2	2	35	51	Q	71	q	Q	21
33	3	13	DC3	3	36	52	R	72	r	R	22
34	4	14	DC4	4	37	53	S	73	s	S	23
35	5	15	NAK	5	40	54	T	74	t	T	24
36	6	16	SYN	6	41	55	U	75	u	U	25
37	7	17	ETB	7	42	56	V	76	v	V	26
38	8	18	CAN	8	43	57	W	77	w	W	27
39	9	19	EM	9	44	58	X	78	x	X	30
3A	:	1A	SUB	:	00	59	Y	79	y	Y	31
Display code 00 is undefined at sites using the 63-character set.				:	63	5A	Z	7A	z	Z	32
3A	:	1A	SUB	:	63	5B	[1C	FS	[61
3B	;	1B	ESC	;	77	5C	\	7C		\	75
3C	<	7B	{	<	72	5D]	01	SOH]	62
3D	=	1D	GS	=	54	5E	^	7E	~	^	76
						5F	_	7F	DEL	_	65

†When these characters are copied from/to a tape, the characters remain the same and the code changes from/to ASCII to/from display code.

††These characters do not exist in display code. Therefore, when the characters are copied from a tape, each ASCII character is changed to an alternate display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 61₁₆, from tape, it writes an uppercase A, 01₈.

†††A display code space always translates to an ASCII space.

Table A-5. Nine-Track EBCDIC Coded Tape Conversion

EBCDIC				Display Code		EBCDIC				Display Code	
Code Conversion†		Character and Code Conversion††				Code Conversion†		Character and Code Conversion††			
Code (Hex)	Char	Code (Hex)	Char	ASCII Char	Code (Octal)	Code (Hex)	Char	Code (Hex)	Char	ASCII Char	Code (Octal)
40	space	00	NUL	space	55	C4	D	84	d	D	04
4A	¢	1C	IFS	[61	C5	E	85	e	E	05
4B	.	0E	SO	.	57	C6	F	86	f	F	06
4C	<	C0	{	<	72	C7	G	87	g	G	07
4D	(16	BS	(51	C8	H	88	h	H	10
4E	+	0B	VT	+	45	C9	I	89	i	I	11
4F		D0	}	!	66	D1	J	91	j	J	12
50	&	2E	ACK	&	67	D2	K	92	k	K	13
5A	!	01	SOH]	62	D3	L	93	l	L	14
5B	\$	37	EOT	\$	53	D4	M	94	m	M	15
5C	*	25	LF	*	47	D5	N	95	n	N	16
5D)	05	HT)	52	D6	O	96	o	O	17
5E	;	27	ESC	;	77	D7	P	97	p	P	20
5F	-	A1	~	~	76	D8	Q	98	q	Q	21
60	/	0D	CR	-	46	D9	R	99	r	R	22
61	/	0F	SI	/	50	E0	\	6A	l	\	75
6B	,	0C	FF	,	56	E2	S	A2	s	S	23
6C	%	2D	ENQ	%	63	E3	T	A3	t	T	24
6C	%	2D	ENQ	space†††	55	E4	U	A4	u	U	25
6D		07	DEL		65	E5	V	A5	v	V	26
6E	>	1E	IRS	>	73	E6	W	A6	w	W	27
6F	?	1F	IUS	?	71	E7	X	A7	x	X	30
7A	:	3F	SUB	:	00	E8	Y	A8	y	Y	31
Display code 00 is undefined at sites using the 63-character set.											
7A	:	3F	SUB	:	63	E9	Z	A9	z	Z	32
7B	#	03	ETX	#	60	F0	0	10	DLE	0	33
7C	@	79	\	@	74	F1	1	11	DC1	1	34
7D	'	2F	BEL	'	70	F2	2	12	DC2	2	35
7E	=	1D	IGS	=	54	F3	3	13	TM	3	36
7F	"	02	STX	"	64	F4	4	3C	DC4	4	37
C1	A	81	a	A	01	F5	5	3D	NAK	5	40
C2	B	82	b	B	02	F6	6	32	SYN	6	41
C3	C	83	c	C	03	F7	7	26	ETB	7	42
						F8	8	18	CAN	8	43
						F9	9	19	EM	9	44

†When these characters are copied from/to a tape, the characters remain the same (except EBCDIC codes 4A, 4F, 5A, and 5F) and the code changes from/to EBCDIC to/from display code.

††These characters do not exist in display code. Therefore, when the characters are copied from a tape, each EBCDIC character is changed to an alternate display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 81₁₆, from tape, it writes an uppercase A, 01_g.

†††All EBCDIC codes not listed translate to display code 55_g (space). A display code space always translates to an EBCDIC space.

Table A-6. Seven-Track Coded Tape Conversions

External BCD	ASCII Character	Octal Display Code	External BCD	ASCII Character	Octal Display Code
01	1	34	40	-	46
02	2	35	41	J	12
03	3	36	42	K	13
04	4	37	43	L	14
05	5	40	44	M	15
06	6	41	45	N	16
07	7	42	46	O	17
10	8	43	47	P	20
11	9	44	50	Q	21
12†	0	33	51	R	22
13	=	54	52	!	66
14	"	64	53	\$	53
15	@	74	54	*	47
16†	%	63	55	'	70
17	[61	56	?	71
20	space	55	57	>	73
21	/	50	60	+	45
22	S	23	61	A	01
23	T	24	62	B	02
24	U	25	63	C	03
25	V	26	64	D	04
26	W	27	65	E	05
27	X	30	66	F	06
30	Y	31	67	G	07
31	Z	32	70	H	10
32]	62	71	I	11
33	,	56	72	<	72
34	(51	73	.	57
35	_	65	74)	52
36	#	60	75	\	75
37	&	67	76	^	76
			77	;	77

† As explained previously in this section, conversion of these codes depends on whether the tape is being read or written.

DIAGNOSTIC MESSAGES

B

The appendix contains an alphabetical listing of the messages that may appear in your dayfile. Lowercase characters are used to identify variable names or fields. All messages are sorted according to the first nonvariable word or character. Messages beginning with special characters (such as hyphens or asterisks) are sorted as if the special characters were not present. For example, the message

filename ALREADY PERMANENT.

is listed with the messages beginning with A, and the message

****ENDING SUPPORT OF LFM FUNCTION nnB.

is listed with the messages beginning with E.

Dayfile messages usually issued only to COMPASS programs are listed in appendix B of Volume 4, Program Interface.

If you encounter a diagnostic or informative message that does not appear in this appendix, consult the NOS 2 Diagnostic Index. This publication catalogs all messages produced by NOS and its products and specifies the manual or manuals in which each message is fully documented.

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ACCOUNT BLOCK LIMIT.	The monitor detected the expiration of the account block SRU limit.	Reset account block SRU limit with SETASL command or macro. If the account block limit is set at its maximum, issue another CHARGE command to begin a new account block.	1AJ
ADDRESS OUT OF RANGE.	An address in a parameter block is outside the job's field length.	Specify parameter block address within field length.	LFM
ADDRESS OUT OF RANGE addr.	LOC read an address addr on a correction command that is greater than or equal to the user's field length. The correction command is ignored and LOC continues.	Correct the correction command(s) and retry.	CPMEM
filenam ALREADY PERMANENT.	You have already saved or defined a file with the name specified.	Save or define file using different file name or purge existing file.	PFM
APPENDED - type/name	The record with type and name on the replacement file was not matched; it has been appended to the new file. A option selected.	None.	COPYL COPYLM
APPLICATION FAILED.	The application has ceased functioning and you can no longer access it.	Access another application or log off and try later.	NETVAL
APPLICATION NOT PRESENT.	The requested application is currently not available for use.	Enter the name of another application or log off and try later.	NETVAL
APPLICATION RETRY LIMIT.	Four unsuccessful attempts were made to enter a legal application name, after which the terminal is disconnected.	Ensure the accuracy of the entry. If problems persist, contact installation personnel.	NETVAL
ARA BURST DEFFECTIVE.	First block of tape at 1600 or 6250 cpi cannot be read or written. Another unit or tape should be tried. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
ARG. ERROR.	LDR parameters were outside your job's field length or the central memory address in call is not within your job's field length.	Examine program to determine error.	LDR
ARGUMENT ERROR.	The command has an invalid parameter or an incorrect format.	Reenter the command in the correct format and with valid parameter specifications.	RECOVER
ARGUMENT ERROR.	An incorrect parameter was entered on a command.	Reenter the command using the correct parameters and syntax.	FOTD SUBSYST
ARGUMENT ERROR.	The command syntax is incorrect. When the system processes tape management commands, 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 a tape assignment command. The address parameter on DMPECS, DMDECS, LBC, LOC, PBC, or WBR must be numeric.	Recheck parameters.	CONVERT COPY COPYBR COPYBF COPYEI COPYX CPMEM LO72 RESEX TCOPY
ARGUMENT ERRORS.	The ENQUIRE command's syntax is incorrect.	Check parameters on command and retry.	ENQUIRE
ARITHMETIC INDEFINITE.	The CPU floating-point arithmetic unit attempted to use an indefinite operand.	Analyze the job output and dumps to determine the cause.	1AJ
ARITHMETIC OVERFLOW.	The CPU floating-point arithmetic unit received an operand too large for computation.	Analyze the job output and dumps to determine the cause.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ARITHMETIC UNDERFLOW.	The CPU floating-point arithmetic unit received an operand too small for computation.	Analyze the job output and dumps to determine the cause.	1AJ
eq, ASSIGNED TO lfn.	Informative message specifying equipment type assigned to file lfn.	None.	LFM
jsn ASSIGNS EXCEED DEMANDS.	RESEX internal problems occurred. The resources actually assigned to the job with the specified job sequence name exceed the resources demanded on a RESOURC command.	Inform site analyst.	RESEX
n BAD FORMAT BLOCKS.	Block(s) not meeting X or SI-coded conversion requirements have been encountered on input.	Check tape format for errors.	TCOPY
BINARY SEQ. ERROR, RECxxxx CDyyyy.	A binary card was found to be out of sequence and the job was terminated without EXIT processing. Error is on card yyyy (octal) of record xxxx (octal).	Examine sequencing of binary deck and correct error.	1AJ
BLANK LABELS DO NOT VERIFY.	The tape labels written on the blank tape could not be verified, probably due to hardware failure.	Request that the operator blank label the tape using a different tape drive.	BLANK
BLANK TAPE, lfn AT addr.	A blank tape was read. Blank tape is defined as more than 25 feet of erased tape.	Ensure correct tape is specified on command.	1MT
BLOCK SEQUENCE ERROR, lfn AT addr.	The block length recorded in the file did not match the length of the block read, or the block number recorded in the file did not match the system block count (this message applies to I format tapes only).	Ensure accuracy of format parameter (F) on command or macro.	1MT
BLOCK SIZE NOT APPLICABLE.	Either the CC or BS parameter was specified on a COPY command for which neither the input nor output file was an S or L format tape or the CC parameter was specified on a TCOPY command without the E or B conversion parameter specified.	Refer to the COPY or TCOPY command description, correct the command, and retry.	COPY TCOPY
BLOCK SIZE TOO LARGE ON filename.	S tape block size exceeds maximum for that format (1000B) and COPY or character count for E or B tape TCOPY exceeds maximum for that format (5120 characters).	Reduce the block size or character counts.	COPY TCOPY
BLOCK SIZE TOO SMALL ON filename.	On F to F tape copy the maximum frame count for the first file exceeds that of the second file. On S and L tape copy, block size is less than noise size. On E and B tape TCOPY, block size is less than the corresponding S tape noise size.	Increase block size so that it is greater than the noise size. On F to F tape copy, increase output file block size so it is greater than the block size of the input file.	COPY TCOPY
BLOCK TOO LARGE, lfn AT addr.	The tape being read contained a data block greater in size than that allowed by the specified format or by your specification.	Ensure accuracy of format parameter (F) on command or macro.	1MT
BOT/EOT ENCOUNTERED, lfn AT addr.	Indicates an abnormal tape position.	Inform site analyst if persistent.	1MT
BREAKPOINT CONDITION.	The job executed an address for which a breakpoint was requested by the system.	Inform site analyst.	1AJ
BUFFER ARG. ERROR.	CM address in call is not less than the field length minus the word count; buffer extends past the job's field length.	Verify that operation does not reference address beyond end of buffer or job's field length.	TCS
BUFFER ARG. ERROR	Central memory address in call is not within the job's field length.	Correct call and retry.	TCS
BUFFER ARGUMENT ERROR, lfn AT addr.	For tape operations, this message indicates one of the following. - FET less than 7 words long for S/L format - MLRS greater than 1000 octal for S format - POSMF issued and no HDR1 label found in FET or extended label buffer Refer to Volume 4.	Examine program to determine error.	1MT
BUFFER ARGUMENT ERROR ON lfn AT addr.	A buffer pointer did not conform to the following constraints. - FIRST .LE. IN - FIRST .LE. OUT - IN AND OUT .LT. LIMIT .LE. FL Refer to Volume 4.	Examine program to determine error in buffer pointers.	CIO

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
BUFFER CONTROL WORD ERROR.	Dayfile message indicating that the word count in the disk linkage is greater than 100B.	Inform site analyst.	SLL
BUFFER CONTROL WORD ERROR, lfn AT addr.	Either an attempt was made to write a block smaller than the noise size on an S, L, or F format tape, or a control word error occurred in a write (such as bad byte count).	Examine program to determine error.	1MT
BUFFER SPACE ERROR (TOV) pfn BUSY.	Internal error. 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.	Notify site analyst. Reissue ATTACH until file becomes available, or issue ATTACH specifying NA option.	IAF PFM
CALLING JOB MAY NOT DROP ITSELF.	You attempted to drop your current job.	Do not attempt to drop your current job.	DROP
CARD READER ERROR.	Job was terminated without exit processing.	Resubmit job.	1AJ
CATALOG ARGUMENT ERROR.	The syntax of the CATALOG command is incorrect.	Check the CATALOG command description and correct the command.	CATALOG
CATALOG COMPLETE.	Informative message indicating that catalog processing is complete.	None.	CATALOG MODVAL PFATC
CATALOG FILE NAME CONFLICT.	The same file was named as the file to be cataloged and as the file onto which the catalog is written.	Change one of the file names on the CATALOG command.	CATALOG
CATLIST COMPLETE.	Informative message indicating a successful completion of the CATLIST command.	None.	CATLIST
CCL100-SEPARATOR FOLLOWING VERB MUST BE COMMA OR LEFT PARENTHESIS	Fatal user error. Separator following verb in a command must be a comma or a left parenthesis.	Change separator following verb to a comma or a left parenthesis.	CCL
CCL101-LAST NON-BLANK CHARACTER MUST BE SEPARATOR	Fatal user error. Last character string of line was not followed by a separator or a terminator.	To terminate command, make last nonblank character a period or right parenthesis. To continue command on next card or line, make last nonblank character a valid separator.	CCL
CCL102-EQUAL SIGN MUST FOLLOW FIRST SYMBOLIC NAME	Fatal user error. First parameter following a SET verb is a symbolic name to be set. An equal sign must follow the symbolic name.	Change separator following the symbolic name to an equal sign. Equal sign must be followed by an expression.	CCL
CCL103-STATEMENT INCOMPLETE	Fatal user error. A terminator was detected immediately following a verb.	Check command format (refer to section 6) and rewrite command, using a comma or left parenthesis after verb.	CCL
CCL104-EXPRESSION AND STATEMENT TERMINATED BY t	Fatal user error. An expression or a verb was followed by a command terminator (period or right parenthesis) instead of a comma. Any of following conditions may produce this error. This message is informative and is issued in conjunction with message CCL123. - The label string parameter was omitted. - A relational or logical operator was delimited by a period on left side only. - An unmatched right parenthesis appeared in expression. If separator following verb was a left parenthesis, an unmatched right parenthesis may appear matched. For example, IFE(R1+R2)=3, LABEL. is terminated by right parenthesis.	- Place a comma after expression or verb and add label string parameter. - Delimit operator with periods on both sides. - If error was caused by a misleading left parenthesis following verb, add a comma immediately after verb. Otherwise, recheck all parentheses and match unmatched right parenthesis.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL105-TERMINATOR MUST FOLLOW SUBSYSTEM NAME	Fatal user error. Improper formatting appears after the symbolic name SS.	Place a period in the appropriate location.	CCL
CCL120-ALL CARDS SKIPPED - xxxxxx	Fatal user error. In process of skipping, end of command record was reached. Any of following conditions may produce this error. - An IFE, SKIP, or ELSE expression was false and no ENDIF command terminated resultant skip. - A WHILE expression was false and no ENDW command terminated resultant skip. - An ENDW command was not preceded by a WHILE command.	- Either add an ENDIF command with a matching label string or correct existing ENDIF command. - Either add an ENDW command with a matching label string or correct existing ENDW command so that WHILE and ENDW label strings match. - Precede ENDW command with a WHILE command.	CCL
CCL121-LABEL STRING MUST BEGIN WITH ALPHA CHARACTER	Fatal user error. First character of a label string must not be numeric.	Change label string to a label string of 1 to 7 alphanumeric characters, beginning with an alphabetic character.	CCL
CCL123-ALPHANUMERIC LABEL STRING REQUIRED	Fatal user error. The commands ELSE, ENDIF, ENDW, IFE, SKIP, and WHILE require label string parameter. This message is issued in conjunction with the message CCL104.	Refer to message CCL104 for cause and suggested action.	CCL
CCL124-PRECEDING ERR MSG. APPLIES TO FOLLOWING	Informative message. Dayfile message which precedes CCL124 is an error message which was caused by the command printed after CCL124.	Refer to error message which precedes this in the dayfile for cause and suggested action.	CCL
CCL125-NUMBER OF CARDS SUPPRESSED=x	Informative message. A command which the system is printing from an internal buffer continued over more lines of input than buffer size. The system prints the first three lines and then prints this message to indicate the number of lines suppressed, x. Remaining lines of the command follow this message.	None.	CCL
CCL126-STATEMENT TERMINATOR MUST FOLLOW LABEL	Fatal user error. Either of the following conditions may produce this error. - A label string is last parameter on an ELSE, ENDIF, ENDW, IFE, SKIP, or WHILE command. It must be followed by a period or a right parenthesis. - An expression contained a comma, and character string that followed was assumed to be a label string.	- Terminate command with a period or a right parenthesis. - Either remove comma or replace it with a left parenthesis, depending on nature of error.	CCL
CCL127-LABEL STRING EXCEEDS xx CHARACTERS	Fatal user error. Label string parameter on an ELSE, ENDIF, ENDW, IFE, SKIP, or WHILE command is greater than xx characters. xx is maximum number of characters defined by installation.	Change label string on any command it appears to a label string with xx or fewer alphanumeric characters.	CCL
CCL150-EXPRESSION CONTAINS INVALID OPERATOR x	Fatal user error. x is not a legal operator in preceding command.	If x is part of a character string, \$-delimit character string. If x is an intended operator, refer to section 6 for a list of valid operators.	CCL
CCL151-EXPRESSION FORMAT ERROR (IN FILE FUNCTION)	Fatal user error. Format of expression is illegal. Probable cause is use of commas or unmatched left parenthesis within an expression. If error was detected within a FILE function, IN FILE FUNCTION is included in message. FILE function must be of form FILE(lfn,exp), where exp is a valid function or symbolic name.	Check parentheses and match unmatched left parenthesis or remove excess commas from FILE function.	CCL
CCL152-POORLY FORMED FUNCTION (NUM, DT, SS)	Fatal user error with following causes. - Separator after function name is not a left parenthesis or function is not terminated by a right parenthesis. - Character string evaluated by function contained special characters but was not \$-delimited.	- Correct format of function. - \$-delimit character string.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL153-ILLEGAL FUNCTION CALL WITHIN FILE FUNCTION	Fatal user error. The NUM function, FILE function, or SS symbolic name appears within a FILE function.	Restructure expression so NUM, SS, or FILE is not within a FILE function.	CCL
CCL154-ILLEGAL EXPONENT	Fatal user error. A negative exponent in a expression is invalid.	Remove minus sign preceding exponent.	CCL
CCL155-OPERATOR OR OPERAND SEQUENCE ERROR	Fatal user error with following causes. - Sequence of adjacent operators is illegal. For example, 3*-4 is not allowed. - An operand is adjacent to a left or right parenthesis. Implied multiplication is not allowed. For example, 3(R+1) is invalid.	- Use parentheses to separate adjacent operators. - Use an operator to separate operand and parenthesis.	CCL
CCL156-STRING TOO LONG -strng	Fatal user error. Listed character string is too long. A character string may not be longer than 10 characters, excluding \$ delimiters for literals.	Shorten character string to 10 characters or less.	CCL
CCL157-UNKNOWN NAME -strng	Fatal user error. Alphanumeric character string strng is not a symbolic name recognized by the system.	Replace strng with a valid symbolic name. Refer to Section 6 for a list of valid symbolic names.	CCL
CCL158-OPERATOR/TERMINATOR MUST FOLLOW FUNCTION (DT, FILE, NUM)	Fatal user error. A function was properly formed and positioned within a command, but it was not followed by an operator, separator, or terminator.	Place an operator, separator, or terminator after function.	CCL
CCL159-FIRST PARAMETER INVALID IN SET STATEMENT	Fatal user error. First parameter in SET statement was not one of following symbolic names: DSC, EF, EFG, R1, R2, R3, R16, or SS.	Replace first parameter with a valid symbolic name.	CCL
CCL160-NUMERIC OR LITERAL IN FILE FUNCTION	Fatal user error. Expression of a FILE function may contain only symbolic names defined for function.	Remove numeric or literal character string. Refer to section 6 for a list of FILE symbolic names.	CCL
CCL161-STACK OVERFLOW	Fatal user error. Evaluation of an expression overflowed either operator or operand stack.	Check expression for errors and correct. If no errors, either simplify expression or break it apart so that two or more commands have same effect as one.	CCL
CCL162 - PERIOD BRACKETED OPERATORS PREFERRED.	Informative message indicating that a single character operator was used in an expression. (i.e., > instead of .GT.).	Use period bracketed operators.	CCL
CCL163-SUBSYSTEM REFERENCE ERROR	The system does not recognize the name associated with the SS.	Refer to section 6 for a list of valid names.	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 calls 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 procedure call was greater than 40 characters. - pfile parameter on procedure call was null. Your site does not have a default for pfile.	- 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. 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, the system discovered a keyword x on the procedure call that was not specified on the procedure header directive.	Remove keyword x from procedure call.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL206-SYMBOLIC SPECIFICATION INVALID xxx+	Fatal user error. A parameter, xxx, on procedure call is followed by a plus sign. Plus sign indicates that the system 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 valid 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 xxx with a valid symbolic name. Refer to section 6 for a list of valid symbolic names.	CCL
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 or default value on a procedure header directive is greater than xx characters. xx is defined by installation.	Specify a value with xx or fewer characters.	CCL
CCL212-SEPARATOR INVALID string s	Fatal user error. s is invalid separator and string is character string that precedes s. Any of following conditions produces this error. - On a procedure call, 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 invalid separator s, the string is null. - #DATA or #FILE has been specified on the procedure call.	- Change separator s to a comma. - \$-delimit character string. - Remove #DATA or #FILE from procedure call and specify it on procedure header directive.	CCL
CCL230-PROCEDURE FILE NOT FOUND	Fatal user error. The file named on the procedure call was not assigned to the job and the installation chose 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 procedure call.	CCL
CCL231-PROCEDURE NOT FOUND	Fatal user error. Local or permanent file indicated on procedure call was found, but the system could not find procedure on the file.	Check procedure name for errors and correct. If name is correct, check file for procedure.	CCL
CCL233 - PROC HEADER MUST LEAD LIBRARY PARTITION/PNAME MUST EQ PARTITION NAME.	The name of the procedure must match a partition in the library where the procedure resides. The procedure header must be the first entry in the partition.	Rebuild library with the procedure header as the first entry in the named partition.	CCL
CCL234-UNABLE TO LOCATE LIBRARY PARTITION xxx	Fatal user error. The system could not find procedure xxx, specified on call-by-name command, on currently defined library set.	Check procedure name for errors and correct. If name is correct, check library for procedure.	CCL
CCL235-FORMAL PARAMETER GT xx CHARACTERS	Fatal user error. Number of characters in a keyword on procedure header directive exceeds xx, as defined by installation.	Define a keyword with xx or fewer characters.	CCL
CCL236-SPECIAL DEFAULT SPECIFICATION UNKNOWN	Fatal user error. Equivalence symbol, which specifies a special default option on procedure header directive, must be followed by either FILE or DATA.	If equivalence symbol is part of a character string, \$-delimit character string. If a special default option is desired, follow equivalence symbol with either DATA or FILE.	CCL
CCL237-SEPARATOR FOLLOWING SECOND DEFAULT IS */*	Fatal user error. In form keyword=default1/ default2/ on procedure header directive, 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 directive has exceeded xx, as defined by installation.	Remove parameters from procedure header directive until xx or fewer parameters remain.	CCL
CCL239-PROCEDURE HEADER NOT TERMINATED	Fatal user error. Procedure header directive was not terminated by a period, and no commands were found after header directive.	Terminate header Procedure with a period, and add at least one command to procedure.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL249 - DATA FILE LFN EXCEEDS 7 CHARACTERS - lfn	The file name lfn specified on a .DATA directive is longer than seven characters.	Specify a shorter file name.	CCL
CCL250-CONCATENATED STRING EXCEEDS 80 CHARACTERS	Fatal user error. Use of a right arrow (or underline character in ASCII) produced a linked string of more than 80 characters, or a string of 80 characters with a following separator.	Reduce number of linked characters.	CCL
CCL251-DATA COMMAND SPECIFIED CCL FILEfilenam	Fatal user error. File name that you specified on a .DATA directive is a working file.	Select another file name.	CCL
CCL252-PROCEDURE CONTAINS NO CONTROL STATEMENTS	Informative message. A procedure should contain at least one command.	None.	CCL
CCL261-ENDW WITH MATCHING LABEL FOUND	Nonfatal user error. The system encountered an ENDW command in the job command record before finding a WHILE command with a matching label string. A search is initiated for a corresponding WHILE command. If a WHILE command with a matching label string is found and WHILE expression is true, normal processing continues with command following WHILE command. If WHILE expression is false or if no WHILE command with a matching label string is found, the system skips all remaining commands in command record.	Check for a WHILE command and if not present, add. If present, see if WHILE and ENDW label string match and are correct. If they match, place WHILE command before ENDW command.	CCL
CCL262-CONTINUING SEARCH, PRINTING SKIPPED CARDS	Informative message. To assist in isolation of WHILE command errors such as CCL261, all skipped commands will be printed in the job dayfile until WHILE search is complete.	None.	CCL
CCL263 EXTERNAL ABORT DURING BEGIN.	An external abort during BEGIN processing occurred. Check the job dayfile to find where error occurred.	Correct BEGIN command or PROC directive as indicated by dayfile.	CCL
CCL263 EXTERNAL ABORT DURING ENDW.	An external abort occurred during CCL search for the WHILE command. The abort is due to a DROP or RERUN command an exceeded time limit, mass storage limit, or extended memory limit.	Check validation limits. Modify job to fit within limits or ask site to increase the limits.	CCL
CCL270-ERR IN CCL WORK FILES, REVERT TO JOB FILE	Fatal user error. An error was encountered in a file used by the system because of user manipulation of procedures or data. The system attempts to return to your assigned job file. If unsuccessful, the job terminates immediately. If successful, control goes to the EXIT command.	Rewrite the job and/or procedures to prevent manipulation of the system work files.	CCL
CCL271-REVERT NOT ALLOWED WITHIN JOB FILE	Fatal user error. A REVERT command cannot appear in commands of job file.	Remove REVERT command.	CCL
CCL272-INVALID REVERT PARAMETER - xxx	Fatal user error. Parameter xxx was used on REVERT command. REVERT command allows only ABORT parameter or no parameters.	Remove parameter xxx or replace with ABORT.	CCL
CCL273 - EXPECTING . OR) AFTER REVERT,EX	REVERT,EX will not accept any other parameters. The command must be terminated and a valid command must immediately follow the terminator.	Correct command and resubmit.	CCL
CCL300-NON-NUMERIC CHARACTER WITHIN NUMERIC TERM	Fatal user error. An element of an expression which begins with a numeric character cannot contain a nonnumeric character except B or D as a post radix.	Remove any nonnumeric characters other than a post radix B or D.	CCL
CCL302-8/9 DIGIT CONFLICT WITH POST RADIX OF B	Fatal user error. An 8 or 9 is invalid in an octal number.	Remove any 8's or 9's if number is octal or remove post radix B if number is decimal.	CCL
CCL303-LITERAL NOT TERMINATED	Fatal user error. A literal which begins with a dollar sign was not terminated by a second dollar sign.	Place a second dollar sign at end of character string denoted as a literal.	CCL
CCL304-STRCL - SCATTER BUFFER HEADER INVALID	Fatal system error.	Inform site analyst.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL320 - ERROR...RE-ENTER CCL320 - ENTER	This message precedes a system prompt for a parameter when a procedure call contains a parameter entry in error.	None.	CCL
CCL321 - EXPECTING (AFTER PARAMETER= ON PROC HEADER	In a parameter-prompting procedure, the equal sign after a parameter name or description must be followed by a parenthesis to indicate the start of a checklist.	Correct procedure header and resubmit.	CCL
CCL322 - END HEADER WITH . OR USE , / ? BEFORE PARAM	The *I after a procedure name or a parameter definition must be followed by a period, comma, slant, or reverse slant.	Correct procedure header and resubmit.	CCL
CCL323 - EXCESS PATTERNS/VALUES - PROC HEADER	The sum of checklist values and patterns is too large (exceeded the internal table size of CCL, which is an installation option).	Reduce the number of checklist options or have CCL reassembled and installed with an increased value for installation parameter IP.NPV.	CCL
CCL324 - CHECKLIST PATTERN MUST BE *N *K *F *A *S	The asterisk in a checklist pattern was not immediately followed by N, K, F, A, or S.	Correct procedure header and resubmit.	CCL
CCL325 - *SN(SET) N MUST BE .LE. 40 and .GE. 1	The number of characters specified for a set was more than 40 or specified as a zero (*S0(SET)).	Correct procedure header and resubmit.	CCL
CCL326 - I DOES NOT FOLLOW * AFTER PNAME ON HEADER	The I was probably omitted from *I following the procedure name.	Correct procedure header and resubmit.	CCL
CCL327 - EXPECTING VALUE AFTER PATTERN = ON HEADER	The system has encountered unrecognizable syntax immediately following the equal sign in the checklist. The system expects an alphanumeric value, a literal, or a special name as a value.	Correct header and resubmit.	CCL
CCL328 - HEADER PARAMETER DESCRIPTION .GT. 40	A parameter description - the string in quotes immediately following a parameter name - was more than 40 characters long.	Shorten the string to 40 or less characters and resubmit.	CCL
CCL329 - MISC. AFTER HEADER PARAMETER DESCRIPTION	The only characters allowed to follow the terminating quotation mark of a parameter description are equal sign, period, right parenthesis, slant, reverse slant, or comma.	Correct procedure header and resubmit.	CCL
CCL330 - EXPECTED , = OR) AFTER CHECKLIST PATTERN	The pattern part of a checklist item can only be followed by another pattern preceded by a comma, a replacement value preceded by an equal sign, or a right parenthesis to indicate the end of the checklist.	Correct procedure header and resubmit.	CCL
CCL331 - EXPECTED , OR) AFTER PATTERN VALUE	The replacement value part of a checklist item can only be followed by another pattern preceded by a comma or a right parenthesis to indicate the end of a checklist.	Correct procedure header and resubmit.	CCL
CCL332 - WRITE SET AS *SN(SET) ON PROC HEADER	A set pattern in a checklist must be written as *S followed by an optional unsigned integer in the range 1-40 followed by up to 40 characters enclosed in parentheses.	Correct procedure header and resubmit.	CCL
CCL333 - DUPLICATE *K *N - PROC HEADER	A single checklist may not contain more than one *K or more than one *N pattern.	Correct procedure header and resubmit.	CCL
CCL334 - HEADER PARAMETER DESCRIPTION NOT TERMINATED	If a description string follows a parameter name it cannot span more than one line and must be terminated by a matching quotation mark.	Correct procedure header and resubmit.	CCL
CCL336 - PATTERN OR VALUE ON PROC HEADER .GT. 40 CHAR	A single checklist item may contain literal patterns and replacement values (values which stand for themselves). These values may not exceed 40 characters in length. Literals may not exceed 40 characters after evaluation.	Correct procedure header and resubmit.	CCL
CCL352 - HELP PARAMETER NAME NOT FOUND	If a comma follows .HELP, the system expects to find a parameter name or the parameter NOLIST.	Correct directive and resubmit.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL353 - ONLY COMMA ALLOWED AFTER .HELP	The system found something other than comma after .HELP.	Correct directive and resubmit.	CCL
CCL354 - .HELP STATEMENT INCORRECTLY WRITTEN.	The .HELP directive can have one of four formats: .HELP .HELP,,NOLIST .HELP,keyword .HELP,keyword,NOLIST The .HELP directive must not have a terminator.	Correct directive resubmit.	CCL
CCL355 - NO .ENDHELP FOR .HELP	The system found an end of record before it encountered a .ENDHELP directive.	Add an .ENDHELP directive and resubmit.	CCL
CCL357 - DUPLICATE HELP STATEMENTS IN PROC BODY	Either more than one .HELP directive with no parameter name or more than one .HELP directive with the same parameter name was found.	Delete duplicate .HELP directives and retry.	CCL
CFO FROM JOB jsn	The CFO command was issued to the job by the job jsn.	None.	QAC 1RI
CFO FROM OPERATOR	The CFO command was issued to the job by the console operator.	None.	QAC 1RI
CHANNEL MALFUNCTION, lfn AT addr.	A hardware malfunction occurred.	Inform site analyst.	1MT
* CHARGE(charge,project)	Indicates that you entered the default format of the CHARGE command (CHARGE,*). charge Your default charge number. project Your default project number.	None	CHARGE
CHARGE ABORTED.	Dayfile message indicating that a central site operator action caused the CHARGE operation to abnormally terminate.	Reenter CHARGE command.	CHARGE
CHARGE CARD REQUIRED.	A CHARGE command is required before the current USER command may be processed.	Enter a CHARGE command and retry.	ACCFAM
CHARGE FILE BUSY.	Dayfile message indicating that the file which the system uses to validate charge numbers and project numbers is busy.	Reenter CHARGE command.	CHARGE
CHARGE INVALID 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
CHECK DAYFILE FOR ERRORS.	Informative message indicating that you should check the dayfile for errors.	Examine error messages in dayfile.	COPY PFATC PFCAT PFCOPY PFDUMP PFLOAD TCOPY
.CHECKPOINT nnnn COMPLETE.	Indicates that checkpoint nnnn has completed. Issued if only one checkpoint file is present. For a checkpoint operation, more than two checkpoint files or an invalid combination of checkpoint files was specified.	None.	CHKPT
.CHECKPOINT nnnn COMPLETED TO filenam.	Indicates that checkpoint nnnn has been completed to file filenam. Issued if alternate CB checkpoint files are used.	None.	CHKPT
.CHECKPOINT FILE ERROR.	During a checkpoint operation, more than two checkpoint files were present or two checkpoint files of type "CK" were present.	Correct the checkpoint job.	CHKPT RESTART
.CHECKPOINT NOT FOUND.	The specified checkpoint (nn parameter on RESTART command) could not be found on the file.	Verify that checkpoint is on file.	RESTART

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CHKPT UNABLE TO READ TAPE.	Indicates a tape file had an unrecovered parity error when CHKPT tried to write it to the checkpoint file.	None.	CHKPT
CIO ERROR.	Updating of resource file returned error status other than end-of-device.	Inform site analyst.	RESEX
.CKP REQUEST.	A checkpoint has been initiated.	None.	CHKPT
CM BLOCK OUT OF RANGE.	Data transfer from extended memory specified a CM address outside the job field length.	Analyze the job output and dumps to determine the cause.	1AJ
CM NOT VALIDATED.	The number of CM words specified on the job command either exceeds that for which you are authorized or exceeds the machine size available to user jobs.	Rerun job with smaller CM specification or ask site personnel for larger CM validation.	CPM
CM OUT OF RANGE.	The program referenced an address outside the job CM field length.	Analyze job output and dumps to determine the cause.	1AJ
CM PARITY ERROR.	Double data parity error (two data bits failed) between central memory control (CMC) and CM as detected by the single-error correction double-error detection (SECDED) network, or a single parity error when operating in default mode (SECDED network disabled).	Inform customer engineer.	1AJ
CM RANGE EXIT MODE NOT DESELECTABLE.	You attempted to deselect system checking for CM out of range errors. This cannot be done on a model 176.	Determine if the program can run with CM range error checking. If it cannot, the program must be changed.	CPM
CMC PARITY ERROR.	The CPU sent the central memory control (CMC) data or an address having incorrect parity.	Inform customer engineer.	1AJ
CMDFILE ERROR.	The format of the HELP command file is incorrect.	Contact installation personnel.	HELP
COMMAND NOT PROCESSED DUE TO OVERLAP	You have specified that the selected edit command should not be processed due to overlapping line numbers.	None.	IAF
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
COMMAND TOO LONG.	A command longer than 80 characters was entered.	Reenter using a shorter version of the command.	IAFEX
COMMENT FROM JOB jsn	A COMMENT command was issued to the job by the job jsn.	None.	QAC 1RI
COMMENT FROM OPERATOR	A COMMENT command was issued to the job by the console operator.	None.	QAC 1RI
CONFLICTING PARAMETERS.	You queued a file and specified an input disposition more than once.	Reenter with correct parameters.	SUBMIT
CONFLICTING RESOURCE TYPES.	PE, HD, and GE resources cannot be specified concurrently with an NT resource in the same job.	Correct the RESOURC command.	RESEX
CONNECT REJECT, lfn AT addr.	The system was unable to connect a peripheral device.	Inform site analyst.	1MT
CONNECTION PROHIBITED, TRY LATER.	The network is configured so that only one user with a given user name and family name can access the requested application at one time.	Request a different application or log off and try later.	NETVAL
CONNECTION REJECTED.	The requested application is already servicing the maximum number of terminals.	Request a different application or log off and try later.	NETVAL
CONTROL CARD ERROR.	An invalid parameter was specified on the command.	Correct command and retry.	EDIT LISTLB

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CONTROL STATEMENT ARGUMENT ERROR	Internal error.	Notify site analyst.	IAF
CONTROL STATEMENT LIMIT.	You have entered too many commands.	Reduce the number of commands to a value within the specified limit. Refer to LIMITS command.	TCS
CONTROL STATEMENT LIMIT.	The additional 8 commands have been executed.	None.	1AJ
CONTROLLED BACKSPACE ERROR, lfn AT addr.	Controlled backspace operation failed during write error recovery. Position of tape is uncertain.	Inform customer engineer.	1MT
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 command.	Correct error and retry.	CONVERT
CONVERSION NOT SUPPORTED.	The PC and NC parameters on the FCOPY command specify an unsupported conversion.	Refer to the description of the FCOPY command for valid PC and NC values.	FCOPY
COPY COMPLETE.	Copy termination condition was satisfied before EOI was encountered.	None.	COPY COPYBR COPYBF COPYCF COPYCR COPYSBF COPYX TCOPY
COPY FL ABOVE USER LIMIT.	Field length for L or F tape copy exceeds your current maximum.	Use smaller block size on L and F tapes or increase maximum FL.	COPY
COPY INDETERMINATE.	Copy with S, L, or F tape is unpredictable. Only the COPY command supports these formats.	Retry copy using the COPY command.	COPYBF COPYBR COPYEI COPYX
COPYING - type/name	The record with the type and name on the old file was copied to the new file.	None.	COPYL COPYLM
COPYL COMPLETE.	All records on the old file have been processed.	None.	COPYL COPYLM
COPYL DID NOT FIND type/name	The record with type and name on the replacement file was not found on the old file. Since the A option was not selected, the record is ignored.	None.	COPYL COPYLM
CPeq, CHcc Abbbb INCOMPLETE TRANSFER.	An incomplete data transfer occurred and involved the card punch equipment eq. eq EST ordinal of Card punch cc Channel number bbbb Octal byte count not transferred	Inform customer engineer.	QAP 1IO 1CD
CPeq, CHcc CONTROLLER HUNG BUSY.	The specified controller did not drop BUSY status for card punch equipment eq. eq EST ordinal of card punch cc Channel number	Inform customer engineer.	1IO QAP 1CD
CPeq, CHcc Fffff FUNCTION TIMEOUT.	No response (inactive) was received after a function code was issued to the 415 card punch (converter and equipment status unavailable). eq EST ordinal of 415 card punch cc Channel number ffff Function code	Inform customer engineer.	BIO
CPeq, CHcc Fffff FUNCTION TIMEOUT.	No response (inactive) was received after a function code was issued to the specified card punch equipment eq (converter and equipment status unavailable). eq EST ordinal of card punch cc Channel number ffff Function code	Inform customer engineer.	QAP 1CD 1IO
CPeq, CHcc Fffff REJ Paaaa,Cbbbb,Emmmm.	Function reject or transmission parity error was detected on the specified card punch equipment eq. eq EST ordinal of card punch cc Channel number ffff Function code aaaa Driver (1CD) address	Inform customer engineer.	QAP 1CD 1IO

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
	bbbb Converter status mmmm Equipment status		
CPEq, CHcc RESERVED.	The 415 card punch eq is reserved and cannot be connected on channel cc.	Inform customer engineer.	110
CPEq, CHcc TURNED OFF.	The specified 415 card punch equipment eq was logically turned off (OFF status set in EST). This message is preceded in the error log by a message for the same equipment which specifies the failing condition. eq EST ordinal of 405 card punch cc Channel number	Inform customer engineer.	110 QAP
CPM - ARGUMENT ERROR.	An invalid argument was specified.	Retry the command using valid arguments.	CPM
CPM - ILLEGAL PACK NAME.	An invalid pack name has been specified.	Ensure that a legal pack name is used.	CPM
CPM - ILLEGAL REQUEST.	A job was invalid CPM function code, subfunction code was specified, or extended relieve is set (for EREXIT).	Specify auto recall on RA+1 call to CPM.	CPM
CPM - ILLEGAL USER ACCESS.	1. You tried to perform a CPM operation for which you are not authorized. 2. A CPU program issued a DSDOUT or a DSDINP macro request when it did not have access to the L display CMR buffers.	Ensure that the Ldisplay utility was initiated at the system console with the appropriate commands.	CPM
CPM - SYSTEM ERROR.	1. For function 104 (Get Job Start Information), specified family name was not found. 2. For functions 42 (Special Charge) and 106 (Set Job Characteristics), CPM was unable to read the input file.	Inform site analyst.	CPM
CPU ERROR EXIT AT addr.	The errors listed after this message occurred at address addr, causing job termination.	Refer to the descriptions of the error messages issued with this message.	1AJ
CReq, CHcc Abbbb INCOMPLETE TRANSFER. CReq, CHcc CONTROLLER HUNG BUSY. CReq, CHcc Fffff FUNCTION TIMEOUT. CReq, CHcc Fffff REJ Paaaa,Cbbbb,Emmmm. CReq, CHcc TURNED OFF.	Card reader messages. See CPEq...		110 1CD QAP
CReq, CHcc RESERVED.	The 405 card reader is reserved and cannot be connected on channel cc. eq EST ordinal of card reader cc Channel number.	Inform customer engineer.	110
CRC ERROR.	An error was detected in cyclic redundancy character. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
CUMULATIVE LIMIT EXCEEDED.	Dayfile and output file message indicating that one of the installatio*-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.	The system has detected an error in its validation file.	Contact installation personnel.	CHARGE
DATA BASE ERROR n - NOTIFY ANALYST.	System error dayfile message indicating that an abnormal situation exists. n is displayed for consideration by the analyst. The internal documentation, obtained by using the DOCUMENT command, contains an explanation of each error n for use by the analyst. (Refer to section 9 for a description of DOCUMENT.)	Inform site analyst.	PROFILE

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
DATA/PERMIT ERRORS.	When loading a file from tape, errors were encountered in both data and permit information.	Enter CHANGE statement or macro with CE parameter to allow access to the file. Make the file local and check if data is accurate. Do a CATLIST to see if the permits are accurate.	PFM
DATA TRANSFER ERROR.	An error occurred in a read operation during a file transfer.	Inform site analyst.	PFM
DAYFILE - ARGUMENT ERROR.	Keyword specified is not recognizable or command is not properly formatted.	Check keyword and command 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 job 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
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 nonnumeric.	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 nonnumeric. Print density must be 3,4,6, or 8.	Retry with a valid option.	DAYFILE
DAYFILE - INVALID JOB TERMINATION OPTION.	An invalid value was specified for the *JT* option.	Correct option and retry.	DAYFILE
DAYFILE - JOB FIELD DISALLOWED ON USER DAYFILE.	Options specified are not allowed for your job's dayfile processing.	Retry with a correct OP=op parameter.	DAYFILE
DAYFILE - LOCAL DAYFILE PROCESSED.	DAYFILE has successfully completed processing a local file as input.	None.	DAYFILE
DAYFILE - RESERVED FILE NAME.	The file name specified for the L=fn or I=infile parameter of the DAYFILE command is a reserved name.	Retry using a nonreserved name.	DAYFILE
DAYFILE - TTY INPUT FILE NOT ALLOWED.	Dayfile data cannot be entered directly from the keyboard.	Specify dayfile input from other than the terminal (TT device).	DAYFILE
DAYFILE - UNEXPECTED EOF/EOI ENCOUNTERED.	An EOF or EOI was encountered before the PRU count was depleted on the input file. The dayfile is shorter than expected based on the PRU count.	Retry operation.	DAYFILE
DAYFILE - UNKNOWN *OP* FIELD.	The option specified is not valid.	Retry using a valid option.	DAYFILE
DC REQUIRED IF JSN/UJN NOT SPECIFIED.	If the JSN or the UJN parameters are omitted, the DC parameter must be entered.	Add the DC parameter and retry.	DROP
DExx,Ccc,l,sec,ann,Stttt,Aaddr.	An error has been detected on extended memory. Refer to appendix B of the NOS 2 Operator/Analyst Handbook for further information.	Inform site analyst.	6DE
-DEBUG-GARBAGE IN ZZZZDS FILE.	File ZZZZDS contains information other than the suspended job of the current session. Retrieval of debug information from session is impossible.	Return file ZZZZDS and restart debug session.	DEBUG
DEBUG-ILLEGAL PARAMETER-p.	You specified an invalid parameter on DEBUG command. The three parameters allowed are ON, OFF, or RESUME.	Reenter DEBUG command with correct parameter.	DEBUG

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
DEBUG-ONLY ONE PARAMETER ALLOWED.	You entered more than one parameter on the DEBUG command.	Reenter DEBUG command with correct parameter.	DEBUG
-DEBUG-ZZZZDS FILE NOT FOUND.	You 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
DEFAULT CHARGE ABORTED	A system error aborted default charge processing.	Inform site analyst.	CHARGE
DEFAULT CHARGE NULL.	An informative message. It indicates that the default charge information was not specified in the accounting block.	No action required.	CHARGE
DEFERRED BATCH VALIDATION EXCEEDED - lfn.	You have too many queued files.	Enter the LIMITS command to ascertain the number of queued files you can have.	DSP
DEL	The system has deleted the last line of input.	None.	CCP
DEMAND EXCEEDED.	You attempted to assign more units than were scheduled on the RESOURC command.	Increase appropriate parameter value on RESOURC command.	RESEX
DEMAND FILE ERROR.	Resource execution error was encountered. This error occurred because the demand file (RSXdid) entry does not match the job identification.	Inform site analyst.	RESEX
DEMAND VALIDATION ERROR.	The specified number of units exceeds your validation limits.	Decrease appropriate parameter value on RESOURC command.	RESEX
DETACH LIMIT EXCEEDED.	You have too many detached jobs or are not authorized to detach jobs.	If you are authorized to detach jobs, enter the ENQUIRE,JSN command to identify existing detached jobs and the DROP,jsn to drop selected jobs.	IAFEX
DEVICE ERROR ON FILE lfn AT addr.	An irrecoverable error occurred on the mass storage device containing the file lfn.	Inform site analyst.	CIO
DEVICE UNAVAILABLE.	Access to the permanent file device requested is not possible. You may have attempted to access files on a device not present in the alternate system.	Ask the operator to make the device available.	PFM
Dlxx,Ccc,l,sec,ann,Stttt,Fnqqqq. or Dlxx,Ccc,l,sec,ann,Stttt,Uuu Cyyyy Sttss.	An error has been detected on mass storage device with EST ordinal xx. Refer to appendix B of the NOS 2 Operator-Analyst Handbook for further information.	Inform site analyst.	7DI
Dlxx,Ccc,l,sec,ann,Stttt,Fnqqqq. or Dlxx,Ccc,l,sec,ann,Stttt,Uuu Cyyyy Sttss.	An error has been detected on mass storage device with EST ordinal xx. Refer to appendix B of the NOS 2 Operator-Analyst Handbook for further information.	Inform site analyst.	7DI
DIRECT ACCESS DEVICE ERROR.	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 you are on an alternate system, your master device may not have been transferred to that system.	Specify correct device type.	PFM
DIRECTIVE ERRORS.	Dayfile message indicating that one or more input directives were in error. Fatal error.	Examine output file to determine reason for error.	MODIFY OPLDIT LIBTASK MODVAL PROFILE SYSEDIT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
DIRECTIVE ERRORS.	A LIBEDIT directive has incorrect syntax.	Examine the LIBEDIT output to determine cause of error.	LIBEDIT
n DIRECTIVE ERRORS.	The System could not interpret n directives.	Correct errors as listed in LIBEDIT output and rerun job.	LIBEDIT
DIRECTORY OVERFLOW.	The format of the HELP command file is incorrect.	Contact installation personnel.	HELP
DISK FULL.	Job was terminated without exit processing.	Resubmit job.	1AJ
DM* TOO SHORT.	System file DM* cannot accomodate the dump.	Inform site analyst.	CPMEM
DPxx,Ccc,l,sec,ann,Stttt,Fnqqq. or DPxx,Ccc,l,sec,ann,Stttt,Aaddr,Wwww DPxx,Ccc,l,Gggg...g. DPxx,Ccc,l,Bbbb...b. or DPxx,Ccc,l,sec,ann,Stttt,Aaddr,Wwww DPxx,Ccc,l,ddd...d.	An error has been detected on distributive data path (DDP). Refer to appendix B of the NOS 2 Operator/Analyst Handbook for further information.	Inform site analyst.	6DP
DROP FROM JOB jsn	The job was dropped by the job jsn.	None.	QAC 1RI
DROP FROM OPERATOR	The job was dropped by the operator.	None.	QAC 1RI
DROPPING SUPPORT OF SUMMARY, USE ENQUIRE.	SUMMARY will be removed in a future release.	Use ENQUIRE instead of SUMMARY.	ENQUIRE
DSP - CAN NOT ROUTE JOB INPUT FILE.	An attempt was made to route the job input file.	Copy job input file to a local file and route it.	DSP
DSP - COMPLETE BIT ALREADY SET.	The complete bit was not cleared before DSP was called.	Clear complete bit before calling DSP.	DSP
DSP - DEVICE FULL.	There is no space on the device for current use.	Retry route at a later time.	DSP
DSP - DEVICE UNAVAILABLE.	DSP attempted to create a file on a device that was turned off or is currently unavailable for access.	Specify different device or contact site operator.	DSP
DSP - FILE NAME ERROR.	An attempt was made to create a file with an invalid file name.	Specify valid file name.	DSP
DSP - FILE NOT ON MASS STORAGE.	An attempt was made to route a file not on mass storage.	Copy file to mass storage before routing.	DSP
DSP - FILE ON REMOVABLE DEVICE.	A file on a removable device cannot be routed.	Copy file to nonremovable device before routing.	DSP
DSP - FORMS CODE NOT ALPHANUMERIC.	Forms code must consist of two alphanumeric characters.	Specify alphanumeric forms code.	DSP
DSP - I/O SEQUENCE ERROR.	A request was made on a busy file.	Wait until file is not busy.	DSP
DSP - ILLEGAL REQUEST.	One of the following. - DSP was not called with recall (does not apply when the call is made by a subsystem). - Parameter list address was out of range. - RA+1 call was formatted incorrectly.	Specify auto recall with DSP call or determine why parameter list address is out of range.	DSP
DSP - ILLEGAL USER CARD.	You attempted to queue a file with an input disposition and the file has an invalid USER command.	Ensure that valid USER command is being used.	DSP
DSP - IMMEDIATE ROUTING - NO FILE.	The specified file for the immediate routing could not be found.	Ensure that file to be routed is available to job for processing.	DSP
DSP - INVALID DISPOSITION CODE.	Specified disposition code is not recognized.	Verify disposition code.	DSP
DSP - INVALID EXTERNAL CHARACTERISTICS.	You specified an undefined external characteristic code.	Verify external characteristic code.	DSP
DSP - INVALID FILE MODE.	You attempted to route an execute-only file.	Verify that the file to be routed is not an execute-only file.	DSP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
DSP - INVALID FILE TYPE.	The file being processed is not a local or queued file type.	Ensure that file being processed is of correct type.	DSP
DSP - INVALID INTERNAL CHARACTERISTICS.	You specified an undefined internal characteristics code.	Verify internal characteristics code.	DSP
DSP - INVALID ORIGIN TYPE.	The System cannot queue the file for input with the specified origin type.	Specify valid origin type.	DSP
DSP - INVALID TID.	One of the following. - User name and family name parameters were not in CM field length. - Terminal identifier (TID) is greater than or equal to IDLM for batch jobs. - The family name or user name is not valid.	Verify that TID parameters are valid.	DSP
DSP - LOCAL FILE LIMIT.	Your local file validation limit has been exceeded.	Return one or more local files to the system.	DSP
DSP - OUTPUT FILE LIMIT.	Your output file validation limit has been exceeded.	If possible, split job into two or more jobs and retry. Otherwise, reduce number of files by copying output to single file and then routing the file.	DSP
DSP - ROUTE TO INPUT NOT IMMEDIATE.	You attempted to queue a file for input using a deferred ROUTE.	Change to immediate route.	DSP
DSP - THIS ROUTING NOT ALLOWED.	You attempted to change the queue disposition of a deferred routed file.	Rescind prior routing by using the SC disposition code with the ROUTE command or macro. ROUTE the file again with the desired final disposition code.	DSP
DSP - TOO MANY DEFERRED BATCH JOBS.	You attempted to submit more deferred batch jobs than allowed by your validation limit.	Wait for jobs to complete or request a larger validation limit from site personnel.	DSP
**** DUPLICATE CHARGE NUMBER.	An existing charge number was referenced on a create run.	Rerun using correct charge number, if required.	PROFILE
DUPLICATE ENTRY POINT FOUND - xxx.	The input file contains records with identical entry point names. The name xxx specifies the duplicate entry point.	Remove one of records that appear in duplicate.	LIBGEN
DUPLICATE FILE NAME.	The file specified already exists in the system.	Use different name in request.	LFM
DUPLICATE FILE NAME	The read or destination file name matches the primary file name.	Enter the edit command with the read or destination file name different than the primary file name.	IAF
**** DUPLICATE PROJECT NUMBER.	An existing project number was referenced on a create run.	Rerun using correct project number, if required.	PROFILE
DUPLICATE UJN FOUND.	Two of your jobs have the same UJN.	Use the ENQUIRE command to determine the appropriate JSM to use.	QGET
DUPLICATE USER NUMBER.	Output file message indicating that the user name encountered on a create run is a duplicate of a user name previously entered. The first entry is used.	Rerun the corrected job or correct the new validation file, if necessary.	MODVAL
DUPLICATED LINES.	Duplicate lines, which would have been printed during a dump operation, were suppressed.	None.	CPMEM
EC NOT VALIDATED.	The number of extended memory blocks specified on the Job command either exceeds that for which you are authorized or exceeds the machine size available to user jobs.	Check your validation with the LIMITS command.	CPM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ECS BLOCK OUT OF RANGE.	Data transfer between CM and extended memory specified an extended memory address outside the job field length.	Analyze the job output and dumps to determine the cause of the error.	1AJ
ECS FLAG REGISTER PARITY.	Parity error detected on extended memory flag register operation.	Inform customer engineer.	1AJ
ECS OUT OF RANGE.	Job referenced extended memory address outside job field length.	Analyze the job output and dumps to determine the cause of the error.	1AJ
ECS READ ERROR.	An unrecoverable extended memory read error occurred.	None.	CPHEM
ECS ROLLIN ERROR.	An irrecoverable extended memory parity error occurred while the system was writing the job's extended memory field length to a rollout file.	Inform the site analyst of the parity error. Rerun the job after extended memory has been repaired.	1RI
EDITING COMPLETE.	Library editing is complete.	None.	LIBEDIT
lfn EMPTY.	The file specified on a SAVE command contains no data.	Verify that file contains data and retry.	PFM
EMPTY CATALOG.	No entries exist for options selected.	None.	CATLIST NDA
EMPTY FILE (SSF)	Internal error.	Notify site analyst.	IAF
EMPTY LINES.	Blank lines were encountered.	Delete or replace the empty lines.	NOTE
EMPTY SORT INPUT FILE.	Dayfile message indicating that the file specified on the SORT command contains no data.	Correct and rerun.	SORT
END eq, nn.	The operator ended batch equipment eq for nn copies.	None.	1CD
END OF SET ILLEGAL REQUEST, lfn AT addr.	A multifile set tape is positioned at the end of set after searching for a file set that was not found. Until the tape is returned or repositioned within the multifile set, all other tape operations on this tape are invalid.	Return the tape via a RETURN, UNLOAD, or EVICT command, or reposition the tape with the LABEL command.	1MT
END OF TAPE, lfn AT addr.	The end-of-tape was encountered.	Ensure that correct file manipulation operation is specified.	1MT
**** ENDING SUPPORT OF STS FUNCTION 01 ***.	The system will phase out support of this function.	Consult the site analyst to find a different method to obtain the required data.	SFP
ENQUIRE ABORTED.	ENQUIRE has been aborted as a result of a system error.	Inform site analyst.	ENQUIRE
ENQUIRY COMPLETE.	Message issued when ENQUIRE command processing is completed.	None.	ENQUIRE
ENTER GO TO CONTINUE CURRENT JOB. RELIST TO LIST RECOVERABLE JOBS, OR DESIRED JSN:	You have recoverable jobs and the system awaits your input.	Enter appropriate response.	RECOVER
ENTRY POINT NOT FOUND.	The specified entry point could not be found.	Verify that entry point is valid.	1AJ
EOF ENCOUNTERED.	End-of-file was encountered before copy termination condition was satisfied.	None.	COPIX
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 CHANGED BY RECOVERY.	The file was truncated during deadstart recovery.	Use CHANGE command or macro with CE parameter to allow access to the file. Make the file local and list it to determine how much of the file was lost.	PFM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
EOI ENCOUNTERED.	End-of-information was encountered on the input file.	If the message was not expected, check if the copy was performed as desired. The source file may not have been positioned correctly before the copy.	COPY COPYBF COPYBR COPYCF COPYCR COPYEI COPYSBF COPYX TCOPY
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
EQxx, DNdn, BAD CATALOG/PERMIT SECTOR.	PFM has encountered a catalog or permit sector which does not have a valid sector length. EQ Equipment type as defined in EQxx, DNdn, DIRECT ACCESS FILE ERROR message description. xx EST ordinal of device. dn Device number.	Inform site analyst.	PFM
EQxx, DNdn, DIRECT ACCESS FILE ERROR.	The system sector data for the file does not match the catalog data (error log and dayfile message). EQ One of the following equipment types. DI 844-21 disk (half track) DJ 844-41/44 disk (half track) DK 844-21 disk (full track) DL 844-41/44 disk (full track) DM 885 disk (half track) DQ 885 disk (full track) DV 819 disk (single density) DW 819 disk (double density) xx EST ordinal of device dn Device number	Inform site analyst.	PFM
EQxx, DNdn, FILE LENGTH ERROR.	The length of a file does not equal the length specified in the catalog. EQ Equipment type as defined in EQxx, DNdn, DIRECT ACCESS FILE ERROR message xx EST ordinal of device dn Device number The cause depends on the type of command or macro issued. GET A local file is created with length being the actual length retrieved. SAVE If file length is longer than TRT specification, file is truncated. REPLACE Same as for SAVE.	Inform site analyst.	PFM
EQxx, DNdn, MASS STORAGE ERROR.	An error was encountered in reading a portion of the permanent file catalog or permit information (error log and dayfile message). EQ Equipment type as defined in the EQxx, DNdn, DIRECT ACCESS FILE ERROR message xx EST ordinal of device dn Device number	Inform site analyst.	PFM
EQxx, DNdn, RANDOM INDEX ERROR.	The random disk address of the permit sector is in error (error log and dayfile message). EQ Equipment type as defined in the EQxx, DNdn, DIRECT ACCESS FILE ERROR message xx EST ordinal of device dn Device number	Inform site analyst.	PFM
EQxx, DNdn, REPLACE ERROR.	The same file was found twice during a catalog search. This error can occur for APPEND or REPLACE commands or macros after a file is found and purged and the catalog search is continued (error log and dayfile message). EQ Equipment type as defined in the EQxx, DNdn, DIRECT ACCESS FILE ERROR message xx EST ordinal of device dn Device number	Retry job step.	PFM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
EQxx,DNdn, SYSTEM SECTOR ERROR.	The system sector of an indirect access permanent file contains an error (error log and dayfile message). EQ Equipment type as defined in EQxx, DNdn, DIRECT ACCESS FILE ERROR message description xx EST ordinal of device dn Device number	Inform site analyst.	PFM
EQxx,DNdn, TRACK LIMIT.	No allocatable tracks remain on equipment xx (error log and dayfile message). EQ Equipment type as defined in the EQxx,DNdn, DIRECT ACCESS FILE ERROR message xx EST ordinal of device dn Device number	Inform site analyst.	PFM
EQxx,FM=family,PF=filenam,UI=userin.	Additional line is written only in error log after one of the following messages. - DATA TRANSFER ERROR. - EQxx,DNdn,DIRECT ACCESS FILE ERROR, AT addr. - EQxx,DNdn,FILE LENGTH ERROR - FILE BOI/EOI/UI MISMATCH. - EQxx,DNdn,MASS STORAGE ERROR - EQxx,DNdn,RANDOM INDEX ERROR - EQxx,DNdn,REPLACE ERROR - EQxx,DNdn,SYSTEM SECTOR ERROR - EQxx,DNdn,TRACK LIMIT EQ Equipment type as defined in the EQxx,DNdn, DIRECT ACCESS FILE ERROR message. xx EST ordinal of device family Family name filenam Permanent file name userin User index	None.	PFM
EQUIPMENT NOT AVAILABLE.	Tape assignment error was encountered; requested equipment is either in use or is not defined in the system.	Ensure accuracy of macro or command and/or retry at a later time.	LFM RESEX
ERASE LIMIT, lfn AT addr.	The system made 20 erasures (10 feet of tape) without being able to successfully write the tape.	Clean tape or use different tape.	1MT
ERR..	While entering a terminal definition or interactive status command, you made a format error or entered a command which is invalid for your terminal type or not supported at your installation.	Correct format. If format is correct, check validity of command for this terminal type.	CCP
ERROR CODE ec, lfn AT addr.	1MT error code ec has occurred but no specific message is issued. This would normally not occur unless the job was dropped by the operator.	Inform site analyst.	1MT
.ERROR - FILE(S) NOT PROCESSED.	One or more files were not checkpointed because CHKPT detected address errors.	Determine and correct address errors and retry.	CHKPT
ERROR FLAG TERMINATION, lfn AT addr.	The job was aborted with a tape operation in progress. The operation/request is not complete. For example, the operator could kill the job while tape error recovery is in progress.	None.	1MT
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
ERROR IN ARGUMENT.	One of the following. - The pfn is blank or fn = lfn. - You specified no arguments or a blank argument. - You specified too many files. - You entered an invalid parameter.	Reenter the command with correct parameters.	PFILES
ERROR IN ARGUMENTS.	The system did not recognize one or more parameters.	Compare the parameters specified with the command description.	MFILES
ERROR IN ARGUMENTS.	One or more of the following command errors were detected. - More than one date was entered. - No options were selected. - The parameter was invalid or could not be recognized. - The TM option was selected but no data was specified.	Correct control statement and retry.	BLANK PURGALL ITEMIZE

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
	- Both the device number parameter and the pack name or auxiliary device parameter were selected; auxiliary devices do not have device numbers.		
ERROR IN ARGUMENTS.	RESTART detected invalid arguments on the command call.	Correct the command.	RESTART
ERROR IN ARGUMENTS.	The parameter list specified on the JOB command and/or the USER command was not correct.	Correct the parameter list and retry.	ACCFAM
ERROR IN BACKUP REQUIREMENT.	The option specified on the BR parameter on the DEFINE, CHANGE, or SAVE command is incorrect.	Check the BR parameter description and correct the command.	PFILES
ERROR IN CONTROL ARGUMENTS.	Either the number of arguments was invalid, an invalid argument was specified, or a numeric parameter was nonnumeric.	Check the description of the command.	CONTROL
ERROR IN DATE.	The format of the date (ad, md, or cd) parameter in a PURGALL request was incorrect.	Check the format of the PURGALL command in section 10.	PURGALL
ERROR IN DEVICE NUMBER.	The file residency as specified by the device number parameter was illegal.	Correct the device number parameter.	PURGALL
ERROR IN FCOPY ARGUMENTS.	The format of the FCOPY command is in error.	Refer to the description of the FCOPY command.	FCOPY
ERROR IN FILE CATEGORY.	The option specified on the CT parameter is incorrect.	Check the CT parameter description and correct the command.	PFILES PURGALL
ERROR IN FILE DATA.	When loading a file from tape, an error was encountered in the data.	Enter CHANGE command or macro with CE parameter to allow access to the file. Make the file local and ensure the accuracy of the data.	PFM
ERROR IN FILE TYPE.	You specified an invalid file type.	Check the PURGALL statement description and correct the statement.	PURGALL
ERROR IN KRONREF ARGUMENTS.	The command entered contained an invalid parameter.	Correct and rerun.	KRONREF
ERROR IN LIBRARY ARGUMENTS.	The LIBRARY command has an invalid argument or the format of the command is incorrect.	Correct the error and reenter the command.	LIBRARY
ERROR IN LIMITS ARGUMENT.	Dayfile message indicating that parameters other than *L=* were included on the LIMITS command.	Correct and rerun.	MODVAL
ERROR IN MODE.	The option specified on the M parameter is incorrect.	Check the M parameter description and correct the command.	PFILES
ERROR IN PASSWOR ARGUMENTS.	PASSWOR command parameters are incorrect.	Correct control statement or command and reenter.	PASSWOR MODVAL
ERROR IN PERMIT DATA.	When loading a file from tape, an error was encountered in the permit entries.	Enter CHANGE command or macro with CE parameter to allow access to the file. Do a CATLIST to see if the permits are accurate.	PFM
ERROR IN PREFERRED RESIDENCE.	The option specified on the PR parameter on the DEFINE or CHANGE command is incorrect.	Check the PR parameter description and correct the command.	PFILES
ERROR IN PROFILE ARGUMENTS.	Dayfile message indicating there was an error on the PROFILE command.	Correct the command, and rerun.	PROFILE
ERROR IN ROUTE FUNCTION, LFN=filenam.	Informative message issued to the system dayfile indicating that an error occurred while routing filenam.	Examine the job's dayfile for a more specific error message.	DSP
ERROR IN SUBSYSTEM.	The option specified on the SS parameter on the CHANGE or SAVE command is incorrect.	Check the SS parameter description and correct the command.	PFILES

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ERROR IN SYSTEM TEXT.	The system text referenced by the KRONREF command did not contain symbol definitions.	Ensure that the system text specified by the G or S parameter contains symbol definitions.	KRONREF
ERROR IN TIME.	The format of the time parameter in a PURGALL request was incorrect.	Check the format of the PURGALL command in section 10.	PURGALL
ERROR IN TRMDEF ARGUMENT.	One or more parameters specified in the TRMDEF command were in error.	Ensure that parameters are in correct format and new values are valid.	TRMDEF
ERROR IN 5TH COPYL PARAMETER	Fatal error; only A, R, E, and T are recognized parameters.	Correct processing option.	COPYL COPYLM
ERROR LIMIT EXCEEDED.	Number of parity and block-too-large errors exceed the error limit.	If dayfile shows block too-large errors have occurred and tape is S, L, or F format, increase block size and retry; otherwise, tape is probably assigned in the wrong format. If parity errors have occurred, the tape is bad and the data on it cannot be correctly recovered.	COPY TCOPY
ERROR ON DETACH.	The system was not able to detach the job upon your request.	Retry command.	IAFEX
EXCHANGE PACKAGE/MEMORY DUMP ON FILE ZZZDUMP.	The exchange package and memory dump is written on local file ZZZDUMP because the job is of interactive origin and file OUTPUT is assigned to a terminal.	To examine the exchange package and dump, list file ZZZDUMP.	CPMEH
EXECUTE	The system is processing one of your commands.	None.	IAFEX
EXECUTE ONLY FILE.	The permanent file accessed can only be executed and cannot be read, written, or listed.	Ensure accuracy of entry.	IAFEX
FCOPY COMPLETE.	The FCOPY conversion is finished.	None.	FCOPY
FILE BOI/EOI/UI MISMATCH.	Data in the system sector for file lfn does not match information from the EOI sector and/or catalog information.	Inform site analyst.	PFM
FILE EMPTY.	The file specified was empty.	Verify that the file contains data and retry.	LFM SFM
.FILE ERROR filename.	An invalid address was detected on file filename.	Correct error and retry.	RESTART
FILE IDENTIFIER NOT FOUND.	You failed to specify or improperly specified the FI=fileid or QN=seqno parameters of the LABEL command.	Correct the FI=fileid and QN=seqno parameter specifications of the LABEL command.	1MT
FILE/JOB NOT FOUND.	The specified file or job was not found in the system.	Use the ENQUIRE command to ensure job is still in the system.	CONTROL
FILE NAME CONFLICT.	The same file name has been specified for more than one parameter.	Correct the command so that all files have a unique name.	LIBEDIT
.FILE NAME CONFLICT.	For the L072 command, the S and L file names are the same. For the ITEMIZE command, the output and binary file names are the same.	Rename either output file or file to be itemized.	L072 ITEMIZE
FILE NAME CONFLICT.	Either you tried to process two files having the same name or you specified a reserved file name.	Correct the command so all files have a unique name.	CONVERT FCOPY TCOPY COPY VERIFY
FILE NAME CONFLICT.	Dayfile message indicating that the file name specified with the P option (terminal network description) is the same as the file specified with the L option (error listing).	Specify unique file names with the P and L options.	ITEMIZE COPYL COPYLM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
FILE NAME ERROR	The file name specified contains more than seven characters or non-alphanumeric characters.	Reenter command with correct file name.	IAF
FILE NAME ERROR.	File name contains invalid characters or contains more than seven characters.	Ensure that a valid file name is specified.	LFM LISTLB 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 NAME TOO LONG.	You specified a file name that has more than seven characters.	Use a file name that has seven or less characters.	COPYL COPYLM
FILE NOT FOUND.	Requested file could not be found.	Verify that file exists and retry.	LFM SFM QFM ENQUIRE STIMULA
FILE NOT FOUND - filename.	The designated input file did not exist prior to the copy or verify.	If file should have existed, reaccess the file and retry copy.	COPY COPYBR COPYBF COPYEI COPYX VERIFY
FILE NOT ON MASS STORAGE.	The specified file does not reside on mass storage.	Copy file to mass storage and retry.	1AJ
FILE NOT OVERLAY FORMAT.	First record on the file was not an overlay.	Correct file format on LDR call.	LDR
FILE STRUCTURES NOT COMPATIBLE.	Verification results cannot be guaranteed correct when the logical structures of the files being verified are not compatible.	None.	VERIFY
FILE TOO LONG.	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 you are authorized.	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 you are authorized.	Reduce the length of the file or ask site personnel for a larger direct access file length limit.	CIO
FILENAME CONFLICT.	The first two parameters of the GTR command are identical.	Specify different file names for the first two parameters on the GTR command.	GTR
nnn FILE(S) PROCESSED.	The operation was performed on nnn files.	None.	MFILES
FILL STATUS ILLEGAL.	The system has detected an odd number of frame, a condition which is invalid for the data format of the tape being read. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
FL BEYOND MFL.	Request field length exceeds maximum field length.	Increase MFL via MFL request or increase your validation FL.	1AJ
FL REQUEST BEYOND MFL CM.	CM field length requirements for the job step exceed the CM field length allowed.	Increase job step CM field length or contact the data base administrator.	1MA TCS MSABT
FL REQUEST BEYOND MFL (ECS).	Extended memory field length requirements for the job step exceed the extended memory field length allowed.	Increase job step extended memory field length.	1MA
FL TOO SHORT FOR PROGRAM.	The job's field length is too short for the program.	Rerun the job with larger field length specification.	TCS 1AJ
FILE TOO SHORT FOR LOAD.	You attempted to load extended memory data beyond the job's extended memory field length.	None.	1AJ LDR

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
FM NOT LEGAL FAMILY.	Dayfile message indicating that an invalid family name was specified with the FM parameter.	Correct the FM parameter and rerun.	PROFILE MODVAL
**** FORMAT ERROR.	Your entry is not recognized as valid system input. The line of input is disregarded.	None.	MODVAL
FORMAT ERROR.	One of the following. - The GTR command format was incorrect. - An invalid library type was specified. - A record name longer than seven characters was specified.	Refer to the description of the GTR command.	GTR
FORMAT ERROR ON CONTROL CARD.	An error was detected in the format of the command.	Check the description of command format.	TCS
FORMAT REQUIRES UNLABELED TAPE.	The format specified (F) is valid only for unlabeled tapes.	The tape must be assigned as an unlabeled tape.	RESEX
FOT FULL - FILES IGNORED.	The queued files were not processed because there was no FOT entry for the creation and/or destination family and the table was full.	A level 0 deadstart is required to create space in the FOT.	QREC QDUMP QLOAD QMOVE
pfn FOUND. or un FOUND.	The file name specified on the CATLIST (LO=O) command was found in your catalog or the file name or user name specified on the CATLIST (LO=P) command was granted permission.	None.	PFM
FOUND SECTION yyyy.	This is the last line of a three-line message: MULTIFILE NOT FOUND, lfn AT addr. REQUESTED SECTION xxxx. FOUND SECTION yyyy. The system has reached the end of a multifile set. Ignore lfn, addr, xxxx, and yyyy.	None.	1MT
FR NOT FOUND.	The specified string was not in the dayfile. If your job is interactive you will get a full dayfile dump. A dayfile dump is not produced for a batch job.	Retry with a corrected FR=string.	DAYFILE
FROM LOP...message	The local operator sent this message.	If necessary, you can respond using the CMS command.	CS
FUNCTION REJECT, lfn AT addr.	Function was rejected (possible hardware problem).	Inform site analyst.	1MT
FWA .GE. LWA+1.	The first word address parameter was greater than the last word address parameter on DMP, DMD, DMPECS, DMDECS, LOC, or PBC command or DMP, DMD, DED, or DEP system requests.	Correct error and retry.	CPMEM
FWA/LWA .GE. FL.	Either first word address parameter of LOC or the last word address of LOC or PBC was greater than or equal to your field length.	Reduce FWA and/or LWA and retry.	CPMEM
GLOBAL LIBRARY DIRECTORY DEFECTIVE.	The global library directory file ZZZZLD contains errors.	Regenerate the library directory file.	1AJ
GLOBAL LIBRARY DIRECTORY NOT FOUND.	Global library set processing is active, but there is no directory file (ZZZZLD).	Enter the LIBRARY command with no parameters to clear the global library set processing, or regenerate ZZZZLD by reentering the LIBRARY command with the appropriate libraries specified.	1AJ
GLOBAL LIBRARY SET CLEARED.	An informative message.	None.	LIBRARY
GO FROM JOB jsn	A GO command was issued to the job by the job jsn.	None.	QAC 1RI
GO FROM OPERATOR	A GO command was issued to the job by the console operator.	None.	QAC 1RI

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
GTR ERRORS.	There are errors on the GTR command..	Check the description of the GTR command.	GTR
.H VALUE INVALID.	The H parameter was zero or greater than the buffer length.	None.	L072
HELLO/LOGIN DO NOT RESET TERMINAL CHARACTERISTICS.	Informative message resulting from HELLO or LOGIN command. It indicates terminal characteristics of previous session are still in effect, rather than the default login terminal characteristics.	None.	NETVAL
HELP AVAILABLE ONLY TO TIMESHARING JOB.	You can only use HELP from an interactive job.	Log into an interactive facility.	HELP
HELP NOT AVAILABLE.	HELP file is not in library.	Contact installation personnel.	HELP
HTIME xxxxxxxxxxxxxx.kCYCLES.	Dayfile message giving the model 176 CPU clock cycle count for the job. The count is in kilocycle units.	None.	1AJ
HTIME NOT AVAILABLE.	The HTIME command or macro is valid only on a model 176.	None.	1AJ
I/O ON EXECUTE-ONLY FILE.	An overlay load was attempted from an execute-only file other than that from which the main routine was loaded.	Correct overlay calls and retry.	LDR
I/O ON EXECUTE-ONLY FILE lfn AT addr.	You attempted an action not allowed on an execute-only file.	Assign file in proper mode to allow the desired operation.	CIO
I/O SEQUENCE ERROR.	Action was requested on a busy file.	Wait until file is not busy and retry.	LFM PFM 1AJ
I/O SEQUENCE ERROR ON FILE lfn AT addr.	You attempted to perform a function on a local file before another function on the same file was complete.	Modify program to wait until each function is complete before attempting another.	CIO
IAF CLOSED.	IAF is not available for use by terminals.	None.	IAFEX
IDLE	The system is waiting for you to enter a command.	Enter a command.	IAFEX
ILLEGAL APPLICATION, TRY AGAIN.	The requested application is unknown or unavailable to you.	Ensure the accuracy of the entry. If problems persist, contact installation personnel concerning validations.	NETVAL
ILLEGAL CHARACTER.	Invalid character read from 9-track tape. If a 1 is detected in bit 6 of a translated character, the character is invalid. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry of inform site analyst.	1NT
ILLEGAL CHARACTER NUMBER.	In a copy request, one of the following was detected. - Last character position was less than first character position. - Last character position was greater than 150. - Either first character position or last character position was unrecognizable.	Ensure accuracy of command parameters.	COPYCF COPYCR
ILLEGAL CODE, FILE xxxx, REC yyyy.	In file xxxx, record yyyy of lfn1 (as named in the FCOPY command) a code exists which does not have a corresponding code in the character code set of lfn2. FCOPY converts the code to a space and continues converting lfn1.	List lfn1 using the TDUMP command to determine the code converted to a space. Refer to appendix A for listings of the character code sets.	FCOPY
ILLEGAL COMBINATION OF QN/SI.	SI was specified and QN was not. If QN is specified, then SI must also be specified.	Specify SI and rerun the job.	LISTLB
ILLEGAL COMMAND.	You entered an invalid command or a command you are not authorized to use.	Correct command and reenter.	IAFEX
ILLEGAL CONTROL CARD.	The system could not identify the command.	Check validity of command.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ILLEGAL CONTROL CARD.	One of the following has occurred. <ul style="list-style-type: none"> - The command could not be identified. - The USER command does not have a user name specified. - An invalid parameter was specified or no terminator was detected. - You included too many parameters on the program call command or the program was not present. - You submitted a command that you were not authorized to use (for example, the unauthorized use of PASSWOR). - The user submitted a command that is illegal for a particular job type or file type (for example, the use of a FAMILY command in a nonsystem origin job). 	Ensure accuracy and suitability of command.	TCS CONFIG MODVAL RESEX PASSWOR 026
ILLEGAL COPY.	File and/or conversion types do not meet copy requirements.	Correct error and retry.	COPY TCOPY
ILLEGAL COUNT.	The copy file count is nonnumeric or specified as zero.	Correct command and retry.	COPYCF COPYCR COPYSBF
ILLEGAL DEVICE REQUEST.	The device type (R parameter) specified on a request for an auxiliary device cannot be recognized or does not exist in the system. If the auxiliary device specified by the PN parameter is not the same type as the system or job default, the R parameter must be included; if not, the message is issued.	Examine auxiliary device request and ensure its accuracy.	PFM
ILLEGAL DEVICE TYPE - lfn.	File lfn was specified as either the old file or the correction file and is not a mass storage file.	Choose an old file and a correction file that reside on mass storage.	LIBEDIT
ILLEGAL EQUIPMENT.	Equipment specified does not exist or is not allowed (for example, an interactive terminal is requested from other than interactive origin, or a tape is being requested with the REQUEST macro).	Ensure that file resides on a legal equipment type.	LFM RESEX
ILLEGAL EQUIPMENT ON FILE lfn AT addr.	The equipment in the FST entry for the file is not supported for CIO functions.	Contact site analyst.	CIO
ILLEGAL ERROR EXIT ADDRESS.	Error exit address is beyond the job's current field length.	None.	1AJ
ILLEGAL EXTERNAL CALL.	RESEX did not recognize an external call.	Inform site analyst.	RESEX
ILLEGAL FILE COUNT.	The file count was not numeric.	Correct the N=n parameter.	ITEMIZE
ILLEGAL FILE MODE.	You tried to unlock a file which was in execute-only mode.	None.	LFM
ILLEGAL FILE NAME lfn AT addr.	The file name does not conform to established rules. The file name must have a maximum of seven characters, alphanumeric, and must be left-justified with binary zero fill.	Use valid file name.	CIO
ILLEGAL FILE STATUS.	Specified file status is invalid.	Check value.	LFM
ILLEGAL FILE TYPE.	The specified file is of a type not allowed in the requested operation. Possible causes include attempts to <ul style="list-style-type: none"> - Change a nonlocal file to file type library. - Designate a direct access file as the primary file. - Route the primary file. 	Verify that file type is appropriate.	LFM
ILLEGAL HOLL. CODE, RECxxx Cdyyyy.	Command yyyy (octal count) in record xxx was found to contain an invalid Hollerith code. The job was terminated without EXIT processing.	Correct command in error.	1AJ
ILLEGAL I/O REQUEST ON FILE lfn AT addr.	The system could not recognize the specified function code, or the code was not valid for the type of device to which the file was assigned. The system provides a dump of the FET on file OUTPUT.	Verify CIO function code being used.	CIO

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ILLEGAL ID CODE.	One of the following. - An identification code specified on the SETID command or macro is not a valid device identification code as defined in the installation EST. - An identification code not in the range 0-67B or 77B is present on the LDI command.	Reissue the request with the correct identification code.	LDI
ILLEGAL INPUT FILE.	An attempt was made to pack a file that is assigned to an interactive terminal. For example, file INPUT for an interactive origin jobs cannot be packed.	Verify that file being packed is not assigned to an interactive terminal.	PACK
ILLEGAL JOB ORIGIN TYPE FOR TRMDEF.	The TRMDEF command was entered from a job that was not of an interactive origin.	Retry from an interactive origin job or remove the TRMDEF command.	TRMDEF
ILLEGAL LABEL TYPE, lfn AT addr.	Illegal label type. The only valid label types are ANSI labeled and nonstandard labeled.	Use correct label type.	1MT
ILLEGAL LIST OPTION.	The list option specified in a CATLIST command is illegal.	Refer to description of CATLIST command for valid list options.	CATLIST
ILLEGAL LOAD ADDRESS.	The load address is less than 2.	Specify larger load address and retry.	1AJ
ILLEGAL MODIFICATION OF lfn AT addr.	Either you have attempted to shorten a modify-only file or the file cannot be modified at all.	Determine whether file can be modified.	CIO
ILLEGAL PARAMETER.	One of the following. - Parameter value is outside legal bounds. - You specified a parameter that cannot be included on the command. - Command is invalid.	Ensure accuracy of command/control statement.	IAFEX LISTLB CATLIST
ILLEGAL PASSWORD.	One of the following. - 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 command (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 name is disregarded.	Correct error and retry.	PASSWOR MODVAL PFILES
ILLEGAL QUEUE SPECIFIED.	You specified an invalid disposition code.	Retry with valid disposition code.	SUBMIT
ILLEGAL REQUEST.	A program issued a request (such as a MEMORY request) in which the request word was outside the job's field length.	If the message occurs after you entered a user break 2, ignore the message. If not, correct the program.	1MA
ILLEGAL REQUEST.	No parameters were specified on a DMPECS or DMDECS command.	Retry job with corrected command.	CPMEM
ILLEGAL RESOURCE COUNT.	Total resource demand exceeds maximum allowed, as defined by installation parameter MAXD (defined in RESEX).	Reduce RESOURC demands.	RESEX
ILLEGAL SORT PARAMETER.	An illegal parameter has been specified on the SORT command.	Consult description of SORT command for valid parameters.	SORT
ILLEGAL TERMINAL.	You have entered invalid family name, username, or password during login too many times.	Check login parameters and retry.	IAFEX
ILLEGAL TERMINAL REQUEST.	A command intended for interactive origin jobs only has been used in a noninteractive origin job.	None.	TLX
ILLEGAL TERMINAL TYPE FOR TRMDEF.	The TRMDEF command/ was entered from a terminal that was not a network terminal.	Retry from a network terminal.	TRMDEF

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ILLEGAL USER.	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.	NETVAL
ILLEGAL USER ACCESS.	You are not authorized to create direct access or indirect access files or to access auxiliary devices.	Contact site personnel concerning validations.	PFM
ILLEGAL USER ACCESS.	You tried to perform an operation for which you are not authorized. Possible causes include attempts to - 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 command, or determine proper validation requirements via LIMITS command.	DSD DO MSI QFSP RESEX TMA IAFEX
ILLEGAL USER ACCESS.	You tried to perform an operation for which you are not authorized. Possible causes include attempts to access a file or equipment which you are not validated to access.	Ensure accuracy of command or determine proper validation requirements via LIMITS command.	LFM
ILLEGAL USER ACCESS - CONTACT SITE OPR.	The security count for the user name specified has been decremented to zero. You are denied all access to the operating system until the operator resets your security count.	Contact site personnel to reestablish access.	CPM NETVAL
ILLEGAL USER ACCESS - CONTACT SITE OPR.	A bad entry on the VALIDUS file was accessed.	Contact site operator or analyst to check validation on that user name.	IAFEX
ILLEGAL USER CARD.	The user name or password could not be validated, or a secondary USER command was encountered while secondary USER commands were disabled.	Verify that user name and password are valid. If secondary USER commands are disabled, ensure that no secondary USER commands are present.	CPM ACCFAM
IMPROPER ACCESSIBILITY.	You did not specify the correct file accessibility on the LABEL command or macro, or volume accessibility was set and you attempted to assign the tape as unlabeled from a nonsystem origin job.	Ensure accuracy of request.	RESEX
IMPROPER LOGIN, TRY AGAIN.	The user has entered an invalid family name, user name, or password during login.	Check parameters and retry.	NETVAL IAFEX
IMPROPER VALIDATION.	A validation program (one containing a VAL= entry point, such as that used for CHARGE and USER) is required before continuing.	Verify that USER command precedes rest of job (followed by CHARGE, if required) and is the first command after the job.	TCS
IMPROPER VALIDATION.	A CHARGE command is required to continue.	Enter a CHARGE command.	1AJ
INCORRECTLY FORMATTED DATA.	A terminal definition parameter specified on the TRMDEF command or via the 0016 control byte was not correct.	Ensure accuracy of entry and retry.	IAFEX
INCORRECTLY-FORMATTED LIBRARY, lname.	There is an error in the ULIB record of file lname. The system does not create a directory for this file.	Correct the error in the ULIB record of lname.	LIBRARY
INDEX TOO LARGE ON OPEN, FILE lfn AT addr.	For an OPEN function, the random index length exceeded the index buffer size.	Set EP bit in FET+1 to get error status without abort or increase index buffer size.	CIO
INITIAL CONTROL STATEMENT LIMIT.	The number of control statements processed by the job exceeded the limit for which the user is validated. At this point, an additional eight statements are allowed for error processing.	Split the job into two or more jobs, reduce the number of control statements in the job, or request a larger control statement limit from site personnel.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INITIAL MESSAGE LIMIT.	MONITOR has detected that the message limit for the job has expired.	Decrease job size or increase limit.	1AJ
.INPUT FILE ERROR.	An error on the input file was encountered during the unpack operation.	None.	L072
INPUT FILE IN NORERUN STATUS.	Informative message.	None.	QFM
INPUT FILE IN RERUN STATUS.	Informative message.	None.	QFM
INPUT RESUMED.	The system overload condition has cleared.	Resume terminal input.	CCP
INPUT STOPPED message.	Because of a system overload condition, the network cannot accept terminal input. The message field is an installation-defined text string.	Wait until the message INPUT RESUMED is issued before proceeding.	CCP
INQUIRY COMPLETE.	Dayfile message indicating that the inquiry is completed.	None.	MODVAL
INSERTED LINES OVERLAP EXISTING LINES, AT nnnn. ENTER Y TO CONTINUE OR N TO STOP.	The MOVE, DUP, or READ command cannot be completed without line number overlap processing. The processing will begin at line nnnn.	Type Y to complete the edit command or type N to terminate the edit command.	IAF
INSUFFICIENT RESOURCES ON SYSTEM.	Resource demand exceeds the number of units physically available on the system.	Reduce resource demand.	RESEX
INSUFFICIENT STORAGE FOR LIBRARY GENERATION.	Additional memory is required for the LIBGEN command.	Increase field length and retry.	LIBGEN
INVALID 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 name. - The charge or project number exists but is inactive.	Check to see that charge and project numbers are correct and reenter.	CHARGE
INVALID CHARGE.	Dayfile and output file message indicating one of the following. - The charge or project number does not exist. - The project number cannot be used with this user name. - The charge or project number exists but is inactive.	Check to see that charge and project numbers are correct and reenter.	CHARGE
INVALID CONTROL STATEMENT.	The command specifies neither a JSN nor a UJN.	Specify a JSN or UJN in the command.	QGET
INVALID CONTROL STATEMENT	One of the following has occurred: - The CHARGE command has an invalid parameter or it has no terminator. - The user entered an illegal command or a command he was not validated to use.	Use the LIMITS command to check your validation and, if validated to use command, correct the format of the command and reenter.	CHARGE IAFEX
INVALID DC PARAMTER.	The specified DC parameter is not valid.	Correct the DC parameter and retry.	QGET DROP
INVALID DEFAULT CHARGE.	1. A default charge number or project number does not exist. 2. The default project number cannot be used with this account number. 3. A default charge or project number exists but is inactive.	Check to see that the default charge number and project number are valid.	CHARGE
INVALID ENTRY.	You entered something other than GO, RELIST, or a valid job sequence name.	Enter GO, RELIST, or a valid job sequence name.	RECOVER
INVALID ENTRY, TRY AGAIN	You entered incorrect characters in response to an SRU limit or time limit.	Enter correct characters.	IAFEX
INVALID INPUT FILE SYSTEM SECTOR	Job terminated without exit processing due to system error.	Inform site analyst.	1AJ
INVALID INSTRUCTION.	The CPU attempted to execute an illegal or unavailable instruction.	Analyze job output and dumps to determine the cause of the error.	1AJ
INVALID JOB START INFORMATION.	A system error occurred during charge processing.	Inform the site analyst.	CHARGE

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INVALID LINE LENGTH.	Either of the following out of bounds conditions exists with respect to the Ix, Nx, Ox, and H parameters of the L072 command. - (Ox+Nx).GT.H - (Ix+Nx).GT.(buffer length) where 1 .LE. x .LE. b	None.	L072
INVALID NOISE SIZE ON filenam.	Noise size on S, L, or F format tape does not meet copy requirements.	For copy, ensure that noise size on input file is greater than or equal to that of the output file. For TCOPI, correct noise size to meet requirements and retry (noise size of 8 for 7-track or 6 for 9-track tapes is required for X and SI format conversions).	COPY TCOPY
INVALID **/ OPTION.	An invalid command option followed the start.	Inform site analyst.	MFILES
INVALID OPTION X.	The option specified is not defined for ENQUIRE.	Check parameters on command and retry.	ENQUIRE
.INVALID PARAMETER.	The S or L parameter was entered as zero on the L072 command.	None.	L072
INVALID PARAMETER LENGTH.	A parameter has more characters than allowed.	Decrease the number of characters in the incorrect parameter and retry.	QGET DROP
tc - INVALID PARAMETER VALUE.	The value assigned to the terminal characteristic parameter specified by tc was not valid.	Correct error and retry.	TRMDEF
INVALID QUEUE TYPE.	System error.	Inform site analyst.	ENQUIRE
INVALID REPLY LIMIT REACHED.	The number of consecutive invalid entries exceeds the limit.	None.	RECOVER
INVALID SYNTAX	A required parameter is missing, parameters are given in the wrong order, or an invalid separator has been encountered.	Check proper command format and reenter the command.	IAF
INVALID USER ACCESS.	You made an authorized call to a utility reserved for system origin privileged users.	Ensure that you have System origin privileges before using this utility.	FOTD SUBSYST
INVALID USER ACCESS.	You attempted to use RECOVER in a noninteractive job.	Initiate an interactive job to use RECOVER.	RECOVER
INVALID USER ACCESS.	You attempted to use a command that can be used only in interactive jobs and your job is not interactive.	None.	IAFEX
proc IS A CCL PROCEDURE.	Procedure specified in CALL command is a CCL procedure.	Use BEGIN command.	CONTROL
pfn IS DIRECT ACCESS.	An indirect access file operation was attempted on a direct access file.	Use the appropriate direct access file request.	PFM
lfn IS EXECUTE ONLY.	The specified local file cannot be accessed by SAVE, REPLACE, APPEND or DEFINE since it is an execute-only file.	None.	PFM
pfn IS INDIRECT ACCESS.	A direct access file operation was attempted on an indirect access file.	Use the appropriate indirect access file request.	PFM
ITEMIZE COMPLETE.	Specified processing is finished.	None	ITEMIZE
ITEMIZING xxxx	Record xxxx is being processed.	None.	ITEMIZE
.IX OR OX NOT DEFINED.	The Ix or Ox parameter was not specified in conjunction with the Nx parameter on an L072 command.	Check the description of the L072 command in section 9.	L072
JOB ACTIVE.	The system is unable to process the command because the previous command has not completed.	Retry when current operation is complete.	IAFEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
JOB CARD ERROR (jobcmdnd)	The Job command of a routed job contains an error. jobcmdnd is the first 20 characters of the job command.	Correct the Job command and retry.	DSP
JOB CARD ERROR.	The Job command on the file being submitted is in error.	Compare the Job command in error with the Job command description in section 7. Also check your validation limits.	1AJ
JOB DETACHED, JSN=oldjsn JSN: newjsn, NAMI AF.	Informative message. The system creates the new JSN newjsn for your current terminal session.	No action required.	IAFEX
JOB IN NORERUN STATE ON RECOVERY.	Identifies a job recovered on level 0 deadstart that was aborted because it was in a no-rerun mode (due to NORERUN command or macro).	Refer to the NORERUN command or macro description.	1AJ
JOB NOT DETACHED.	Condition encountered during terminal job recovery attempt.	Retry command.	IAFEX
JOB NOT FOUND.	Condition encountered during terminal job recovery attempt.	Retry command.	IAFEX
JOB NOT RECOVERABLE.	Condition encountered during terminal job recovery attempt.	Retry command.	IAFEX
JOB NOT RERUN.	The system was unable to successfully rerun a job because of a mass storage read error or because the QFT is full.	Resubmit the job to the system.	1AJ
JOB REPRIEVED.	The job has been successfully reprieved.	None.	1AJ
JOB RERUN.	The job has been terminated and queued for input.	None.	1AJ
JOB STEP EXCEEDS ACCOUNT BLOCK.	You tried to set your job step limit to a value greater than your account block limit or tried to set your account block limit to a value less than your job step limit.	Check values on SETJSL and SETASL commands.	CPM
JOB STEP LIMIT.	The monitor detected the expiration of the job step SRU limit.	Reset job step limit with SETJSL command or macro and retry. If job step SRU limit is set at maximum, request increased SRU validation.	1AJ
JOBNAME IS jsn	An LDI command queued a file for input. The entered job has specified job sequence name.	None.	LDI
JSN NOT FOUND.	The specified JSN is not in the system or does not have the Specified disposition.	Use the ENQUIRE command to locate your jobs.	QGET
JSN OR UJN MUST BE SPECIFIED.	The JSN or the UJN must be specified within the command parameters.	Specify the JSN or the UJN and retry the command.	QGET
JSN/UJN SPECIFY DIFFERENT JOBS.	The JSN and the UJN parameter do not identify the same job.	Determine which parameter is correct and change the one in error.	QGET DROP
LABEL CONTENT ERROR, lfn AT addr.	A block read was the correct size for a label but one or more required fields (such as the label name) were incorrect.	Use LISTLB command to obtain label data.	1MT
LABEL MISSING, lfn AT addr.	During a read operation, a required label was missing.	Ensure that tape has label.	1MT
LABEL NOT EXPIRED.	You attempted to write on a tape with an unexpired label.	If current contents of tape can be sacrificed, have operator blank label tape. Otherwise, wait until label has expired.	1MT
LABEL PARAMETER ERROR ON OPEN, lfn AT addr.	Label fields did not match on open request. An additional message, FIELD BEGINNING AT addr NO COMPARE, specifying the decimal character position in HDR1 of the first field that did not compare correctly, is also issued.	Use LISTLB command to obtain label data.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
LDD - ARGUMENT ERROR - addr.	One of the following conditions is true: - FET address is less than 2. - FET address is greater than FL-2. - Auto recall was not set when LDD was called. addr Parameter block address.	Inform site analyst.	FDL
LDD - DEVICE ERROR - FILENAM AT addr.	An unrecovered device error occurred on file filename.	Inform site analyst.	FDL
LDD - I/O SEQUENCE ERROR.	I/O was attempted on a file that is currently active on another I/O request.	Check program for concurrent I/O.	FDL
LDQ - ARGUMENT ERROR - addr.	The FET address is less than 2 or greater than FL-4.	Inform site analyst.	FDL
LDQ - DEVICE ERROR - FILENAM AT addr.	An unrecovered device error occurred on file filename.	Inform site analyst.	FDL
LDQ - I/O SEQUENCE ERROR.	I/O was attempted on a file that is currently active on another I/O request.	Check program for concurrent I/O.	FDL
LDR ERROR.	Issued before one of the following errors. - OVERLAY NOT FOUND IN LIBRARY. - ARGUMENT ERROR.	Correct error and retry.	LDR
LFM ERROR.	LFM detected an error in the request. See other error messages in dayfile for more information.	Ensure that a valid LFM request is being made.	LFM
LFM ILLEGAL REQUEST.	One of the following: - The LFM function detected was not recognized as an invalid function. - An LFM function was issued without the auto recall bit set.	Verify that a valid LFM request is being used.	LFM
LIBEDIT ARGUMENT ERROR(S).	The LIBEDIT command contains an incorrect parameter.	Check the LIBEDIT command description for the correct format.	LIBEDIT
LIBGEN ARGUMENT ERROR.	An invalid parameter was used on the LIBGEN command.	Check the format of the LIBGEN command.	LIBGEN
LIBGEN FILE NAME CONFLICT.	The LIBGEN command named the same file as the input file and as the output file.	Change the input file or output file name.	LIBGEN
LIBRARY FILE NOT ON MASS STORAGE.	An attempt was made to generate a user library onto a file not assigned to mass storage.	Choose an output file that resides on mass storage.	LIBGEN
LIBRARY GENERATION COMPLETE.	Message issued when generation of a library is completed.	None.	LIBGEN
LIBRARY GENERATION FILE EMPTY.	The file to be processed is empty.	Verify that file is local to job and contains data.	LIBGEN
LIBRARY NOT FOUND, lname.	User library lname is not a local file.	Make file lname local.	LIBRARY
LINE NUMBER INCREMENT ERROR	Line number increment specified is less than one or greater than 4095.	Reenter command with correct line number increment.	IAF
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
LINE NUMBER LIMIT EXCEEDED	An attempt to move, duplicate, or read a group of lines in the edit file required a line number greater than 99999 to be assigned.	Resequence the file (using RESEQ command) and reenter the command.	IAF
LINE NUMBER OVERLAP ERROR	Two or more line ranges have at least one line number in common.	Reenter command with non-overlapping line ranges.	IAF
LINE RANGE ERROR	An invalid line number was specified in a line range.	Reenter command with correct line range.	IAF
LINE(S) TRUNCATED.	Informative message. File lfn1 contains lines longer than the maximum length processed by FCOPY. These lines were truncated when written to lfn2.	None.	FCOPY

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
nnnn LINE(S) TRUNCATED.	nnnn lines were truncated because they were longer than 150 characters.	Determine if relevant data was lost in the truncation. If possible, split the too long lines and repeat the copy operation.	COPYCF COPYCR COPYSBF
nnnnn LINES TRUNCATED ON FILE xxxxxxx	nnnnn lines on file xxxxxxx have been truncated.	None.	IAF
LIST OF FILES LENGTH TOO LONG OR ZERO.	The list of files exceeds the length of the local FNT, starts or extends beyond the job's field length, or has a length of zero.	Contact the site analyst.	1AJ
LIST OPTION ERROR.	An invalid list option was specified for the *L0* parameter in the LIBEDIT command.	Change the *L0* parameter to specify only valid options.	LIBEDIT
LISTLB ABORT.	A fatal error has occurred while processing the LISTLB command.	Refer to dayfile for cause of problem.	LISTLB
LISTLB COMPLETE.	Informative message indicating that the LISTLB operation has finished.	None.	LISTLB
LOCAL FILE LIMIT.	Your job has too many local files, or your job's local file limit was exceeded in a GET or ATTACH command or macro.	Return one or more local files and retry.	QFM, PFM
LOGIN ABORTED, TRY LATER.	Insufficient resources are available to allow you to gain access to the network.	None.	NETVAL
.L072 COMPLETE.	Informative message indicating that the program has completed processing.	None.	L072
LPeq, CHcc Abbbb INCOMPLETE TRANSFER. LPeq, CHcc CONTROLLER HUNG BUSY. LPeq, CHcc FUNCTION TIMEOUT. LPeq, CHcc Fffff REJ Paaaa,Cbbbb,Emmmm. LPeq, CHcc TURNED OFF.	Line printer messages. See CPeq...		110 1CD QAP
LPeq, CHcc PRINT ERROR LIMIT EXCEEDED.	Maximum number of consecutive print errors was detected on line printer. eq EST ordinal of line printer cc Channel number	Inform customer engineer.	B10 1CD
LPeq, CHcc RESERVED.	The line printer is reserved and cannot be connected on channel cc. eq EST ordinal of line printer cc Channel number.	Inform customer engineer.	110
LPeq, nnnn PRINT ERRORS.	Print errors detected on line printer. eq EST ordinal of line printer nnnn Octal number of print errors	Inform customer engineer.	B10 1CD
LReq, CHcc Abbbb INCOMPLETE TRANSFER. LReq, CHcc CONTROLLER HUNG BUSY. LReq, CHcc Fffff FUNCTION TIMEOUT. LReq, CHcc Fffff REJ Paaaa,Cbbbb,Emmmm. LReq, CHcc TURNED OFF.	580-12 line printer messages. See CPeq...		110 1CD QAP
LReq, CHcc PRINT ERROR LIMIT EXCEEDED.	Maximum number of consecutive print error was detected on 580-12 line printer. eq EST ordinal of line printer cc Channel number	Inform customer engineer.	B10 1CD
LReq, CHcc RESERVED.	The 580-12 line printer is reserved and cannot be connected on channel cc. eq EST ordinal of line printer cc Channel number.		110
LReq, nnnn PRINT ERRORS.	Print errors detected on 580-12 line printer. eq EST ordinal of 580-12 line printer nnnn Octal number of print errors	Inform customer engineer.	B10 1CD
LRC ERROR.	The longitudinal redundancy check character was read incorrectly (9-track NRZI). The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
LSeq, CHcc Abbbb INCOMPLETE TRANSFER. LSeq, CHcc CONTROLLER HUNG BUSY. LSeq, CHcc Fffff FUNCTION TIMEOUT. LSeq, CHcc Fffff REJ Paaaa,Cbbbb,Emmmm. LSeq, CHcc TURNED OFF.	580-16 line printer messages. See CPeq...		110 1CD QAP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
LSeq, CHcc Emmm PFC ERROR.	580-16 line printer message. See LReq...		110 1CD QAP
LSeq, CHcc PRINT ERROR LIMIT EXCEEDED.	Maximum number of consecutive print errors was detected on the 580-16 line printer. eq EST ordinal of the 580-16 line printer cc Channel number	Inform customer engineer.	B10
LSeq, CHcc RESERVED.	The 580-16 line printer is reserved and cannot be connected to channel cc. eq EST ordinal of line printer cc Channel number.		110
LSeq, nnnn PRINT ERRORS.	Print errors detected on 580-16 line printer. eq EST ordinal of 580-16 line printer nnnn Octal number of print errors	Inform customer engineer.	B10
LTeq, CHcc Emmm PFC ERROR.	580-20 line printer message. See LReq...		110 1CD QAP
LTeq, CHcc INCOMPLETE TRANSFER. LTeq, CHcc CONTROLLER HUNG BUSY. LTeq, CHcc Ffff FUNCTION TIMEOUT. LTeq, CHcc Ffff REJ Paaaa,Cbbbb,Emmm. LTeq, CHcc TURNED OFF.	580-20 line printer messages. See CPeq...		110 1CD QAP
LTeq, CHcc PRINT ERROR LIMIT EXCEEDED.	Maximum number of consecutive print errors was detected on the 580-20 line printer. eq EST ordinal of the 580-20 line printer cc Channel number	Inform customer engineer.	B10
LTeq, CHcc RESERVED.	The 580-20 line printer is reserved and cannot be connected to channel cc. eq EST ordinal of line printer cc Channel number.		110
LTeq, nnnn PRINT ERRORS.	Print errors detected on 580-20 line printer. eq EST ordinal of 580-20 line printer nnnn Octal number of print errors	Inform customer engineer.	B10 1CD
M.T. NOT AVAILABLE ON FILE lfn AT addr. MAGNET NOT ACTIVE.	The magnetic tape executive is not executing. One of the following occurred during a magnetic tape request. - There was not UDT address in the FST. - The UDT address in the FST was invalid. - The MAG Subsystem was not present.	Inform site operator. Inform site analyst.	C10 LFM RESEX
MASS STORAGE DIRECTORY NOT WRITTEN.	On a GTR command, you have requested that a mass storage directory record be written on a nonmass storage file.	Ensure that file resides on mass storage.	GTR
MASS STORAGE ERROR.	Job was terminated without exit processing.	Resubmit job.	1AJ
MASTER USER NUMBER REQUIRED.	Dayfile message indicating that the job did not enter a user name (via USER command). This is needed for a master user list run and for a master user inquire run.	Rerun job with USER command.	PROFILE
MAXIMUM FL ENCOUNTERED - JOB LIST INCOMPLETE.	The system has found too many recoverable jobs.	Proceed as normal or wait for other jobs to time out and try again.	RECOVER
MESSAGE LIMIT.	The number of messages issued by the job has exceeded the limit for which you are authorized. Message functions issued by compilers or applications programs that run in the job's field length are also counted as job dayfile messages and thus are subject to your dayfile message limit.	Split job into two or more jobs and retry.	1AJ
MFL LESS THAN ECS MINIMUM CM FL.	To use extended memory you must have a required minimum amount of CM FL. This message indicates that you do not have the required CM FL.	Increase CM FL.	1MA

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MFL REQUEST TOO SMALL, MINIMUM USED.	MFL request was less than CONTROL's RFL= value. CONTROL's RFL= value is used for this MFL request, thus allowing further MFL requests.	Check the description of field length control in section 3.	CONTROL
MISSING DEMAND FILE ENTRY.	Dayfile message indicating system problem. The overcommitment algorithm was initiated without a demand file entry having been defined previously.	Inform site analyst.	RESEX
jsqn MISSING RESOURCE.	Dayfile message indicating that system problem occurred. The system expected but did not find a resource unit assigned to the job with job sequence number jsqn. This could occur if the MAG subsystem was stopped while tapes were assigned.	Inform site analyst.	RESEX
MISSING VSN OR EQUIPMENT ASSIGNMENT.	Dayfile message indicating system malfunction (expected VSN or equipment assignment was not found).	Inform site analyst.	RESEX
MODE ERROR.	An arithmetic error, or CPU/CM parity error has been encountered during processing of job.	Examine reprieve block error flag to determine the nature of the error.	1AJ
MONITOR CALL ERROR, xxxx nnnnnnnn.	One of the following occurred during the processing of an RA+1 request: - The RA+1 call was not recognized by the system. - Bad parameters were supplied in the request. - Illegal user access was encountered. - An invalid request was included in the request.	Examine program to determine why invalid RA+1 call is being made.	1AJ
MT,Ccc,Eec,Hhhhhhhh,B.C.RESTART.	Magnetic tape controller controlware restarted.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,BAD ERASE.	Error detected after an erase was attempted to recover a write error.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhh,BID RECOVERY-x.	A single block mispositioning error was recovered by block ID recovery. If x is B, the error was caused by backspacing the tape too far; if x is F, the tape was not backspaced far enough.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,BLOCK TOO LARGE.	Data block is at least one byte longer than length bbbb shown in third line of message.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,BUSY.	Unit was still busy after one second.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhh,CHANNEL ILL.	Channel is not accepting function for status requests properly.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhh,CON.REJ.	Connect reject; unable to connect to the unit.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhh,CON REJ. MDI.	Connect reject; unable to connect to unit because of marginal detection indication (thermal warning). Unit turned off.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhh,CON.REJ.OFF.	Connect reject; unable to connect to unit. Unit turned off.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhh,Fnff,Pyyyy.	Function ff was rejected by the controller; yyyy is the address in 1MT where the function was initiated.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhh,Lbbbb,Bnnnnnn.	The length (bbbb) and block number (nnnnn) read from trailer bytes in block did not match the actual length or the block number read; given in previous message line.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,LOAD CHECK.	Load sequence failed on the unit.	Push CLEAR button and reload tape, or inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhh,MARGINAL DOWN.	Indicates controller failure. Channel has been logically turned off and maintenance is required.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhh,MARGINAL OFF.	Unit has been logically turned off because of read/write failure. This occurred when a special function to check the read/write path to a unit failed during initial label scan. Maintenance is required.	Inform customer engineer.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MT,Ccc,Eec,Hhhhhhhh,NO EOP.	No end-of-operation detected from unit within 1 second.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhh,NOISE.	A noise block was skipped on the tape.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,NOT READY.	Tape unit dropped ready status.	Make unit ready.	1MT
MT,Ccc,Eec,Hhhhhhhh,ON THE FLY.	Error was corrected as the data was read.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,POSITION LOST.	The last good block written cannot be found during write recovery.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,RECOVERED.	Previously reported error has been successfully recovered.	None.	1MT
MT,Ccc,Eec,Hhhhhhhh,STATUS.	Error type cannot be determined so actual controller status is returned.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhh,WRONG PARITY.	Tape was written in parity opposite that being read.	None.	1MT
MT,Ccc-e-uu,vsn,rw,xx,Ss,GSggggggg. MT,Ccc,Dddd...d. MT,Ccc,Uuu...u,Tttt. MT,Ccc,Fff,Iii,Bnnnnnn,Lbbbb,Pppppppp. MT,Ccc,Eec,Hhhhhhhh,type. or MT,Ccc-e-uu,vsn,rw,xx,Ss,GSggggggg. MT,Ccc,Dddd...d. MT,Ccc,Fff,Iii,Bnnnnnn,Lbbbb,Pppppppp. MT,Ccc,Eec,Hhhhhhhh,type.	Four or five-line message describing a magnetic tape hardware malfunction on a 66x or 67x tape unit. Message as illustrated indicates 7-track, model 667 or 677 unit. If NT appears in place of MT, message indicates 9-track, model 669 or 679 unit. Message is issued to error log and dayfile. The first line of each message provides the following information. cc-e-uu Channel, equipment (tape controller), and physical unit number of tape unit on which error was encountered. vsnn Volume serial number associated with tape on the specified unit. rw Read (RD) or write (WR) operation; any operation not involving an actual read or write is listed as a read. xx EST ordinal of the unit on which the tape was written. This is provided only for labeled tapes generated under NOS; otherwise, the field is blank. s Channel status. gggggggg General status of magnetic tape unit. Last byte is block ID. The MT,Ccc,Dddd...d line of the message provides the following information. cc Channel number; the channel number is repeated to allow the analyst to associate this message with the first message if errors are occurring on more than one tape channel at the same time. ddd...d Detailed status of magnetic tape unit. The MT,Ccc,Uuu...u,Tttt line of the message provides the following information. cc Channel number; repeated to associate this message with the previous messages. uu...u Unit status of the magnetic tape unit. tttt Third byte of the tape unit format parameters (refer to the magnetic tape subsystem reference manual for descriptions of unit format parameter fields). The MT,Ccc,Fff,...,Pppppppp line of the message provides the following information. cc Channel number; repeated to associate this message with the previous messages. ff Software function on which the error occurred. ii Error iteration; number of times error has been encountered on this unit without successful recovery. nnnnnn Block number on which error	Refer to the separate listing of the last line message (MT,...,type.) for the appropriate action.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
	<p>occurred.</p> <p>bbbb Length of block on which error occurred in octal bytes.</p> <p>pppppppp 1MT internal error parameters.</p> <p>The last line of each message provides the following information.</p> <p>cc Channel number; repeated to associate this message with the previous messages.</p> <p>ec Octal error code value.</p> <p>hhhhhhhh Parameters passed to the tape unit for the format function (refer to the tape drive's hardware reference manual for descriptions of the unit format parameter fields).</p> <p>type Additional description of the error. Refer to individual listing of the last line message.</p>		
MT/NT CONFLICT.	Conflict exists between 7-track and 9-track tape descriptors (track type, density, and conversion mode). For example, a request for a 9-track tape specified 200 bpi density. This message can also be issued if the device type specified in FET+1 conflicts with the track type specified in FET+1008?, bit 56. If dt=MT and bit 56 is set, the message is issued.	Ensure accuracy of command.	RESEX BLANK
MULTI-FILE NOT FOUND, lfn AT addr.	<p>Either LISTLB has reached the end of the multifile set or the requested file was not found. The following additional messages are also given.</p> <p>- REQUESTED SECTION xxxx.</p> <p>- FOUND SECTION yyyy.</p> <p>or</p> <p>- FILE IDENTIFIER NOT FOUND.</p> <p>The lfn, addr, xxxx, and yyyy given can be ignored.</p>	If LISTLB reached the end of the multifile set, the operation is complete and no action is required. Otherwise, ensure that the correct tape is being used and that it contains the desired file(s). All label parameters must match in order to position to the specified file.	1MT
MULTI-TRACK PHASE ERROR.	Multiple tracks were found to be in error at 1600 cpi, making recovery impossible. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
MUST HAVE SYSTEM ORIGIN PRIVILEGES.	The Mass Storage Subsystem Utility must be run from job with system origin privileges.	Make sure job has system origin privileges.	ASDEBUG ASDEF ASLABEL ASMOVE ASUSE ASVAL
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.	1MT
NO APPLICATIONS DEFINED - CONTACT SITE OPR.	You are not authorized for any applications.	Contact installation personnel.	NETVAL
NO CLOSING DELIMITER FOUND	No closing delimiter was found on a string. A string is considered to end after the last non blank character.	None.	IAF
NO CONNECT TIME AVAILABLE.	Dayfile message indicating that you have 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 you have accumulated the maximum CPU time allowed for the specified project number.	Contact installation personnel in order to increase maximum CPU time allowed.	CHARGE
NO ECS.	A DMPECS or DMDECS command or DED or DEP request was entered and no extended memory field length is assigned to you.	None.	CPNEM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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 HDR1 LABEL RETURNED ON OPEN.	No HDR1 label was found in the label buffer after the OPEN function was completed. Indicates a possible system error.	Inform site analyst.	LISTLB
NO INPUT FILE FOUND.	No valid input file exists; functions cannot be performed.	Verify that input file is valid.	QFM
NO JOBS FOUND.	The System can find none of your jobs.	None.	ENQUIRE
NO JSN-S RECOVERED.	The system encountered an EOR, EOF, or EOI while reading input directives.	None.	RECOVER
NO LINE NUMBER, FN=filenam.	Input line in file filenam does not contain a line number.	Correct and rerun.	
NO LINE NUMBER ON LINE	Either the primary file or a line-numbered READ file contain a line without a line number.	Use XEDIT or EDIT to remove/replace the bad line.	IAF
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 10-character limit.	Check the format of the input file. //RTN //RTN	SORT
NO LINE TERMINATOR AT EOR(S).	You attempted to copy a file in which the last line of one or more records did not have a line terminator.	Check the structure of the file to be copied.	COPYCF COPYCR COPYSBF
NO LINES FOUND	No line in the edit file meets the criteria specified by the edit command.	None.	IAF
NO MASS STORAGE AVAILABLE.	You attempted to reserve a track for a null primary file.	Retry later.	LFM
NO NEW FILE.	N=0 was inappropriately specified for the *N* parameter.	Either omit the *N* parameter to allow the default file *NEW* to be used or set it equal to the valid file name.	LIBEDIT
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
NO PRIMARY FILE.	A command needed the primary file as the default file. No primary file was found.	Specify a file name on the command or create a primary file with the NEW, OLD, or PRIMARY command, and then retry.	PFILES
NO PROCEDURE FILES EXECUTED.	An informative message indicating that there is no system prologue or user prologue (a procedure file automatically executed upon job initiation).	None.	CHARGE
NO READ/DESTINATION FILE SPECIFIED	No read or destination file was specified in the edit command.	Reenter command with a specified read or destination file.	IAF
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 RECOVERABLE JOBS.	You have no recoverable jobs.	No action required.	RECOVER
NO SOURCE FILE SPECIFIED.	No file name was specified on the SUBMIT command.	Specify a file name on the SUBMIT command.	SUBMIT
NO TERMINATOR IN CONTROL STATEMENT CALL	No terminator was found between the NOTE command and the first delimited line.	Insert delimiter and retry.	NOTE
NO USER STATEMENT EXECUTED.	A CHARGE command was entered before a USER command.	Enter a USER command before entering a CHARGE command.	CHARGE
NO WRITE ENABLE, ON lfn AT addr.	Either you attempted to write on a tape mounted with no write ring, or no write was allowed because of additional constraints described in an additional message line.	If no additional message line appears, ensure the inserting of a write ring by specifying the W processing option on the tape request (for example, P0=W on the LABEL command). Otherwise, refer to the	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
n NOISE BLOCKS DELETED. n NOISE BLOCKS PADDED.	Block(s) on S or L output tape smaller than noise size have been deleted/padded.	description of the message in the additional message line. None.	COPY
NON-EXISTENT EDIT VERB SPECIFIED	Internal error.	Notify site analyst.	IAF
NON-MATCHING CONVERSION.	Informative message indicating conversion mode on labeled 9-track tape differs from that specified by assignment request. System writes tape in specified mode, or reads tape with write ring out in correct mode. However, reading tape with write ring in or using wrong conversion mode generates conversion errors.	If reading tape with write ring in, return and reassign with correct conversion mode.	RESEX
NON-MATCHING DENSITY.	Informative message indicating that the density specified on the command or macro is not the same as the density of the assigned tape. Issued only to 9-track tapes with write ring out. 9-track tapes are read at the current density on tape. They are written at specified density if write initiated from load point; otherwise, tape is written at the current density on the tape.	None.	RESEX
NORERUN/RERUN IGNORED FROM TTY JOBS.	You entered NORERUN or RERUN command from an interactive terminal. The command is ignored.	None.	QFM CONTROL
filenam NOT DECLARED NRANDOM.	An EOF was encountered on the nonrandom file, filenam.	Verify that file name is in correct format.	LIBEDIT
.filenam NOT FOUND.	RESTART was unable to retrieve a file named, but not included, on filenam.	Verify that filenam is valid.	RESTART
nnnnnn NOT FOUND.	The specified JSN or UJN could not be found.	Verify that the name is valid.	ENQUIRE
filenam NOT FOUND. or un NOT FOUND.	One of the following: - The specified permanent file could not be found. - The specified user name could not be found. - You are not allowed to access the specified file. - The specified local file could not be found.	Verify that file name/user name is correct, that access permission has been granted, and that correct access is being attempted.	PFM
jsn NOT FOUND.	You specified a JSN that you cannot recover.	Refer to your list of recoverable jobs. Type RELIST for another list.	RECOVER
proc NOT FOUND.	Procedure specified in CALL command cannot be found.	Verify that procedure name is correct and retry.	CONTROL
xxx NOT IN PP LIB.	Dayfile message indicating that PP package xxx was not found in PP libraries.	Ensure that the correct PP package name was specified.	SFP
xxx NOT IN PP LIB. CALLED BY yyy.	Dayfile message indicating that PP package xxx, which was called by package yyy, was not found in the PP libraries.	Ensure that the correct PP package name was specified or inform site analyst.	SFP
filenam NOT ON MASS STORAGE.	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
NOTICE*** DATA READ ERROR.	Read error caused loss of words in the dayfile.	None.	DAYFILE
NT DENSITY CONFLICT.	9-track tape unit specified by EST ordinal on ASSIGN command does not support the required density.	Ensure that density compatible tape unit is specified.	RESEX
NT DRIVE CONFLICT.	One of the following. - 9-track tape unit specified by EST ordinal on ASSIGN command conflicts with other resource requirements for this job. In this case the system rejects assignment to prevent the job from deadlocking itself. - Increased resource demands (RESOURC command) cannot be satisfied due to	Check the description of the RESOURC command in section 7. Reduce resource demand which causes conflict.	RESEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
	conflicts with currently assigned resources (job would deadlock itself).		
**** OBSOLETE DIRECTIVE IGNORED.	The input directive entered is no longer meaningful and was ignored.	None.	MODVAL
OFFSW, x FROM JOB jsn	An OFFSW command was issued to the job by the job jsn. x represents the sense switch(es) that has been turned off.	None.	QAC 1RI
OFFSW, x FROM OPERATOR	An OFFSW command was issued to the job by the console operator. x represents the sense switch(es) that has been turned off.	None.	QAC 1RI
OLD MASTER FILE EMPTY OR MISPOSITIONED	None of the replacement file records were copied to the new file.	Ensure that the file is assigned to the job. If file is assigned, rewind the file.	COPYL COPYLM
OLDPL ERROR.	Update program library format contains an error.	Correct error and rerun.	
ONSW,x FROM JOB jsn.	An ONSW command was issued to the job by the job jsn. x represents the sense switches that sense switches that have been turned on.	None.	QAC 1RI
OP OPTION NOT VALID.	You specified some value other than T for the OP=op parameter.	Either omit the OP=op parameter or specify T for its value.	RECOVER
OPERATOR INPUT TERMINATION.	Job was terminated without exit processing.	Resubmit job.	1AJ
OPERATOR KILL.	The operator entered a KILL command to drop the job. This disallows erexit processing. A job with extended reprieve processing is reprieved once. Exit processing is allowed.	Correct job as needed and rerun.	1AJ
OPERATOR OVERRIDE.	The operator entered an OVERRIDE command to drop the job. This disallows exit, erexit, and reprieve processing.	Correct job as needed and rerun.	1AJ
OPERATOR RERUN.	The operator is rerunning the job.	No action required.	1AJ
OUTPUT NOT ASSIGNED TO TERMINAL.	The user must direct the output from HELP to the terminal.	Assign OUTPUT to the terminal.	HELP
OVER..	Another page of output exists.	Enter carriage return to view next page of output.	CCP
OVERLAPPING INSERT OR DELETE.	LIBEDIT encountered an overlap in the record names specified in the directives.	Correct the directives listed in the LIBEDIT output so that there is no overlapping of file names for insertions or deletions.	LIBEDIT
OVERLAY FILE EMPTY.	No data appears in the requested file.	Verify that overlay file is valid.	1AJ
OVERLAY FILE NOT FOUND.	The specified file was not available.	Verify that file is local to job and retry.	1AJ
OVERLAY LOST.	The specified overlay was not found.	Verify that file with specified overlay is local to job.	1AJ
OVERLAY NOT FOUND IN LIBRARY - ovlname.	The overlay ovlname was not found in the system library.	Verify that call is to valid overlay.	LDR
OVL	You have entered a logical line that exceeds the maximum logical line length and IAF has discarded it.	Reenter data in logical lines of shorter length.	IAFEX
PACK PARAMETER ERROR.	The PACK command contains too many or no file names.	Check the format of the PACK command in section 9.	PACK
n PARITY/BLOCK TOO LARGE ERRORS.	Parity and/or block-too-large errors have been encountered on the input file during the copy operation.	If dayfile shows block too-large errors have occurred and tape is S, L, or F format, increase block size and retry; otherwise, tape is probably assigned in the wrong format. If parity errors have occurred, the tape is bad and the data on it cannot be	COPY TCOPY

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
PARITY ERROR.	The tape could not be read/written correctly. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	correctly recovered. Retry or inform site analyst.	1MT
.PARITY ERROR - RESTARTED FROM kk.	Because RESTART detected a parity error in attempting to restart from the specified checkpoint nn, the alternate checkpoint kk was used instead.	None.	RESTART
PASSWORD TOO SHORT.	In the PASSWOR command the new password specified contains fewer characters than the minimum required.	Use a longer password.	PASSWOR MODVAL
PAUSE FROM JOB jsn	A PAUSE command was issued to the job by the job jsn.	None.	QAC 1RI
PAUSE FROM OPERATOR	A PAUSE command was issued to the job by the console operator.	None.	QAC 1RI
pfn PERMANENT ERROR.	The specified direct access file resides on the Mass Storage Facility (MSF) and has data errors that must be corrected. The error flag must be cleared or the file must be reloaded from a backup copy.	Inform site analyst.	PFM
PERMIT LIMIT EXCEEDED.	A PERMIT command or macro cannot be processed because the file specified already has the maximum number of permits allowed.	None.	PFM
PF STAGING DISABLED.	The specified direct access file resides on the Mass Storage Facility (MSF) and the site has temporarily disabled all MSF file staging.	Determine from site operator when MSF file staging will resume and retry the job at that time.	PFM
PF UTILITY ACTIVE.	The operation was not attempted because a permanent file utility was currently active.	Wait until PF utility is not active and retry.	PFM
PFM ABORTED.	Error flag detected at PFM control point.	Inform site analyst.	PFM
PFM ARGUMENT ERROR.	One of the following conditions is true: - The PFM call block is outside of the job's field length. - The system read an invalid command code. - Auto recall was not set when PFM was called.	Inform site analyst.	PFM
POSITION ERROR ON-filenam.	File filenam was not repositioned after being checkpointed because CHKPT detected an address error.	None.	CHKPT
POSITION LOST, lfn AT addr.	During write or read error recovery, the system could not find the last good block of data, making it impossible to successfully perform error recovery. Labels are not written after this error and existing data on the tape is not destroyed.	Retry operation on different tape and/or tape drive, if possible.	1MT
POSTAMBLE ERROR.	A missing or defective postamble was detected at 1600 cpi. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
PP CALL ERROR.	The monitor detected an error in a CPU request for PP action.	Verify that correct PP call is issued.	1AJ
PRIOR TAPE ASSIGNMENT LOST.	Magnetic tape executive has been dropped along with tapes assigned. All of the job's prior tape assignments are lost.	From an interactive job, return/unload all prior tapes and reassign. A batch job aborts and must be rerun.	RESEX
PROCEDURE FILE CALLS ABORTED.	The system prologue or user prologue aborted.	Check corresponding dayfile message for cause of abort.	CHARGE
PROCEDURE FILE EMPTY.	Procedure file specified contains no data.	Verify that procedure file contains data and retry.	CONTROL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
PROCESSING OPTION NOT APPLICABLE.	The processing options, P0=R and P0=D, are allowed only on a COPY command copying from mass storage, or an I or SI-binary format tape to an S or L format tape. The processing option, P0=T, is allowed on a TCOPI command that generates an E or B format tape.	Refer to the COPY or TCOPI command description in section 9, correct the command, and retry.	COPY TCOPY
PROFILE FILE CREATE COMPLETE.	Dayfile message indicating that the creation run is complete.	None.	PROFILE
PROFILE FILE DATA BASE ERROR.	Dayfile message indicating that the project file does not contain both a level 0 and level 1 block.	Ensure that the project file is local and contains a level 0 and level 1 block (at least one charge entry) and rerun.	PROFILE
PROFILE FILE INQUIRY COMPLETE.	Dayfile message indicating that the inquire run is complete.	None.	PROFILE
PROFILE FILE LIST COMPLETE.	Dayfile message indicating that the list of PROFILA is complete.	None.	PROFILE
PROFILE FILE REFORMAT COMPLETE.	Dayfile message indicating that the reformat run is complete.	None.	PROFILE
PROFILE FILE SOURCE COMPLETE.	Dayfile message indicating that the source run is complete.	None.	PROFILE
PROFILE FILE UPDATE COMPLETE.	Dayfile message indicating that the update run is complete.	None.	PROFILE
PROGRAM FILE EMPTY.	A load of an empty file was attempted.	None.	1AJ
PROGRAM STOP.	The system processed a program stop (00) instruction.	None.	1AJ
PROGRAM STOP AT addr.	The monitor detected a program stop instruction at address addr.	None.	1AJ
PROJECT NUMBER EXPIRED.	Dayfile and output file message indicating that the project number expiration date has occurred.	None.	CHARGE
PRU LIMIT.	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
PRU LIMIT, FILE lfn AT addr.	The job's mass storage PRU limit was exceeded during an attempt to write or extend this file.	Return one or more local files and retry.	CIO
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
PRUS REQUESTED UNAVAILABLE.	The number of PRUs specified via the S parameter on the DEFINE command or macro is not available.	Request smaller number of PRUs.	PFM
PURGE COMPLETE.	The files have been purged as requested.	None.	PURGALL
QAC ERROR ENCOUNTERED.	System error.	Inform site analyst.	ENQUIRE
QAP - BUFFER ARGUMENT ERROR.	System error.	Inform the site analyst.	QAP
QAP - ILLEGAL REQUEST.	System error.	Inform site analyst.	BIO
QAP - ILLEGAL USER ACCESS.	You tried to perform an operation for which you are authorized (for example, trying to use PFC arrays when not authorized to do so).	Ensure accuracy of command or determine proper validation requirements.	QAP
QFM FILE NAME ERROR.	The lfn specified is not a valid file name.	Verify file name.	QFM
QGET COMPLETE.	Informative only.	None.	QGET
READ AFTER WRITE, lfn AT addr.	You attempted to read a tape on which the last operation was a write.	Ensure accuracy of tape positioning commands (BKSP, BKSPRU, SKIPFB, or REWIND required to read after write).	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
READ FILE BUSY - filename.	The read file is found to be busy (direct access file only).	Retry after file is not busy.	SUBMIT
READ FILE EMPTY	The READ file specified does not exist or contains no data.	Reenter command with a correct read file.	IAF
READY.	The system is ready to process the next command.	Enter your next command.	IAFEX EDIT
READY DROP, lfn AT addr.	Unit dropped ready status.	None.	1MT
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
RECORD TOO LARGE ON filename.	An input record was encountered that exceeded S or L output tape block size.	Reduce input record size, or use COPY command to increase S or L tape block size or allow record splitting with PO parameter.	COPY COPYBF COPYEI
RECORD TOO LONG.	The record is too long for available memory. In response to a WBR request, the record length parameter was greater than or equal to the job's field length. Available memory is filled and the excess data is skipped.	Increase field length and rerun.	CPMEM
nnnnn RECORDS CONVERTED.	Informative message indicating the number of records (nnnnn) converted from one character set to another.	None.	CONVERT
n RECORDS NOT REPLACED.	Informative message. LIBEDIT encountered n records on a replacement file that were not named in the directives and did not replace old file records.	Either change the directives so that the replacement file is a no-replace file or include an *IGNORE directive listing the records that are not to be used.	LIBEDIT
n RECORDS SPLITS OCCURRED.	Multiple blocks per record have been written on an S or L output tape.	None.	COPY
RECOVERY COMPLETE.	Informative message.	None.	1IO
RECOVERY RERUN.	The system is rerunning the job following a Level 3 deadstart.	No action required.	1AJ
REMOVABLE PACKS OVERCOMMITMENT.	Removable pack request without NA selected causes resource overcommitment.	Retry later or retry with NA parameter on ATTACH, DEFINE, etc., with PN parameter specified.	RESEX
RENAME OF PROC TYPE NOT ALLOWED.	A RENAME of a PROC type record was attempted; this is not allowed.	None.	LIBEDIT
REPEAT..	Because of a temporary overload condition, the network has discarded the last logical line of input.	Reenter the information.	CCP NETVAL
REPRIEVE BLOCK ERROR.	An address is out of range or there is an invalid parameter in the reprieve parameter block at the time of an error. The message is also issued if the specified reprieve address itself is out of range. (1AJ issues this message for all errors except terminal interrupts processed by 1RI.)	Ensure parameter block is correct.	1AJ
REPRIEVE CHECKSUM BAD.	The computed checksum does not agree with the checksum specified in the parameter block at the time of the error. (1AJ issues this message for all errors except terminal interrupts processed by 1RI.)	Ensure interrupt handler is still intact. Ensure that code in the area for which checksum was computed has not changed.	1AJ
REQUEST UNDEFINED ON DEVICE, lfn AT addr.	The specified function cannot be performed on the device on which the file resides. The system provides a dump of the FET on file OUTPUT.	Verify that valid device is specified.	CIO
REQUIRED FL EXCEEDS VALIDATED LIMIT.	The job requires more field length than you are authorized to have.	Decrease the job size or request the site to increase your field length validation.	LIBEDIT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
RESEQ CONTROL CARD ERROR.	The RESEQ command contains a syntax error.	Correct error and rerun.	RESEQ
RESEQ ERRORS.	A resequencing error was encountered.	Refer to preceding message for more specific information about the error.	RESEQ
RESEQ NUMERIC PARAM ERROR.	A parameter which is supposed to be numeric contains a nonnumeric character.	Correct error and rerun.	RESEQ
RESERVED FILE NAME	Read or destination file name matches an internal scratch file. (ZZZZZG0-ZZZZZG7)	Rename file and reenter command.	IAF
RESERVED FILE NAME.	A reserved file name was incorrectly used.	Choose a nonreserved file name.	OPLEDIT EDIT DATADEF
RESEX ABORT - OPERATOR TERMINATION.	The operator entered a DROP, KILL or RERUN command. The system performed appropriate cleanup procedures before termination.	Determine reason for operator action. Rerun job if possible.	RESEX
RESEX ABORT - SYSTEM RESOURCE LIMIT.	The job terminated prematurely due to job time limit, SRU limit, or track limit. The system performs appropriate cleanup procedures before termination.	If error caused by SRU or time limit, increase resource limits. If caused by track limit, contact site analyst.	RESEX
RESEX ABORT - TERMINAL INTERRUPT.	You entered the interrupt or terminate sequence. The system performs appropriate cleanup procedures before termination.	None.	RESEX
RESEX DETECTED ERROR.	System error.	Inform site analyst.	LFM
RESEX FAILURE.	System error.	Inform site analyst.	PFM
RESOURCE DEMAND ERROR.	You attempted to decrease the number of scheduled units to less than the number of currently assigned units or increase the number of scheduled units to a point where a deadlock would occur.	Adjust RESOURC command parameters accordingly.	RESEX
RESOURCE ENVIRONMENT ERROR.	System error.	Inform site analyst.	RESEX
RFL BEYOND MFL. //SIG	The RFL request is greater than the maximum field length for a job step. //ACT sharing the pack. //ACT the pack.	Increase maximum field length with MFL command or SETMFL macro. //RTN	CPM //PUB
RESOURCE PF ERROR ec filename.	PFM error ec occurred when attaching resource file filename.	Inform site analyst.	RESEX
RESOURCE SCRATCH FILE ERROR.	An empty entry has been found in the overcommitment algorithm scratch file.	Inform site analyst.	RESEX
RESOURCE TYPE ERROR.	You specified an invalid resource type.	Ensure accuracy of command.	RESEX
jsn RESTARTED FROM yy/mm/dd. hh.mm.ss.	The checkpoint job identified by jsn was restarted from the checkpoint taken on the specified date and time. This message is issued whenever a checkpoint job is restarted.	None.	RESTART
ROLLOUT FILE BAD.	A job could not be rolled out correctly.	Inform site analyst. Check error log dayfile for the job that was aborted and the location of the bad rollout file.	1RI
ROUTE COMPLETE.	Informative message indicating ROUTE operation has completed.	None.	ROUTE
ROUTE COMPLETE. JOB NAME IS jsn.	A ROUTE command queued a job for input. The routed job has the system job name jsn.	None.	ROUTE
ROUTE CONTROL CARD ERROR.	Format of the command is incorrect.	Check the ROUTE command format in section 9.	ROUTE
ROUTE *DC* INCOMPATIBLE WITH *EC*.	You specified a DC/EC combination that is not valid.	If the DC parameter implies a print or punch disposition, the EC parameter must correspond.	ROUTEF

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ROUTE *FID* IGNORED.	Informative message listed for NOS/BE compatibility.	None.	ROUTE
ROUTE ILLEGAL KEYWORD.	Command contains a duplicate or invalid keyword.	Check the ROUTE command format in section 9.	ROUTE
ROUTE ILLEGAL *OT* PARAMETER.	The origin type specified by the OT parameter is illegal.	Verify that correct origin type is specified.	ROUTE
ROUTE INVALID *FC* PARAMETER.	The forms code specified on the command is not valid.	Check the command format and reenter command.	ROUTE
ROUTE INVALID SPACING CODE.	The SC=xx parameter on the ROUTE command specified a spacing code greater than 77B.	Correct the spacing code specified on the ROUTE command.	ROUTE
ROUTE *OT* NOT ALLOWED.	Your program is not system origin and only system origin jobs can use the OT parameter.	Do not attempt to use the OT parameter unless you have system origin privileges.	ROUTE
ROUTE *PRI* IGNORED.	Informative message listed for NOS/BE compatibility.	None.	ROUTE
ROUTE *REP* GT 31. DEFAULT USED.	The repeat count specified was greater than 31; it has been set to 0. This condition does not abort the program.	Use a repeat count less than 32.	ROUTE
ROUTE *ST* IGNORED.	Informative message listed for NOS/BE compatibility.	None.	ROUTE
ROUTE *TID* AND *FM/UN* CONFLICT.	The TID parameter was specified with either the FM or UN parameter. Neither one of these parameters can be specified along with TID.	Refer to section 9 for the correct format of the ROUTE command.	ROUTE
ROUTE *TID/FM/UN* and *ID* CONFLICT.	The ID parameter was specified with the TID, FM, or UN parameter.	Refer to section 9 for the correct format of the ROUTE command.	ROUTE
ROUTE *TID=xx - VALUE IGNORED.	Informative message listed for NOS/BE compatibility.	None.	ROUTE
filename ROUTED. JSN IS jsn.	filename is the name of the file which was routed. jsn is the job sequence name (JSN) assigned to the file.	None.	MFILES
RPV - PREVIOUS ERROR CONDITIONS RESET.	Normal message on RPV reset call.	None.	RPV
RTIME nnnnnn.nnn SECS.	Dayfile message output by RTIME command giving the real-time seconds count.	None.	1AJ
SECURE FILES - CHECKPOINT ABORT.	Indicates a local file being checkpointed had secure file status set.	None.	CHKPT
SECURE MEMORY, DUMP DISABLED.	You either attempted to dump memory protected by the system, or entered a memory dump request after a protected command or from a terminal.	Refer to Security Control in section 3 and the load/dump Memory commands in section 11. To obtain memory dumps from an interactive job, include the dump command in a procedure or in an ENTER command.	1AJ
SFP CALL ERROR.	System error.	Inform site analyst.	SFP
SFP/D00 BAD ERROR TEXT.	The specified systems text is not in the correct format.	Rebuild systems text using correct format.	SFP
SFP/D00 ERROR TEXT NOT FOUND.	The specified systems text could not be found.	Ensure the systems text specified is correctly installed on the system or call site analyst.	SFP
SFP/D00 ERROR TEXT NOT ON MASS STORAGE.	The specified systems text does not reside on mass storage.	Move systems text to mass storage.	SFP
SFP/D00 INVALID MESSAGE NUMBER.	The specified message number could not be found on the systems text.	Ensure that D00 was called with the correct message number.	SFP
SFP/xxx ILLEGAL FUNCTION CODE.	The specified function code to the program xxx is not defined.	Ensure the correct function code was used.	SFP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
SFP/xxx ILLEGAL ORIGIN CODE.	Dayfile message indicating that the function was illegal for the job's origin.	Inform site analyst.	SFP
SFP/xxx PARAMETER ERROR.	Dayfile message indicating that the parameter address was outside the field length.	Inform site analyst.	SFP
SFP/PFE I/O SEQUENCE ERROR.	Action was requested on a file that was already busy.	Wait until the file is not busy and retry.	SFP
SFP/RPV UNABLE TO RESET, NOT REPRIEVED.	Dayfile message indicating that an attempt was made to reset when the job had not been reprieved.	Inform site analyst.	SFP
SFP/SRP SPECIAL REQUEST PROCESSING ERROR.	Dayfile message indicating that the SPCW word was busy.	Inform site analyst.	SFP
SFP/STS UNKNOWN DEVICE TYPE/NAME.	The device code is not recognized by the system.	Check device code for validity.	SFP
SHARE TABLE ERROR.	System error.	Inform site analyst.	RESEX
jsn SHARE TABLE MISMATCH.	A system error occurred while processing the job with the specified job sequence name (jsn).	Inform site analyst.	RESEX
SINGLE FRAME ERROR.	A frame (NRZI only) containing all zeros was read; data will be at least one fram short. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Retry or inform site analyst.	1MT
SL NOT VALIDATED.	The SRU limit requested exceeds that for which you are authorized.	Request smaller SRU limit.	CPM
SPCW CALL ERROR.	A DMP= type call was made, and the program called is either not in the CLD or does not have a DMP= entry point defined.	Inform site analyst.	1AJ
SPECIAL REQUEST PROCESSING ERROR.	Error encountered in *SPCW* call word.	Inform site analyst.	1AJ
SRU LIMIT ENTER S TO CONTINUE OR CR KEY TO STOP:	You have exceeded your system resource unit (SRU) limit during a terminal session.	To continue, allocate nnnn additional SRUs with S,nnnn command or enter S alone to allocate an installation-defined number of SRUs.	IAFEX
STACK PURGING NOT DESELECTABLE.	You attempted to change the purging bit with the EREXIT function. This cannot be done on 6000 or CYBER 70 systems.	None.	CPM
STATUS ERROR, lfn AT addr.	An error was encountered during magnetic tape processing. A second message line describes the error in more detail.	Retry or inform site analyst.	1MT
STEP CONDITION.	Step mode flag set in the PSD register caused the program to interrupt at the end of a program instruction with an exchange jump to EEA (the error exit address in the exchange package).	Inform site analyst.	1AJ
SUBMIT COMPLETE. JOBNAME IS jsne	The SUBMIT command successfully queued the file for input. The system's job name for the entered job is jsne.	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
SUBSYSTEM ABORTED.	Your job was connected (either long term connection or wait response set) to a subsystem which aborted.	Retry later.	1AJ
SYSTEM ABORT.	Possible causes include the following. - Invalid USER command. - Attempt to access a restricted subsystem. - Operator evicted job. - Unrecognizable error flag. - SSJ= block outside field length. - 1RI detected a bad rollout file. - 1R0 detected an unrecoverable extended memory parity error during rollout of a job's extended memory field length.	If the cause was an invalid USER command or an attempt to access a restricted subsystem, correct the job and rerun. Otherwise, inform a site analyst.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
SYSTEM CHECKPOINT ABORT.	A subsystem has aborted due to a CHECK POINT SYSTEM request initiated by the operator.	None.	1CK
SYSTEM CLOSED.	The system is not available for use by terminals.	None.	NETVAL
SYSTEM ERROR.	A software or hardware system error occurred.	Inform site analyst immediately.	MODVAL PFM
SYSTEM ERROR.	The command is not recognized by the system.	Retry command.	IAFEX
SYSTEM ERROR.	A software or hardware system error occurred.	Inform site analyst immediately.	PASSWOR
SYSTEM ERROR.	LFM cannot complete the requested LFM function because the calling program has a DMP= entry point.	Inform site analyst.	BLANK RESEX
SYSTEM ERROR - NOTIFY SITE ANALYST.	An internal software error has occurred.	Notify the site analyst.	QGET
SYSTEM PROCEDURE ACCESS ERROR.	Informative message indicating that the system prologue did not execute.	Inform site analyst.	CHARGE
TABLE OVERFLOW.	The job field length is too small to hold the tables for processing the GTR command.	Increase field length and rerun job.	GTR
TAPE BLANK LABELED.	Dayfile message indicating that the blank label operation successfully completed.	None.	BLANK
TAPE BLOCK DEFINITION ERROR.	You attempted to define data block size via the FC or C keyword or noise block size via the NS keyword in such a manner that the system is unable to correctly define the size of the data block. The omission of the FC or C parameter on a command where it is required also causes this message to be issued.	Ensure accuracy of command.	RESEX
TAPE FORMAT PROBABLY WRONG.	This message is issued in addition to one of the following messages: - BLOCK SEQUENCE ERROR, lfn AT addr. - BLOCK TOO LARGE, lfn AT addr. - WRONG PARITY, lfn AT addr.	Ensure accuracy of format (F) parameter on command or macro.	1MT
TAPE FUNC REJ function.	Tape drive rejected the attempted function. Printed on the preceding line is the name of the current program or command buffer.	Press the space bar to retry the function.	CTI
TCS ILLEGAL REQUEST.	The translate commands (TCS) program was called with an invalid function code.	Correct the call to TCS and retry.	1AJ
TERMINAL BUSY.	You attempted to send a message to a terminal which was receiving output or had an input request outstanding. The message was lost.	Retry operation at a later time.	IAFEX
TERMINAL NOT ACTIVE.	You attempted to send a message to an inactive terminal.	None.	IAFEX
TERMINAL TIME OUT - JSN=aaaa.	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
TIME LIMIT.	The execution time limit for a job step expired resulting in job termination.	If a time limit was set for the job, include a SETTL command requesting a longer time limit for the job step. If the job step time limit was the maximum for which the user is validated, request a larger time limit or decrease the amount of processing to be performed by the job step.	1AJ
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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
TIMEOUT.	You have not responded to a login prompt within the system-specified time interval (usually about 2 minutes) and the terminal was disconnected.	Respond more quickly to prompts.	NETVAL
TL NOT VALIDATED.	The time limit requested exceeds that for which you are authorized.	Request smaller time limit.	CPM ACCFAM
TOO MANY ARGUMENTS.	The number of arguments on the command exceeds that allowed by the program.	Reformat the command with an allowable number of arguments.	SUBMIT TCS
TOO MANY ARGUMENTS.	The number of arguments in the command exceeds GOB.	Check format of command and reduce the argument count.	1AJ
TOO MANY LIBRARY NAMES SPECIFIED.	The number of libraries specified on the LIBRARY command exceeds the maximum of 24. The current global library set remains in effect.	Correct the command.	LIBRARY
TOO MANY PARAMETERS.	More parameters were entered (including null parameters) than are allowed for command.	Ensure accuracy of entry.	GTR IAFEX NOTE
TOO MANY PERMANENT FILES.	The number of files in your catalog exceeds your limit.	You must purge one or more permanent files in order to save or define additional files.	PFM
TOO MANY USER LIBRARIES SPECIFIED.	The number of user libraries specified exceeds the number allowed based on the number of system libraries specified. The current global library set remains in effect.	Correct the command.	LIBRARY
TOO MUCH INDIRECT ACCESS FILE SPACE.	The cumulative size of the indirect access files in your catalog exceeds your limit.	One or more indirect access files must be purged or shortened to allow additional permanent file space.	PFM
TOO SMALL BUFFER SPACE	Internal error.	Notify site analyst.	IAF
TOTAL ASSIGNED COUNT ERROR.	System error (sum of individual resource assigned counts differs from total assigned count in demand file entry).	Inform site analyst.	RESEX
TOTAL DEMAND COUNT ERROR.	System error (sum of individual resource demand counts differs from total demand count in demand file entry).	Inform site analyst.	RESEX
TRAILER BLOCK COUNT ERROR, lfn AT addr.	The block count in the EOF1 or EOV1 label did not match the block count maintained by the tape executive during the read operation.	Inform site analyst.	1MT
TRMDEF ERRORS.	Errors were found in the TRMDEF command. Additional error messages are listed at the terminal or, if the L parameter was included, on the output file specified by L.	Correct error and retry.	TRMDEF
TRMDEF PROCESSING COMPLETE.	The TRMDEF command was processed with no errors.	None.	TRMDEF
TTxx, ASSIGNED TO filenam, VSN=vsfn.	The specified file name is assigned to the tape volume with the specified VSN that is mounted on tape drive TTxx (the seven-track (MT) or a nine-track (NT) tape unit with EST ordinal xx).	Informative message. Processing of the tape file can begin.	RESEX
UJN NOT FOUND.	The specified UJN is not in the system or does not have the the specified disposition.	Use the ENQUIRE command to locate your jobs.	QGET
UNABLE TO READ IQFT FILE.	An attempt to initialize inactive queues failed because the IQFT file could not be read.	Retry operation.	IMS MSI
UNIDENTIFIED PROGRAM FORMAT.	The file you requested to be loaded was in an unrecognizable format.	Check the format of the file.	1AJ
UNIT HAS MOTION PROBLEMS.	The tape unit cannot properly write the tape. The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Resubmit your job, using a different tape unit or inform site analyst.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
UNIT HUNG UP ON EOP OR BUSY, lfn AT addr.	Unit did not receive EOP on unit busy.	Inform site analyst.	1MT
UNIT PROBLEM.	Unit check bit is set in detailed status (67x units only). The dayfile message STATUS ERROR, lfn AT addr precedes this message and specifies the file (lfn) and the address addr.	Try another unit or inform site analyst.	1MT
UNLABELED TAPE REQUIRED - filename.	An S tape used for E, B, or X tape conversion must be unlabeled.	Use unlabeled tape.	TCOPY
UNNECESSARY CIO FCT. nnn ON lfn AT addr.	The read or write CIO function specified by nnn was unnecessary, since your I/O buffer was already full (read) or empty (write) and no data could be transferred for this CIO call. The message is issued to the job dayfile only if analyst logging is enabled on the system (console entry).	Ignore or correct program to be more efficient.	CIO
UNRECOGNIZABLE TYPE SPECIFIED.	The type of the source file was not valid.	Check type of source file.	L072
UNRECOGNIZED BACKSPACE CODE.	The specified backspace code was not 0, 1, 2, or 3.	Refer to the COPYX command description, correct the error, and retry.	COPYX
UNRECOGNIZED TERMINATION CONDITION.	The specified termination record count or record type was not recognized.	Refer to the COPYX command description, correct the error, and retry.	COPYX
UNRECOVERABLE ERROR ON filename.	An unrecoverable error such as wrong parity, density change, or ready drop was detected on the input file.	Either the tape was bad or the tape drive is malfunctioning.	COPY TCOPY
UNRECOVERED MASS STORAGE ERROR.	A command or overlay calls a system library program. During the load, an unrecoverable read error occurred.	Inform site analyst.	1AJ
UNSUPPORTED CODE SET.	You attempted automatic recognition of a character code set which is not supported at your installation.	Use a supported code set.	CCP
UNUSUAL END-OF-FILE ENCOUNTERED.	GTR detected an EOF mark not preceded by an EOR mark.	Dump file to determine file format error.	GTR
UPDATE COMPLETE.	Your password has been changed.	None.	PASSWOR
UPDATED - type/name	The record with type and name on the old file was replaced with the matching record from the replacement file.	None.	COPYL COPYLM
**** USE *FILINFO* IN PLACE OF *PRM* CALL ****.	You used a *PRM* call to obtain file permission data.	Use a *FILINFO* call to obtain the desired information.	SFP
**** USE *FILINFO* IN PLACE OF *STS* CALL ****.	You used a *STS* call to obtain file allocation data.	Use *FILINFO* calls to obtain the desired data.	SFP
USE SCOPY FOR FILE STRUCTURE.	You entered R parameter on the LIST command. The function of the LIST,R command has been replaced by the SCOPY command.	Enter SCOPY,fn. for file structure information.	IAFEX
USER BREAK TWO ENCOUNTERED.	Informative message.	None.	1AJ
USER DAYFILE PROCESSED.	The job's dayfile dump is complete.	None.	DAYFILE
USER DIRECTIVE INVALID.	The accounting data needed to process the job was not furnished.	Check file and add a user command.	SUBMIT
USER LIBRARY NOT FOUND.	The requested entry point is in the directory file, but the associated user library is not present.	Regenerate the user library.	1AJ
USER LIBRARY NOT ON MASS STORAGE.	The requested entry point is in the directory file, but the associated user library is not on mass storage.	Assign the user library to a mass storage device.	1AJ
USER NOT VALIDATED FOR ECS MINIMUM CM.	You wer authorized for the extended memory specified on the Job command but was not authorized for at least 10000 octal words of memory.	Request a larger CM validation limit from site personnel.	ACCFAM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
**** USER NUMBER ALREADY EXISTS.	You attempted to create a user name that already exists. Your line of input is disregarded.	None.	MODVAL
**** USER NUMBER LIMIT.	An attempt was made to validate more than 4095 user names for the specified charge and project number entry.	Rerun with a maximum of 4095 user names under one charge/project entry.	PROFILE
USER STATEMENT NOT EXECUTED.	The accounting data needed to process the job was not furnished because the USER command was omitted or invalid.	Check file and add a USER command then retry.	SUBMIT
VERIFY ERRORS.	Verification errors were encountered during comparison of data/records/files.	Check output file for errors.	VERIFY
VERIFY FL ABOVE USER LIMIT.	Field length for L or F tape verify exceeds your current maximum.	Use MFL to increase maximum field length or reduce block size on L or F tapes.	VERIFY
VSN FILE ERROR.	System error (VSN file entry does not match job identification).	Inform site analyst.	RESEX
WAIT	The system is waiting for system resources to become available.	None.	IAFEX
WAITING FOR MAGNET.	The job is waiting for the magnetic tape subsystem to be activated.	Wait for the operator to activate MAGNET or terminate job.	RESEX
WAITING FOR PN=packnam, typ.	The job is waiting for the operator to mount pack packnam on device type typ.	Wait until the operator mounts the requested pack or terminate job.	RESEX
WAITING FOR RESOURCE FILE.	The job is waiting for the resource demand file or VSN file to become available.	Wait until resource file becomes available or terminate job. To operator: If job is not rolled out and this message persists, inform site analyst or drop the job. If the operator decides to override an interrupted job at this point, the preview data in the demand file k is not cleared and the E,P display continues to show the VSN request associated with the job until you log off or issue a subsequent request for tape or pack.	RESEX
WAITING FOR RESOURCES.	The job is waiting for sufficient resources to allow assignment of the tape/pack without causing a system deadlock.	Wait until the resources become available or terminate job.	RESEX
WAITING FOR STORAGE.	Issued to DSD B and J displays. BIO is waiting to increase its field length or for a buffer to become available.	None.	110
WAITING FOR STORAGE (CM).	The job is waiting for additional central memory to be made available.	Wait for additional storage or terminate job.	1MA
WAITING FOR STORAGE (ECS).	The job is waiting for additional extended memory to become available.	Wait for additional storage or terminate job.	1MA
WAITING FOR VSN= vsn, typ.	The job is waiting for the operator to mount the tape with VSN vsn on resource type typ (MT, HD, PE, or GE). VSN= SCRATCH indicates that any scratch tape is acceptable.	Wait for the operator to mount the tape or terminate the job.	RESEX
WAITING ON TRACK LIMIT.	The job is waiting for additional tracks on the family device containing the resource demand and VSN files.	Wait for the additional tracks or terminate job.	RESEX
WARNING - CONVERT PROCEDURE TO CCL	The procedure contains a KCL IF, CALL, or GOTO command. KCL will not be supported in a future release of NOS.	Convert procedure to the structure described in section 4.	CONTROL
WRITE ON READ-ONLY FILE lfn AT addr.	Either you 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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
WRITE OVER LABEL.	You attempted to write over the VOL1 label.	Have the operator blank label the tape.	1MT
WRITE OVER LABEL ILLEGAL ON lfn AT addr.	You are not allowed to destroy the VOL1 label.	Use LISTLB command to obtain label data.	1MT
WRONG PARITY, lfn AT addr.	A seven-track tape is being read in opposite parity from which it was written.	Ensure accuracy of format parameter (F) on command or macro.	1MT
XL BUFFER/FET PARAMETER ERROR, lfn AT addr.	One of the following: <ul style="list-style-type: none"> - HDR1 label in extended label buffer or FET contains a nonnumeric display code value in a numeric field. - Character count in header word preceding labels in the extended label buffer does not equal 80. 	Correct condition that caused error and retry.	1MT
ZZZZLD NOT ON MASS STORAGE.	The directory file ZZZZLD does not reside on mass storage.	Assign the directory file to mass storage.	1AJ
200BPI WRITE ILLEGAL.	The tape unit (667 or 677) cannot record data at 200 bpi.	Specify a different tape density.	1MT

GLOSSARY

C

Abort

To terminate a program or job when an error condition (hardware or software) exists from which the program or computer cannot recover.

Account Block

That portion of job from initiation to termination or entry of a CHARGE command which signals a new account block.

Allocatable Device

A storage device that can be shared by more than one job.

Alphanumeric

Consisting of alphabetic and/or numeric characters only.

Application

A program resident in a host computer that provides an information storage, retrieval, and/or processing service to a remote user via the network. IAF is an example of an application.

Application Switching

The process of leaving the control of one application and entering the control of another, without going through another complete login sequence.

Argument

Refer to Parameter.

ASCII

American National Standard Code for Information Interchange. The standard character set and code used for information interchange between systems.

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

1. Beginner's All-Purpose Symbolic Instruction Code, an elementary programming language.
2. The subsystem that uses the BASIC compiler.

Beginning-of-Information (BOI)

A file boundary that marks the beginning of a file. You can reference the BOI by name. For a labeled tape file on a single volume, the HDR1 represents BOI. If the file is a multivolume (multireel) tape file, BOI is the HDR1 label of the first volume. If the tape file is an ANSI multifile file, BOI is the HDR1 associated with the beginning of the multifile.

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.

Block

Blocking is the grouping of user records for efficiency in transfer between memory and storage devices. For magnetic tapes, it is the information between interrecord gaps.

BOI

Refer to Beginning-of-Information.

Byte

A group of 12 bits. Five bytes comprise a 60-bit central memory word. Bytes are numbered 0 to 4 from the left.

Catalog

The list of names of permanent files belonging to a particular user name; this list also contains information about the permanent files.

Character

Any alphabetic, numeric, or special symbol that can be encoded. This term applies to the graphic characters for an input or output device, and to uniquely encoded control characters used by a terminal.

Checkpoint

The process of writing on a magnetic tape or mass storage file a copy of the user's central memory, the system information used for job control, and the names and contents of all assigned files that are identified in a CHECKPT request.

Checkpoint File

File on which the results of a partially completed job are dumped when a checkpoint request is processed.

Command

A sequence of words and characters that call a system routine to perform a job step. The command must conform to format specifications. You can usually place a comment after the command terminator.

Command Record

The first, and possibly only, record of a job file consisting of command images that start with the Job command and end with the first EOR, EOF, or EOI. Also refers to a procedure containing commands.

Compile

To translate a program from a higher level programming language (for example, FORTRAN or BASIC) into machine instructions called object code.

Control Character

1. A terminal-dependent character that is a part of interactive status commands and terminal definition commands. It gives these commands a preemptive status so that they are processed immediately.
2. A character whose occurrence within a file queued for printing, affects carriage control. Examples are the carriage return and line feed characters.

Control Point

The portion of central memory that is assigned to a job. When a job is allocated a portion of central memory, it becomes eligible for assignment to the central processor for execution.

Control Statement

Refer to Command.

CYBER Loader

The utility that prepares programs for execution by placing program instruction and data blocks in central memory.

CYBER Record Manager (CRM)

A software product that allows a variety of record types, blocking types, and file organizations to be created and accessed. Products like COBOL 5, FORTRAN Extended 4, FORTRAN 5, Sort/Merge 4, ALGOL 5, and DMS 170 use CRM to manage execution time input/output. Neither the input/output of the operating system nor that of most of the system utilities such as COPY or SKIPF is implemented through CYBER Record Manager. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

Dayfile

A chronological file created during job execution which forms a permanent accounting and job history record. Dayfile messages are generated by operator action or when commands are processed. A copy of the dayfile is printed with the output for each noninteractive job. You must explicitly request it in an interactive job.

Deadstart

The process of initializing the system by loading the operating system library programs and any of the product set from magnetic tape or disk. Deadstart recovery is reinitialization after system failure.

Default

A system-supplied option used when you do not supply the option.

Detached Job

An interactive origin job that you have dissociated from your terminal by using the Detach command. It continues to execute under system control. You can reattach an executing detached job with the RECOVER command.

Device

Peripheral equipment.

Direct Access File

A NOS permanent mass storage file that can be attached to your job. All changes to this file are made on the file itself rather than on a temporary copy of the file (contrast with Indirect Access File).

Directive

An instruction subordinate to a system command.

Display Code

A 6-bit-character-code set used to represent alphanumeric and special characters.

Disposition Code

A two-character mnemonic that indicates the manner in which the system is to dispose of a queued file. The DC=dc parameter of the LDI, QGET, ROUTE, SETJOB, and SUBMIT commands specify a disposition code.

EBCDIC

Extended Binary Coded Decimal Interchange Code. An 8-bit code which represents a set of 256 characters.

Empty PRU/Record

A PRU that contains no user data. Same as zero-length PRU.

End-of-File (EOF)

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 EOI. For labeled tape, EOF and EOI (denoted by the EOF1 label) are the same. For multifile tape files, EOF and EOI do not correspond. In the product set manuals, an end-of-file is also referred to as an end-of-partition.

End-of-Information (EOI)

The end of data on a file. Information appearing after this point is not considered part of file data. In card decks, it is a card with a 6/7/8/9 multiple punch in column one. On mass storage devices, it is the position of the last written data. On labeled tape, it is the EOF1 label. CYBER Record Manager defines end-of-information in terms of file residency and organization.

End-of-Record (EOR)

An indicator that marks the end of a logical record. Also referred to as end-of-section.

End-of-Tape

A reflective strip near the end of a magnetic tape. It is used to signal termination of operations on a particular tape volume. At least 5.5 meters (18 feet) of tape must follow this marker.

Entry Point

A location within a program or procedure that can be referenced from other programs. Each entry point has a unique name with which it is associated.

EOF

Refer to End-of-File.

EOI

Refer to End-of-Information.

EOR

Refer to End-of-Record.

EOT

Refer to End-of-Tape.

Error Flag

A character or bit that signals the occurrence or presence of an error.

EST Ordinal

The number designating the position of an entry within the equipment status table established at each installation. Devices are identified in operator commands by EST ordinals. The EST ordinal is sometimes referred to as equipment number.

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.

Extended Instruction Stack Purging

Extended instruction stack purging applies to models 825, 835, and 855 only. It causes the instruction lookahead registers to be purged under a wider range of conditions than normal. Under normal conditions, the lookahead registers for models 825, 835, and 855 are purged by the execution of a return jump instruction (010), UEM read instruction (011), exchange jump instruction (013), or unconditional branch instruction (02). Under extended instruction stack purging, these registers are also purged by the execution of any conditional jump instruction (03 through 07) or any central memory store instruction (50 through 57 with $i = 6$ or 7). The instructions that cause instruction stack purging vary from model to model for other computer systems (refer to the hardware reference manual for the individual computer system).

Extended Memory

Optional additional memory. Various called extended core storage (ECS), extended semiconductor memory (ESM), unified extended memory (UEM), large central memory (LCM), and large central extended memory (LCME) depending on the model of the computer system. Extended memory contains 60-bit words. It has a large amount of storage and fast transfer rates. Extended memory can be used only for program and data storage, not for program instructions. Special hardware instructions exist for transferring data between central memory and extended memory.

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 users associated with the additional families to access their permanent files on more than one system.

Family Name

The name that is associated with a group of permanent file devices.

Field Length (FL)

The central memory and extended memory assigned an executing job.

File

1. A set of information that begins at beginning-of-information (BOI), ends at end-of-information (EOI), and is referenced by a local file name.
2. That portion of a multifile file terminated by an end-of-file (EOF).
3. Data recorded on a magnetic tape beginning after an HDR1 label and ending before an EOF1 label.

NOS commands requiring a parameter that is a file name refer to definition 1.

Commands requiring a parameter that specifies the number of files refer to definition 2.

Definition 3 applies only to labeled magnetic tapes.

File Environment Table (FET)

A table within a program's field length through which the program communicates with operating system input/output routines. One FET exists for each file in use by the program.

File Information Table (FIT)

A table through which a user program communicates with CYBER Record Manager. For direct processing through CRM, you must initiate establishment of this table. All file processing executes on the basis of information in this table. You can set FIT fields directly or use parameters in a file access call that sets the fields indirectly. Some product set members set the fields automatically.

File Set

One or more tape files referred to by the set identifier on a tape assignment command. A file set may consist of:

1. One file recorded on a single volume.
2. More than one file recorded on a single volume.
3. One file recorded on more than one volume.
4. More than one file recorded on more than one volume.

All files within a file set have the same set identifier in their HDR1 labels.

FIT

Refer to Local File Information Table.

Flag

A character or bit that signals the occurrence or presence of a particular condition.

FNT

Refer to File Name Table.

FORTTRAN

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.

Frame

A tape recording unit made up of 1 bit from each tape track (7 bits for 7-track tape and 9 bits for 9-track tape). Each frame on a coded tape usually represents one character.

FTN5

The subsystem that uses the FORTRAN Extended version 4 compiler.

Generation

The position of a file within a series of files, each file developed from the preceding file. The generation number and generation version number of a tape file can be entered in its HDR1 label.

Global Library Set

A set of user libraries you specify on the LIBRARY command. The system conditionally searches these libraries when it attempts to process a command.

Graphic

A character that can be printed or displayed. Refer to Control Character.

IAF

Refer to Interactive Facility.

Indirect Access File

A NOS permanent file that you access by making a temporary copy of the file (GET, OLD, or LIB command). You create or alter it by saving or substituting the contents of an existing temporary file (REPLACE or SAVE command).

Input File Type

The file type the system assigns to file INPUT. Its first record is a command record which may be followed by records containing data, directives, or programs used by job steps.

Instruction Stack Purging

The process of clearing the instruction stack in the central processor. It affects the speed and accuracy of the central processor.

Interactive Facility (IAF)

An application program that provides a terminal operator with interactive processing capability. The interactive facility makes terminal input/output and file input/output appear the same to an executing program.

Interrecord Gap

Space skipped between the writing of data blocks on magnetic tape.

Job

A set of commands and the data and directives used by those commands. A batch job must begin with the Job and USER commands. An interactive job is all activity associated with a terminal session from login to logout unless you detach a job during the session. After a job detachment, the terminal session becomes a new job.

Job Sequence Name (JSN)

The unique, system-defined name assigned to every executing job or queued file. The JSN is a string of four alphabetic characters.

Job Step

An individual command or loader sequence. A group of job steps forms a job stream or command record.

Label

A block at the beginning or end of a magnetic tape volume or file, which serves to identify and/or delimit that volume or file.

Level Number

An octal number (0, 1, or 17) in the terminating marker of a PRU. A level number of 17 designates an EOF. A level number of 0 designates an EOR. A level number of 1 designates that the line of data is from a terminal and is not an empty line.

Library

1. A collection of programs or routines.
2. A file containing records that are accessed individually.
3. A file searched by CYBER Loader for entry points referenced by a program.
4. A file containing compressed records in Modify or Update format.

Library File

A read-only file that can be accessed by several users simultaneously or a file you specify on a LIBRARY command.

Line

A unit of data terminated by a zero-byte terminator. Unit used in interactive I/O, line printer output, and card reader input.

Load Point

Metallic strip marking the beginning of the recordable portion of a magnetic tape. Data, including labels, is written after the load point.

Local File

Any file that is currently associated with a job. Local files include all temporary files and attached direct access files.

Local File Name

The file name assigned to a file while it is local (assigned) to a job. The name is contained in the FNT.

Local File Name Table (FNT)

A system-managed table that contains the local file name, the file type, and other job control information.

Local Operator (LOP)

The local operator manages the communications elements of the network. Local operator functions may be performed at the system console or at a remote terminal.

Locked File

A file on which you cannot write.

Logical Device

One or more physical disk units known to the system as a single device.

Logical Record

A data grouping that consists of zero or more PRUs and ends with a short PRU or a zero-length PRU.

Login

A procedure that you must perform initially to establish interactive communication with an application.

Macro

A sequence of COMPASS source statements that are saved and then assembled whenever called.

Mass Storage

Magnetic disk or extended memory that can be accessed randomly as well as sequentially.

Master Device

Mass storage device that contains your permanent file catalog entries; all

your indirect access files; and all, part, or none of your direct access files.

Multifile File

A file containing more than one logical file. It begins at BOI and ends at EOI. On a labeled tape, a multifile file is delimited by corresponding HDR1 and EOF1 labels.

Multifile Set

A tape file set having more than one tape file.

Network

A data and message switching and routing system used to provide communication between terminals and applications.

Noise

Any tape block less than the minimum acceptable block size for its data format. Noise is discarded by the system.

Nonallocatable Device

A device such as a magnetic tape unit, card reader, card punch, or line printer which can be used only by one job at a time.

Object Code

Executable machine language instructions. An object code program need not be recompiled each time it is executed.

Order-Dependent

Used to describe items which must appear in a specific order.

Order-Independent

Used to describe items which need not appear in any specific order. Parameters, particularly those with keywords, may be order-independent.

Origin Type

A job attribute that indicates how a job entered the system. The four origin types are interactive origin, batch origin, remote batch origin, and system origin.

Packet Switching Network (PSN)

A system that provides data communication service between different terminals and computer systems. The PSN is usually licensed as a common carrier. GTE Telenet is an example of a PSN.

Subscribers use a protocol defined by a standards body such as CCITT to interface with a PSN. The packet assembly/disassembly (PAD) access defines the terminal interface. The X.25 protocol defines the PSN interface with NOS.

Parameter

A value that follows a command name or function name and that alters the behavior of the command or function.

Parity

In writing data, an extra bit is either set or cleared in each byte so that every byte has either an odd number of set bits (odd parity) or an even number of set bits (even parity). Parity is checked on a read for error detection and possible recovery.

Passive Procedure

A procedure that does not have the parameter-prompting capability. If the call to the procedure contains errors, the job step terminates and the procedure is not executed.

Password

1. A system access word that must be used in addition to the user name at login.
2. A file access word that defines access to a particular file by alternate users.

Peripheral Processor (PP)

The hardware unit that performs input and output through data channels.

Peripheral Processor Library Directory

A list of all peripheral processor program names.

Permanent File

A file stored on mass storage. This file is cataloged by the system so that its location and identification are always known to the system. Permanent files cannot be destroyed accidentally during normal system operation. They are protected by the system from unauthorized access according to privacy controls specified by the file owner.

Permanent File Device

Mass storage defined by your site to hold permanent files.

Permanent File Family

Permanent files which reside on the family devices of a specific system.

Physical Record Unit (PRU)

The amount of information transmitted by a single physical operation of a specified device. For mass storage files, a PRU is 64 central memory words (640 characters); for magnetic tape files, the size of the PRU depends upon the tape format. A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

Post Radix

A letter following a numeral that indicates the base numbering system.

PP

Refer to Peripheral Processor.

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

A temporary file created with the OLD, NEW, LIB (interactive jobs only), or PRIMARY command. The primary file is assumed to be the file on which most system operations are performed unless another file is specified. There can be only one primary file associated with a user's job.

Private Auxiliary Device

Auxiliary device associated with a specific user name. Only that user name may create files on the device, although other users may be permitted to access files which reside on the device.

Procedure

A user-defined set of instructions that can be referenced by name. The instructions consist of procedure directives and system commands.

Procedure File

A file containing one or more procedures.

Prologue

A procedure the system executes any time you initiate a job. You can define your own prologue to complement the system prologue.

PRU

Refer to Physical Record Unit.

PRU Device

A magnetic tape file or a mass storage device. Records on these devices are written in PRUs.

PSN

Refer to Packet Switching Network.

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 7).

Queue Priority

The variable associated with a queued file that controls when the system processes the file.

Queued File

A file you have submitted for job entry or a file of job output waiting for processing. The queued file may have a print, punch, plot, wait, or input disposition.

Random Access

Access method by which any record in a file can be accessed at any time. Applies only to mass storage files with an organization other than sequential. Contrast with Sequential Access.

Random Access File

A file whose records are accessed through a directory containing the address of each record.

Read-Only Permission

If you have read-only permission on a file, you can read and execute the file. You cannot write, modify, append, or purge the file.

Record

A unit of information. In CYBER Record Manager and its language processors, a record is a unit of information produced by a single read or write request. Eight different record types exist within CRM. You define the

structure and characteristics of records within a file by declaring a record format.

A NOS record is a second-level subdivision of a file. It is the smallest subdivision that NOS recognizes. A NOS record can consist of a group of CRM records.

Recovery

The process by which you reattach a job to the terminal. You could have explicitly detached the job using the Detach command or the job could have been detached because of a system failure or terminal disconnection before logout. You can initiate recovery processing with the RECOVER command.

Reference Address (RA and RAE)

RA is the absolute central memory address that is the starting or zero relative address assigned to a program. Addresses within the program are relative to RA. RA+1 is used as the communication word between your program and the system. RAE is the absolute extended memory starting address assigned to a program.

Rolled Out

The status of an executing job that has been temporarily removed from central memory.

Rubout Characters

Characters created by pressing the RUB OUT key on the terminal. These are considered null input by the system and are required in paper tape output.

Scheduling Priority

The variable associated with an executing job that controls the scheduling of the job for CPU time.

Security Count

The number of security violations you have left before you are denied access to the system.

Sequence Number

1. Number at the beginning of each line of a file.
2. For tape labels, number indicating position of a file within a file set.

Sequential Access

A method in which only the record located at the current file position can be accessed. Refer to Random Access.

Sequential (SQ) File

A file in which records are accessed in the 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.

Service Class

A job classification based on a job's origin type or job's status. It determines how a job is serviced as it flows through the system.

Short PRU

A PRU that does not contain the maximum number of character data allowed for a PRU. Refer to Zero Length PRU.

Source Code

Code input to the computer for later translation into executable machine language instructions (object code).

Special Editing Mode

Mode in which the cancel input character, backspace character, and line feed character are not treated as special characters but are passed to IAF as input data.

SRU

Refer to System Resource Unit.

Standard Labeled Tape

A tape with labels conforming to American National Standard Magnetic Tape Labels for Information Interchange X3.27-1969. Also called a system labeled tape.

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 NOS.

System File

A file that can be accessed only by a system program.

System Resource Unit (SRU)

A unit of measurement of system usage. The number of SRUs includes the central processor time, memory usage, and input/output activity.

Tape Format

A parameter that specifies the internal recording format of a magnetic tape.

Tape Mark

A delimiter written on tapes under operating system control to separate label groups, files, and/or labels. Interpretation depends on the tape format.

Temporary File

A file that is assigned to your job that vanishes once it is returned to the system (either explicitly or at job termination). Local copies of indirect

access permanent files are temporary files; direct access permanent files assigned to your job are not temporary files.

Terminal Name

A network-supplied name used to identify the terminal to the local operator.

Time Slice

The amount of CPU or CM time a job is allowed to use before the system lowers its scheduling priority to allow other jobs to execute.

Timed/Event Rollout

A condition where an executing job has been temporarily removed from central memory but will be rolled back into central memory only when a specified event (such as a file is no longer busy) or a specified time period has elapsed.

Transparent Input Mode

Mode in which all characters entered at the terminal are sent to IAF as 7-bit codes (or as 8-bit codes, depending on the parity) without translation. Normally, characters entered at the terminal are translated and stored as 6-bit and/or 12-bit display codes. This mode is initiated either with the 0006 control byte or with the IN=XK or IN=XP terminal definition commands. In former interactive systems this mode was referred to as binary mode.

Transparent Submit Mode

The mode used in a submit file to transfer information that contains reformatting directives without processing the directives.

User Break 1 Sequence

The character or sequence of characters that causes an executing program to be interrupted (also called the interruption sequence).

User Break 2 Sequence

The character or sequence of characters that causes an executing program to be terminated (also called the termination sequence).

User Index

A number the system assigns each user name and uses for internal record keeping.

User Index Hash

A string of four alphabetic characters derived from your user index that serves as the default user job name (UJN) for interactive jobs. It appears on the banner page/card of all line printer and card punch output. You can ascertain your user index hash with the ENQUIRE command.

User Job Name (UJN)

In general, the user-defined name for an executing job or queued file. If you fail to specify a UJN, the system uses your user index hash. The UJN must be a string of seven or fewer alphanumeric characters. Unlike the job sequence name (JSN), the UJN for jobs is not always unique.

User Name

A one-to seven-character name that identifies the user to the system and for which there is assigned a user index that allows access to permanent files.

User Number

Refer to User Name.

Validation File

File containing validation information for all users (user names, passwords, resources allowed, and so on).

Volume

A reel of magnetic tape. A given file can encompass more than one volume.

Volume Serial Number (VSN)

A one-to six-character identifier that identifies the volume of magnetic tape for the operator.

VSN

Refer to Volume Serial Number.

Word

A group of bits (or 6-bit characters) between boundaries imposed by the computer system. A word is 60 bits in length. The bits are numbered 59 through 0 starting from the left. It is also composed of five 12-bit bytes, numbered 0 through 4 from the left.

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.

Write Ring

A circular device inserted into a tape reel indicating to the tape unit that it can write on that reel. NOS checks for the presence of a write ring when assigning the tape if you request it.

Zero Byte Terminator

The 12 bits of zero in the low order position of a central memory word that are used to terminate a line of coded information. A record with such a terminator in CYBER Record Manager is a zero-byte record (Z type record) or a line.

Files created interactively at a terminal and by commands that manipulate coded lines contain zero-byte terminated records. The image of cards input through a card reader also has such a terminator.

In display code for the 64-character set, two colons create 12 bits of zeros. If two consecutive colons occur in a file that contains zero-byte terminated records, they might be stored in the lower order portion of a word and create a zero-byte terminator.

Zero Length PRU

A PRU that contains system information, but no user data.

6/7/8/9 Multipunch

A card with the characters 6, 7, 8, and 9 multipunched in column 1. Signifies a card deck EOI.

6/7/9 Multipunch

A card with the characters 6, 7, and 9 multipunched in column 1. Signifies a card deck EOF.

7/8/9

A card with the characters 7, 8, and 9 multipunched in column 1. Signifies a card deck EOR.

SAMPLE JOB OUTPUT

D

This appendix lists the output information printed for the sample job shown below. The job consists of the following commands.

```
/JOB.  
OUTPUT.  
/USER  
CHARGE,*.  
DISPLAY,OT.  
LIMITS,L=MYLIMS.  
REWIND,MYLIMS.  
COPY,MYLIMS.  
REWIND,MYLIMS.  
COPYSBF,MYLIMS.
```

OPERATING SYSTEM - NOS 2 crtndate PRINTED - yy/mm/dd 13.48.33.

UJN - OUTPUT FAMILY - MATH JOB ORIGIN - BATCH.
CREATING JSN - ABNS USER NUMBER - FREE211 SERVICE CLASS - BATCH.

```
AAAAAAAAAA JJJJJJJJJJ LL          AAAAAAAAAA AAAAAAAAAA P999999999 44 NN SSSSSSSSS
AAAAAAAAAA JJJJJJJJJJ LL          AAAAAAAAAA AAAAAAAAAA P999999999 44 NN SSSSSSSSS
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AAAAAAAAAA JJJ JJ LL          AAAAAAAAAA AAAAAAAAAA P999999999 44 NN SSSSSSSSS
AAAAAAAAAA JJJ JJ LL          AAAAAAAAAA AAAAAAAAAA P999999999 44 NN SSSSSSSSS
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJ JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJJ JJ LL          AA AA AA AA P9 P9 44 44 NN NN NN SS S
AA AA JJJ JJ LLLLLLLLLL LL          AA AA AA AA P999999999 44 NN SSSSSSSSS
AA AA JJJ JJ LLLLLLLLLL AA AA AA AA P999999999 44 NN SSSSSSSSS
```

The first line indicates the version of the operating system, the creation date of that version (crtndate), the printing date (yy/mm/dd), and the time of day the printing occurred.

The second line contains the user job name (UJN), as specified on the Job command; family name, and job origin.

The third line contains the job sequence name (JSN), the user name associated with the job, and the job's service class.

The first four block letters indicate the user index hash and the last four, the job's JSN.

The first listing of MYLIMS results from the COPY command and the second listing, from the COPYSBF command. Note that the character in column 1 is lost in the first listing.

LIMITS FOR FREE211 FM/MATH yy/mm/dd. 13.44.40.

MAXIMUM NUMBER OF -

MAGNETIC TAPE UNITS THAT MAY BE ASSIGNED.....	3
REMOVABLE AUXILIARY DEVICES THAT MAY BE ASSIGNED	2
CPU SECONDS ALLOWED FOR EACH JOB STEP/10B.....	UNLIMITED
CENTRAL MEMORY WORDS ALLOWED/100B.....	2037B
EXTENDED MEMORY WORDS ALLOWED/1000B.....	0B
JOB'S THAT CAN BE DETACHED.....	3
DEFERRED BATCH FILES.....	8
PERMANENT FILES ALLOWED IN YOUR CATALOG.....	SYSTEM
PRUS AVAILABLE FOR ANY ONE INDIRECT FILE.....	192
PRUS AVAILABLE FOR ANY ONE DIRECT FILE.....	1024
PRUS AVAILABLE FOR ALL INDIRECT FILES.....	4096
PRUS AVAILABLE PEP JOB.....	66048
MESSAGES THAT MAY BE ISSUED TO THE DAYFILES.....	1008
BATCH COMMANDS THAT MAY BE PROCESSED.....	UNLIMITED
CARDS THAT MAY BE PUNCHED PER FILE.....	UNLIMITED
LINES THAT MAY BE PLOTTED PER FILE.....	512
LINES THAT MAY BE PRINTED PER FILE.....	UNLIMITED
SRUS ALLOWED PER JOB.....	UNLIMITED

OTHER CHARACTERISTICS -

USER INDEX.....	36515B
DEFAULT CHARGE NUMBER ASSIGNF	6913
DEFAULT PROJECT NUMBER ASSIGNED.....	AB51296
SYSTEM PROLOGUE.....	46B
USER PROLOGUE.....	STARTUP
INTERACTIVE SUBSYSTEM.....	NULL
INTERACTIVE CHARACTER SET MODE.....	NORMAL

THE FOLLOWING ARE VALID USER PERMISSIONS -

- CHANGE YOUR PASSWORD
- CREATE DIRECT ACCESS FILES
- CREATE INDIRECT ACCESS FILES
- ACCESS/CREATE LIBRARY FILES
- ASSIGN MAGNETIC TAPES
- ACCESS OR CREATE FILES ON AUXILIARY DEVICES

THE FOLLOWING ARE VALID NETWORK APPLICATIONS -

- INTERACTIVE FACILITY (IAF)
- REMOTE BATCH FACILITY (RBF)
- TRANSACTION FACILITY (TAF)
- TERMINAL VALIDATION FACILITY (TVF)

MAXIMUM NUMBER OF -

MAGNETIC TAPE UNITS THAT MAY BE ASSIGNED.....	3
REMOVABLE AUXILIARY DEVICES THAT MAY BE ASSIGNED	2
CPU SECONDS ALLOWED FOR EACH JOB STEP/10B.....	UNLIMITED
CENTRAL MEMORY WORDS ALLOWED/100B.....	2037B
EXTENDED MEMORY WORDS ALLOWED/1000B.....	0B
JOB'S THAT CAN BE DETACHED.....	3
DEFERRED BATCH FILES.....	8
PERMANENT FILES ALLOWED IN YOUR CATALOG.....	SYSTEM
PRUS AVAILABLE FOR ANY ONE INDIRECT FILE.....	192
PRUS AVAILABLE FOR ANY ONE DIRECT FILE.....	1024
PRUS AVAILABLE FOR ALL INDIRECT FILES.....	4096
PRUS AVAILABLE PER JOB.....	66048
MESSAGES THAT MAY BE ISSUED TO THE DAYFILES.....	1008
BATCH COMMANDS THAT MAY BE PROCESSED.....	UNLIMITED
CARDS THAT MAY BE PUNCHED PER FILE.....	UNLIMITED
LINE'S THAT MAY BE PLOTTED PER FILE.....	512
LINE'S THAT MAY BE PRINTED PER FILE.....	UNLIMITED
SRUS ALLOWED PER JOB.....	UNLIMITED

OTHER CHARACTERISTICS -

USER INDEX.....	36515B
DEFAULT CHARGE NUMBER ASSIGNED.....	6913
DEFAULT PROJECT NUMBER ASSIGNED.....	AB51296
SYSTEM PROLOGUE.....	46B
USER PROLOGUE.....	STARTUP
INTERACTIVE SUBSYSTEM.....	NULL
INTERACTIVE CHARACTER SET MODE.....	NORMAL

THE FOLLOWING ARE VALID USER PERMISSIONS -

- CHANGE YOUR PASSWORD
- CREATE DIRECT ACCESS FILES
- CREATE INDIRECT ACCESS FILES
- ACCESS/CREATE LIBRARY FILES
- ASSIGN MAGNETIC TAPES
- ACCESS OR CREATE FILES ON AUXILIARY DEVICES

THE FOLLOWING ARE VALID NETWORK APPLICATIONS -

- INTERACTIVE FACILITY (IAF)
- REMOTE BATCH FACILITY (RBF)
- TRANSACTION FACILITY (TAF)
- TERMINAL VALIDATION FACILITY (TVF)

```

13.44.40.OUTPUT.
13.44.40.USER, FREE211,.
13.44.40.CHARGE,*
13.44.40.* CHARGE(1234,PROJ834)
13.44.40.DISPLAY,OT.
13.44.40.    1    18
13.44.40.LIMITS,L=MYLIMS.
13.44.40. INQUIRY COMPLETE.
13.44.40.REWIND,MYLIMS.
13.44.40.COPY,MYLIMS.
13.44.40. EOI ENCOUNTERED.
13.44.40. REWIND,MYLIMS.
13.44.40.COPYSBF,MYLIMS.
13.44.40. EOI ENCOUNTERED.
13.44.41.UEAD,    0.002KUNS.
13.44.41.UEPF,    0.008KUNS.
13.44.41.UEMS,    0.386KUNS.
13.44.41.UECP,    0.013SECS.
13.44.41.AESR,    2.061UNTS.
13.44.41.$OUT(* /OP-E)
13.44.41. NO FILES PROCESSED.
13.44.41.*DAYFILE(OUTPUT,JT-D)
13.48.38.UCLP, 42,070,    0.256KLNS.

```

The first line lists the current date (yy/mm/dd), the time of job termination (13.44.41), and the time of job initiation (13.44.40). The dayfile includes a listing of the commands, system-supplied status messages, and program output, if any. Spaces precede status messages and program output. Each line includes the time the message was issued to the dayfile.

Six of the last nine lines specify the type and amount of system resources the job used. This job used 0.002 kilounit of application activity, 0.008 kilounit of permanent file activity, 0.386 kilounit of mass storage activity, 0.013 seconds of central processor time, and 2.061 system resource units. The first two numbers after UCLP gives the machine identifier as 62 and the EST ordinal of the printer as 070. The job produced 0.256 kiloline (192 lines) of printable output.

The \$OUT and \$DAYFILE commands are system-supplied and ensure that the job's dayfile is printed.

Depending on the resources used, additional information may be included in the dayfile. Refer to Job Completion in section 3 for the formats of these messages.

TERMINAL CHARACTER CONVERSION

E

Normal input mode from an ASCII code terminal uses a 63- or 64-character set where all lowercase alphabets are converted to uppercase characters. Under ASCII mode, the characters 74 and 76 represent the beginning of a 74xx or 76xx escape sequence. Under normal mode, the characters 74 and 76 are treated as data rather than escape codes. ASCII and normal modes apply to both input and output.

DATA INPUT

The manner in which the system interprets the characters entered from a terminal depends on whether you request ASCII or normal mode. For example, if you enter

aAbBcCdDeEfF

when in ASCII mode, the central memory equivalent is:†

Bit position	59		47		35		23		11		0
Code	76 01		01 76		02 02		76 03		03 76		
Character	a		A	b		B	c		C		
Code	04 04		76 05		05 76		06 06		01 01		
Character	d	D	e		E	f		F	end-of-line		

† Partial words are zero-filled; partial bytes are blank-filled.

However, if you enter the characters in normal mode, the characters are mapped into the 64-character subset of the ASCII character set that contains only uppercase letters; then the central memory equivalent is:

Bit position	59	47	35	23	11	0
Code	01 01	02 02	03 03	04 04	05 05	
Character	A A	B B	C C	D D	E E	
Code	06 06	00 00	00 00	00 00	00 00	
Character	F F	end-of-line				

Refer to appendix A for further description of the code sets for interactive jobs.

DATA OUTPUT

Data output is in either a 64/63- or 128-character set, depending on whether the terminal is in normal or ASCII mode. When the terminal is in normal mode, the codes 74 and 76 represent data rather than escape codes. In ASCII mode, 74 and 76 are treated as the beginning of an escape sequence.

Data can also be transmitted to or from a terminal through a paper tape reader. Refer to appendix N.

CARD FILE DATA CONVERSION

F

Data within the system is stored in binary or coded format. Binary data is variable length central memory images. Coded data consists of display-code characters. Each coded line is stored as an even number of characters. If an odd number of characters is entered, the system appends a space to make it even.

This appendix describes the formats for punch cards. It also describes the conversion performed by the system on data transferred between the system and card readers and punches.

When using the 64-character set, avoid using consecutive colons (00 characters). It is possible for these colons to be interpreted as an end-of-line. An end-of-line is defined as 12 to 66 zero bits, right-justified in one or two central memory words. If consecutive colons appear in the lower 12 bits of a central memory word, they are interpreted as an end-of-line rather than as colons.

Example:

The following characters are punched on a coded card beginning in column 1.

: : : : : : : : A : : : : : : : : AA

This appears in memory as follows:

Bit position	59	47	35	23	11	0
Code	00 00	00 00	00 00	00 00	00 01	
Character	: :	: :	: :	: :	: :	: A
Code	00 00	00 00	00 00	00 00	01 01	
Character	: :	: :	: :	: :	: A	: A
Code	00 00	00 00	00 00	00 00	00 00	
Character						end-of-line

However, if the characters were copied with the COPYSBF utility, the following appears.

Bit position	59	47	35	23	11	0
Code	55 00	00 00	00 00	00 00	00 00	
Character	end-of-line					
Code	01 00	00 00	00 00	00 00	00 01	
Character	A	:	:	:	:	A
Code	01 55	00 00	00 00	00 00	00 00	
Character	A	end-of-line				

Because the COPYSBF utility shifts each line one 6-bit character to the right and adds a space, copying nine colons puts 12 zero bits in the last byte of the first word. This is interpreted as an end-of-line.

NOTE

If a colon is the last character of an input line, the system appends a space to preserve the colon and then appends an end-of-line. If needed, a second space is added to ensure an even number of 6-bit characters. Refer to figure F-1.

INPUT CARD FILE FORMATS

The system reads cards in coded and binary formats. The following conditions apply in both formats.

- A card with a 7/8/9 punched in column 1 is an EOR mark.
- A card with a 6/7/9 punched in column 1 is an EOF† mark.
- A card with a 6/7/8/9 punched in column 1 is an EOI mark.

The remainder of each card is ignored except for columns 79 and 80 of the EOR and EOF cards. These columns can contain the keypunch conversion mode for the input records that follow.†† Conversion modes are discussed in Coded Cards.

†The 6/7/9 keypunch mark is not supported by RBF.
 ††HASP terminals can support other forms of separator cards such as /*EOR and /*EOI. (Refer to the Network Products Remote Batch Facility Reference Manual.)

CODED CARDS

Cards are read in Hollerith punch code. The 3447 card reader controller converts the Hollerith code to internal BCD code and passes the data to the card reader driver. The driver converts the data from internal BCD code to display code. Up to 80 characters can be transferred per card. Trailing blank characters are deleted. If a line has an odd number of characters, one trailing blank is added to make it even. In order to preserve the colon (00g) of the 64-character set, a trailing blank byte is either retained or appended as the last character in an even line. Examples of coded card conversion are shown in figure F-1.

Conversion Modes

Two conversion modes, 026 and 029,† exist for the Hollerith punch code. All data is converted in the system default keypunch mode unless a conversion mode change is specified. This change can be specified on any of the following cards.

The job card, 7/8/9 card (EOR mark), and 6/7/9 (EOF mark) can contain the keypunch conversion mode in columns 79 and 80. A 26 punched in columns 79 and 80 indicates that all subsequent cards are converted in 026 mode. A 29 indicates that subsequent cards are converted to 029 mode. Each conversion change remains in effect until another change card is encountered or the job ends. The user can switch between 026 and 029 mode as often as desired. If 26 or 29 does not appear in columns 79 and 80 of the job card, the initial keypunch mode of that job is the system default mode. If 26 or 29 does not appear on a 7/8/9 or 6/7/9 card, no conversion change is made, and the most recent keypunch mode remains in effect.

Keypunch mode can also be changed by a card containing a 5/7/9 punch in column 1. A blank (no punch) in column 2 indicates 026 conversion mode; a 9 punch in column 2 indicates 029 mode. The conversion change remains in effect until another change card is encountered or the job ends.

Literal Input

The 5/7/9 card also allows literal input when 4/5/6/7/8/9 is punched in column 2. Literal input allows 80-column binary data to be read while transmitting input in coded mode. Cards are read (16 central memory words per card) until a card identical to the previous 5/7/9 card (4/5/6/7/8/9 in column 2) is read. The next card can then specify the new conversion mode.

In order to maintain system integrity, an end-of-information card always terminates 80-column binary input (literal input). Either of the following is interpreted as an end-of-information card even though it appears in a literal input record.

- A card with 6/7/8/9 punched in column 1 and with columns 2 through 80 blank.
- A card with 6/7/8/9 punched in columns 1 and 80 and with columns 2 through 39 and columns 41 through 79 blank. Column 40 may be punched or left blank.

†These codes are ignored by a 200 User Terminal since conversion mode is selected by a hardware switch. (Refer to the Network Products Remote Batch Facility Reference Manual.)

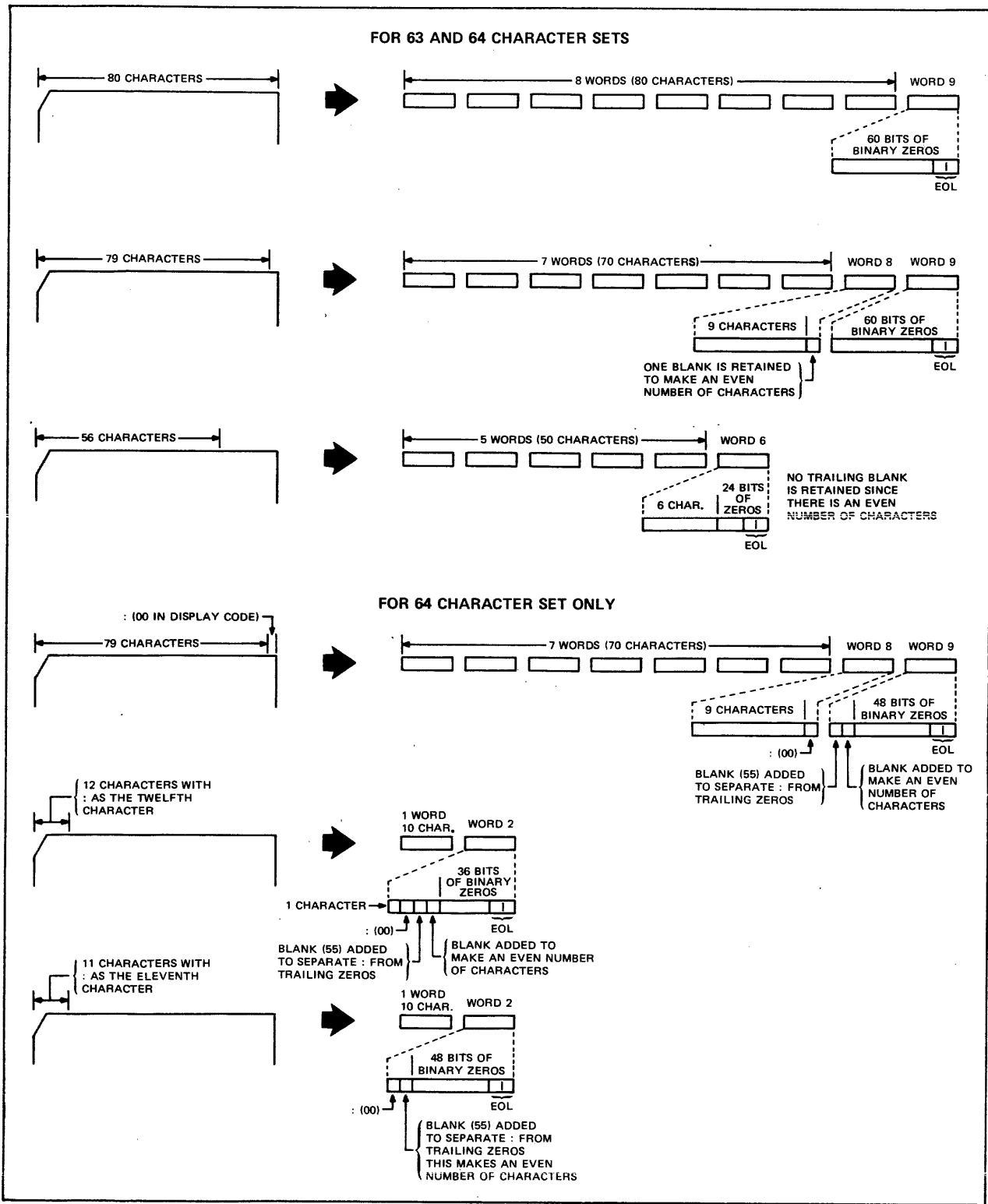


Figure F-1. Examples of Coded Card Conversion

BINARY CARDS

Binary cards are denoted by a 7/9 punch in column 1 and can contain up to 15 central memory words. The 3447 card reader controller reads the binary data and passes it to the card reader driver in 12-bit codes. Each card column row corresponds to a bit position. The driver checks the checksum figure if this option is specified. The driver then passes the data to the central memory buffer.

The fields within a binary card are:

<u>Column(s)</u>	<u>Description</u>
1	7/9 punch indicates a binary card.
	4 punch ignores checksum punch in column 2.
	Rows 0, 1, 2, and 3 contain the binary equivalent of the word count of the card.
2	Binary data checksum (modulo 4095).
3 through 77	Fifteen central memory words of binary data.
78	Blank.
79 and 80	24-bit binary card sequence number.

SUMMARY

The following punches appearing in column 1 of a card have the corresponding meaning to the card reader driver.

<u>Punch</u>	<u>Represents</u>
7/8/9	End-of-record (optional conversion mode change).
6/7/9	End-of-file (optional conversion mode change).
6/7/8/9	End-of-information.
5/7/9	Conversion mode change/read 80-column binary.
7/9	Binary card.
Not 7 and 9	Coded card.

PUNCH FILE FORMATS

Punched cards can be in three formats.

- Coded (Hollerith punch).
- Binary.
- Absolute binary.

The following conditions apply to all three formats.

- When an EOR is encountered, a card is punched with a 7/8/9 in columns 1 and 80. This card is offset.
- When an EOF is encountered for a file, a card is punched with a 6/7/9 in columns 1 and 80; the remainder of the card is blank. This card is offset.
- When an EOI is encountered on a file, a card is punched with a 6/7/8/9 in columns 1 and 80; the remainder of the card is blank. This card is offset.
- If a compare error is encountered, the erroneous card and the following card are offset. These two cards are repunched until no error is detected. An EOI card with 6/7/8/9 punches in columns 1 and 80 contains a binary count in column 40 of the number of compare errors.
- During the punching of each file, the system maintains a count of the number of cards punched for the file. If the number exceeds the limit for which the user is validated, punching of the file is terminated. A special banner card with the word LIMIT is punched and offset as the last card of the deck.

The following methods are used by the system to punch each of the three forms of cards.

CODED CARDS (PUNCH)

With the exception of decks punched via the ROUTE command, the keypunch mode (026 or 029) of coded cards depends on the job origin type. If the job is of local batch origin, decks are punched in the initial keypunch mode (that is, the mode specified on the job card or set by system default). For all other job origin types, decks are punched in the system default keypunch mode.

BINARY CARDS (PUNCH B)

The card punch driver retrieves 15 words of binary data from central memory. The driver then generates a checksum for the data and issues a card number. The card punch controller receives the binary data and punches it on the card unchanged, that is, in 12-bit codes. Each row in a card column corresponds to a bit position. The driver formats the binary card in the following manner.

<u>Column(s)</u>	<u>Contents</u>
1	7/9 punch denotes binary card. Rows 0, 1, 2, and 3 contain the binary equivalent of the word count of the card.
2	Binary data checksum (modulo 4095).
3 through 77	Fifteen central memory words of binary data.
78	Blank.
79 and 80	24-bit binary card sequence number.

ABSOLUTE BINARY CARDS (P8)

Absolute binary cards are central memory images in 12-bit codes. Each row in a card column corresponds to a bit position. Sixteen central memory words are punched per card with no special punches or fields added.

ANSI TAPE LABEL FORMATS

G

ANSI labels perform two functions. They provide information that uniquely identifies a file and the reel on which it resides, and they mark the BOI and EOI of a file and the beginning and end of a reel.

ANSI labels are designed to conform to the American National Standard Magnetic Tape Labels for Information Interchange X3.27-1969. All labels are 80 characters long and are recorded at the same density as the data on the tape. The first three characters of an ANSI label identify the label type. The fourth character indicates a number within a label type.

The following is a summary of each label type, name, function, and whether or not it is required.

<u>Type</u>	<u>No.</u>	<u>Name</u>	<u>Used As</u>	<u>Required/Optional</u>
VOL	1	Volume header label	Beginning-of-volume	Required
UVL	1-9	User volume label	Beginning-of-volume	Optional
HDR	1	File header label	Beginning-of-information	Required
HDR	2-9	File header label	Beginning-of-information	Optional
UHL		User header label	Beginning-of-information	Optional
EOF	1	End-of-file label	End-of-information	Required
EOF	2-9	End-of-file label	End-of-information	Optional
UTL	†	User trailer label	End-of-information	Optional
EOV	1	End-of-volume label	End-of-volume	Required when appropriate
EOV	2-9	End-of-volume label	End-of-volume	Optional

REQUIRED LABELS

The VOL1, HDR1, and EOF1 labels are required on all ANSI-labeled tapes. In addition, an EOV1 label is required if the physical end-of-tape reflector is encountered before an EOF1 label is written or if a multifile set is continued on another volume. In the descriptions of the contents of these labels, n is any numeric digit and a is any letter, digit, or any of the following special characters.

† Any member of the CDC 6-bit subset of the ASCII character set.

Space)	<
!	*	=
"	+	>
	,	?
\$	-	@
%	.	[
&	'	
/	:]
(;	

Some fields are optional. An optional field which does not contain the designated information must contain blanks. Fields which are not described as optional are required and written as specified. n-type fields are right-justified and zero-filled, and a-type fields are left-justified and blank-filled.

VOL 1—VOLUME HEADER LABEL

The volume header label must be the first label on a labeled tape. All reels begin with a VOL1 label. If two or more reels belong to a volume set, the file section field in the following HDR1 label gives the actual reel number.

VOL	1	volume serial number
va	reserved	
reserved		
reserved	owner identification	
owner identification (oid)		
oid	reserved	
reserved		
reserved	lsl	

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default</u>	<u>Checked on Read</u>	<u>Checked on Overwrite</u>
1-3	Label identifier	3	Must be VOL.		Yes	No
4	Label number	1	Must be 1.		Yes	No
5-10	Volume serial number	6	Volume identification assigned by owner to identify this physical reel of tape. If the volume serial number is all blanks, the tape is a scratch tape.	As read from existing label	Yes, if the file was assigned by volume serial number.	No
11	Accessibility (va)	1	An a character which indicates the restrictions, if any, on who may have access to the information on the tape. A blank means unlimited access. Any other character means special handling, in the manner agreed between the interchange parties. Refer to the BLANK command.	Blank (unlimited access)	No (refer to BLANK command.	Yes
12-31	Reserved for future standardization	20	Must be blanks.		No	No
32-37	Reserved for future standardization	6	Must be blanks.		No	No
38-51	Owner identification (oid)	14	Any a characters identifying the owner of the physical volume.	family name, user name	Refer to discussion of fa field of HDRI.	Yes
52-79	Reserved for future standardization	28	Must be blanks.		No	No
80	Label standard level (lsl)	1	A 1 means the labels and data formats on this volume conform to the requirements of the ANSI standard. A blank means the labels and data formats on this volume require the agreement of the interchange parties.	1	No	No

HDR1—FIRST FILE HEADER LABEL

The first file header label must appear before each file. When a file is continued on more than one volume, the file header label is repeated after the volume header label on each new volume for that file. If two or more files are grouped in a multifile set, each HDR1 label indicates the relative position of its associated file within the set.

HDR	1	file identifier (fi)		
file identifier (fi)				
fi	set identification		file section number (secno)	
secno	file sequence number	generation number		gvn
gvn	creation date		expiration date	
expiration date		fa	block count	
system code				
system code		reserved		

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default</u>	<u>Checked on Read</u>	<u>Checked on Overwrite</u>
1-3	Label identifier	3	Must be HDR.		Yes	No
4	Label number	1	Must be 1.		Yes	No
5-21	File identifier (fi)	17	Up to 17 a characters used as the file identification (fileid) parameter on the LABEL command.	Blank	Checked if specified.	No
22-27	Set identification	6	Up to six a characters used as the setid parameter on the LABEL command. To conform to the ANSI tape standard, this value is the same for all files of a multifile set.	Blank; an appended file is given the same set identification as its preceding file.	Checked if specified.	No
28-31	File section number (secno)	4	Four n characters identifying the file section number. The file section number of the first HDR1 label of a file is 0001. If the file extends to more than one volume, this number is incremented by one for each subsequent volume. This value corresponds to the secno parameter on the LABEL command.	0001	Checked if specified.	No
32-35	File sequence number	4	Four n characters used as the seqno parameter on the LABEL command. This parameter specifies the position of a file within a file set. This value is 0001 for the first file, 0002 for the second, and so on. In all the labels for a given file, this field contains the same number.	0001	Checked if specified.	No

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Checked on Default</u>	<u>Checked on Read</u>	<u>Overwrite</u>
36-39	Generation number (optional)	4	Four n characters specifying the generation number of a file. This is the genno parameter of the LABEL command. This value is 0001 for the first generation of a file, 0002 for the second, and so on.	0001	Checked if specified.	No
40-41	Generation version number (grn)	2	Two n characters used to distinguish successive iterations of the same generation. The generation version number of the first attempt to create a file is 00. This value corresponds to the gvn parameter of the LABEL command.	00	Yes	No
42-47	Creation date	6	Date the file was created; it is recorded as a space followed by two n characters for the year followed by three n characters for the day within the year. This value corresponds to the cdate parameter of the LABEL command.	Current date	Yes. The creation date is meaningful only on read operations; on write operations, the current date is always used.	No
48-53	Expiration date	6	The file is considered expired when today's date is the same as or later than the date given in this field. When this condition is satisfied, the remainder of the volume may be overwritten. Thus, to be effective on multifile volumes, the expiration date of a file must be earlier than or the same as the expiration date of	Current date	No	Yes

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default</u>	<u>Checked on Read</u>	<u>Checked on Overwrite</u>
			all preceding files on the volume. The expiration date is written in the same format as the creation date.			
			It corresponds to the rdate parameter of the LABEL command.			
54	Accessibility (fa)	1	An a character which indicates the restrictions, if any, on who may have access to the information in this file. A blank means unlimited access. If fa is A, only the owner of the NOS written tape can access the file. If fa is any other character, all future accesses to the tape must specify this character as the fa parameter.	Blank (unlimited access)	Yes, if a NOS written tape.	Yes
			File accessibility is not checked for system origin jobs.			
55-60	Block count	6	Must be zeros.		No	No
61-73	System code	13	Thirteen a characters identifying the operating system that recorded this file. The tape is considered to have been written under NOS if the first 10 characters match the default.	NOS ver-nn (ver is the system version number and nn is the EST ordinal of the unit on which the file was written).	No	No
74-80	Reserved for future standardization	7	Must be spaces.		No	No

EOF1—FIRST END-OF-FILE LABEL

The end-of-file label is the last block of every file. It is the system end-of-information for the file. A single tape mark precedes EOF1. A double tape mark written after the EOF1 label marks the end of a multifile set.

EOF	1	file identifier (fi)		
file identifier (fi)				
fi	set identification		file section number (secno)	
secno	file sequence number	generation number		gvn
gvn	creation date		expiration date	
expiration date		fa	block count	
system code				
system code		reserved		

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default</u>	<u>Checked on Read</u>
1-3	Label identifier	3	Must be EOF.		Yes
4	Label number	1	Must be 1.		Yes
5-54	Same as corresponding fields in HDR1 (optional)	50	Same as the corresponding fields in HDR1.		Same as HDR1.
55-60	Block count	6	Six n characters specifying the number of data blocks between this label and the preceding HDR label group. This total does not include labels or tape marks.		Yes
61-80	Same as corresponding fields in HDR1 (optional)	20	Same as corresponding fields in HDR1.		Same as HDR1.

EOV1—FIRST END-OF-VOLUME LABEL

The end-of-volume label is required only if the physical end-of-tape reflector is encountered before an EOF1 label is written or if a multifile set is continued on another volume. EOV1 is preceded by a single tape mark and followed by a double tape mark.

EOV	1	file identifier (fi)	
file identifier (fi)			
fi	set identification		file section number (secno)
secno	file sequence number	generation number	gvn
gvn	creation date		expiration date
expiration date	fa	block count	
system code			
system code		reserved	

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default</u>	<u>Checked on Read</u>
1-3	Label identifier	3	Must be EOF.		Yes
4	Label number	1	Must be 1.		Yes
5-54	Same as corresponding fields in HDR1 (optional)	50	Same as the corresponding fields in HDR1.		Same as HDR1.
55-60	Block count	6	Six n characters specifying the number of data blocks between this label and the preceding HDR label group. This total does not include labels or tape marks.		Yes
61-80	Same as corresponding fields in HDR1 (optional)	20	Same as corresponding fields in HDR1.		Same as HDR1.

OPTIONAL LABELS

Six types of optional labels are allowed. They are additional file header (HDR2-9), end-of-file (EOF2-9), user volume (UVLa), header (UHLa), and trailer (UTLa) labels.

HDR2 THROUGH HDR9—ADDITIONAL FILE HEADER LABELS †

HDR2 through HDR9 labels may immediately follow HDR1. Their format is:

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default Written</u>
1-3	Label identifier	3	HDR	HDR
4	Label number	1	2-9	2-9
5-80		76		

Only the label identifier and the label number are checked on read.

EOF2 THROUGH EOF9—ADDITIONAL END-OF-FILE LABELS †

EOF2 through EOF9 labels may immediately follow EOF1. Their format is:

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default Written</u>
1-3	Label identifier	3	EOF	EOF
4	Label number	1	2-9	2-9
5-80		76		

Only the label identifier and the label number are checked on read.

EOV2 THROUGH EOV9—ADDITIONAL END-OF-VOLUME LABELS †

EOV2 through EOV9 labels may immediately follow EOV1. Their format is:

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default Written</u>
1-3	Label identifier	3	EOV	EOV
4	Label number	1	2-9	2-9
5-80		76		

†These are reserved for operating system use. Attempts by user programs to generate them will produce undefined results.

Refer to section 3 in Volume 4, Program Interface, for a description of the use of EOVS labels in conjunction with CLOSER, REWIND, and UNLOAD macros.

USER LABELS

User labels may immediately follow their associated system labels. Thus, user volume labels (UVLa) may follow VOL1, user header labels (UHLA) may follow the last HDRn label, and user trailer labels (UTLa) may follow the last EOVS or EOFn label. Their format is:

<u>Character Position</u>	<u>Field Name</u>	<u>Length (in Characters)</u>	<u>Contents</u>	<u>Default Written</u>
1-3	Label identifier	3	UVL, UHL, or UTL.	UVL, UHL, or UTL.
4	Label number	1	Must be 1, 2, 3, 4, and so on, consecutively for UVL labels. For other labels, any a character.	
5-80	User option	76	Any a characters.	

Only the label identifier and the label number are checked on read. The system checks the number of user labels of a label type; a maximum of 64 is allowed.

CONTROL LANGUAGE (KCL)

H

NOTE

The control language KCL was the system control language available under NOS prior to the introduction of the commands described in sections 4 and 6. NOS will drop support of KCL at its next release. Convert your KCL procedures to the procedure format described in section 4. Unpredictable results may occur if you mix KCL commands with the commands described in sections 4 and 6.

The following paragraphs describe the various commands of KCL.

EXPRESSIONS

The expressions allowed are similar to FORTRAN expressions and may contain constants, operators, functions, and symbolic names.

OPERATORS

The arithmetic, relational, and logical operators are the same in the control language outlined in this appendix as they are in section 4. The only exception is the exclusive OR logical operator (.EOR.).

FUNCTIONS

Two functions are provided for use in expressions specified with control language statements. The FILE function determines the status of any file assigned to the job. The NUM function determines if a specified parameter name has a numeric value. For complete information concerning format and use, refer to Control Language Functions in this section.

SYMBOLIC NAMES

Symbolic names are used to reference values pertaining to the job process. There are three categories of symbolic names, as follows:

- Symbolic names with fixed arithmetic values:

ARE	Arithmetic error.
BCO	Local batch origin.
CMC	Maximum CM field length (MFL setting).
CMN	Nominal CM field length (RFL setting).
CPE	CPU abort.
ECM	Maximum ECS field length.
ECN	Nominal ECS field length.
EIO	Remote batch origin.
FLE	File limit error.
FSE	Forced error.
MNE	Monitor call error.
ODE	Operator drop.
OKE	Operator kill drop.
PPE	PPU abort.
PSE	Program stop error.
RRE	Rerun error.
SRE	SRU limit error.
SSE	Subsystem aborted.
SYO	System origin.
TKE	Track limit error.
TLE	Time limit error.
TXO	Interactive origin.

- Symbolic names with variable arithmetic values which depend upon job state:

EF	Previous error flag.
EM	Current exit mode.
FL	Job field length.
OT	Job origin type.
R1	Contents of control register 1.
R2	Contents of control register 2.
R3	Contents of control register 3.
SS	Job subsystem. This particular symbolic name requires an equal sign. SS may be equivalenced to one of the following:

ACCESS

BASIC

BATCH

EXECUTE

FORTRAN

FTNS

NULL

- Symbolic names with Boolean values:

F	False value.
FALSE	False value.
S _n	Setting (1 is on, 0 is off) of sense switch (1 ≤ n ≤ 6).
T	True value.
TRUE	True value.

EVALUATION OF EXPRESSIONS

The order of evaluation of expressions is:

1. Exponentiation
2. Multiplication, division

3. Addition, subtraction, negation
4. Relations
5. Complement
6. AND
7. Inclusive OR
8. Exclusive OR, equivalence

Nesting of expressions to any depth is allowed within a command.

CONTROL LANGUAGE COMMANDS

Control language commands are described in the following paragraphs. Separators and terminators must be used as shown in the command formats.

GOTO COMMAND

The GOTO command transfers control to another location within the command file.

The command format is:

GOTO,cmd

cmd Name of any command, or a digit (0 through 9) followed by a maximum of six alphanumeric characters, terminated by a period.

<u>Example 1</u>	<u>Example 2</u>
.	.
GOTO,1WX2.	REQUEST(TAPE1)
.	.
.	.
.	.
.	GOTO,REQUEST.
1WX2,REQUEST(TAPE1)	.
.	REQUEST(TAPE2)
.	.

Each time the system processes a GOTO command, it issues the following message:

WARNING - CONVERT PROCEDURE TO CCL

NOS will drop its support of KCL at its next release.

When cmd appears more than once in the command file, the cmd to be executed is the first occurrence of cmd from the beginning of the command file. Hence, in both of the previous examples, the REQUEST (TAPE1) command is processed after the GOTO command.

CALL COMMAND

The CALL command allows you to insert a file consisting of a group of command (procedure file) at the specified position in the command stream. This file is merged, as specified on the CALL command, with the current command record into a third record. This third record becomes the current command record. The remainder of the input file is then copied to the new command record. If the C option is exercised, the current command record is not used. Only the source file is used to generate a new command record. The C and S options are order independent; the RENAME option, if present, must be last.

Lines within a procedure file may contain line numbers to make maintenance easier. Usually, the CALL command strips off these line numbers before copying the procedure to the new command record. However, if a comma immediately follows the line number, the line number remains on the command.

The command format is:

```
CALL(lfn,C,S=ccc,RENAME(oldnam1=newnam1,oldnam2=newnam2,...,oldnamn=newnamn))
```

or

```
CALL(lfn,C,S=ccc(oldnam1=newnam1,oldnam2=newnam2,...,oldnamn=newnamn))
```

lfn	Procedure file name (refer to the description of procedure files in this section for further information). The system obtains lfn by: <ul style="list-style-type: none">● Searching for a local file, lfn.● Searching the system library for lfn.● Attempting to retrieve a working copy of an indirect access file.
C	Replaces all of the command record after the CALL command with lfn.
S=ccc	Sets next command to be processed to command ccc. If S is not specified, the first command in lfn is processed.
RENAME	Each occurrence of oldnam _i is replaced with newnam _i before the command is entered into the command file. As shown by the optional format, the word RENAME does not have to appear.
oldnam _i	Old name; name of a file or command label used in the specified procedure file.
newnam _i	New name; name to replace oldnam _i .

Each time the system processes a CALL command, it issues the following message:

WARNING - CONVERT PROCEDURE TO CCL

NOS will drop support of KCL at its next release.

If lfn is not properly formatted for a procedure file, the following message is issued.

lfn NOT A PROCEDURE FILE.

DISPLAY COMMAND

The DISPLAY command evaluates an expression and sends the result to the job dayfile in both decimal and octal integer form. The largest decimal value which can be displayed is 10 digits. If the value is larger than 10 digits, GT followed by 999999999 is displayed. If the value is negative and larger than 10 digits, LT followed by a minus and 999999999 is displayed. In octal code, numbers as large as $2^{48}-1$, zeros are displayed.

The format of the DISPLAY command is:

DISPLAY(exp)

exp A KCL expression.

Example:

The following sample dayfile shows several display operations.

```
15.14.59.DISPLAY(TIME)
15.14.59.      1514    2752B
15.15.07.SET(R1=99)
15.15.21 SET(R2=901)
15.15.28.DISPLAY(R1)
15.15.28.      99    143B
15.15.38.DISPLAY(R1+R2)
15.15.38.      1000    1750B
15.15.47.DISPLAY(3/2)
15.15.47      1    1B
15.16.04.DISPLAY(2**47)
15.16.04. GT 999999999    4000000000000000B
15.16.15.DISPLAY(-2**47)
15.16.15. LT -999999999    -4000000000000000B
15.16.27.DISPLAY(2**48)
15.16.28      0    0B
```

The first DISPLAY command displays the value of the TIME symbolic name. The current time given is in the form hhmm. The next six lines demonstrate the use of the R1 and R2 symbolic names. The other DISPLAY command specify numeric expressions.

SET COMMAND

The SET command assigns a value to a control register, an error flag, or the flag that determines whether skipped commands are entered in the dayfile. Using the SS function, it also can change the current interactive subsystem.

To assign a value to a symbolic name, the following format is used.

SET(sym=exp)

sym One of the following symbolic names (initially these names are set to 0).

<u>Name</u>	<u>Description</u>
R1, R2, or R3	Local control registers. When a procedure is called, the current values of R1, R2, and R3 are passed to the procedure. The values of these registers may change within the procedure; however, when processing reverts, these registers are restored to the values they had when the procedure was called.
EF	Local error flag. When a procedure is called, the current value of the error flag is passed to the procedure. The value of the error flag may change within the procedure; however, when processing reverts, the error flag is restored to the value it had when the procedure was called.

exp A KCL expression. The value derived through evaluation of the expression is assigned to the symbolic name. Acceptable values for each symbolic name follow.

<u>sym</u>	<u>Suggested Value</u>
R1, R2, or R3	Any integer between -131 071 and 131 071. If the value is outside this range, it is truncated. KCL does not issue a message as a result of the truncation.
EF	Any integer between 0 and 63. If the value is greater than 63, it is truncated. To assign the value defined by the system for an error condition, the user should set the error flag to one of the error condition symbolic names (refer to Symbolic Names at the beginning of this section). KCL sets the EF flag to the appropriate error code when an error occurs.

To change the current interactive subsystem, the following format is used.

SET(SS=subsystem)

subsystem Subsystem name. The subsystem names are ACCESS, BASIC, BATCH, EXECUTE, FORTRAN, FTNTS, and NULL.

Example:

This example illustrates the use of the SET command to control execution of an object program. Because register R1 is set to 1 when the file ABC is called, TAPE 1 is not requested and the object program is not executed.

```
SET(R1=1)
CALL(ABC)
FTNS.
IF(R1=1) GOTO,3.
REQUEST(TAPE1)
LGO.
3,REWIND(TAPE1)
.
.
.
```

IF COMMAND

The IF command is used to evaluate an expression. If the conditions given in the expression are true, the dependent command is processed. The expression is considered true if it is evaluated to a nonzero numeric value.

The command format is:

IF(expression)cmd.

or

IF(SS=ssname)cmd.

expression	Any legal expression.†
cmd	Any legal command.
ssname	Any legal SS subsystem name.

NOTE

A command of the form IF(expression)CALL (lfn) is not recommended. Each time the IF command is processed and the expression is true, the CALL command is processed. This merges the called lfn with the current command stream and creates a copy of this procedure file each time.

†If a permanent file command is included in an IF command, a password (if present) is not deleted in the dayfile.

Example 1:

```
IF(R2=R1.AND.R3)GOTO,REQUEST.  
SET(EF=1)  
:  
:  
:  
REQUEST(TAPE)
```

If the expression is true, the REQUEST command is executed; otherwise, the SET command is executed.

Example 2:

```
IF(SS=BASIC)GOTO,100.  
SET(SS=BASIC)  
:  
:  
:  
100,OLD,BAS.
```

If the command expression is true, the OLD command is processed; otherwise, the SET command is processed.

Each time the system processes an IF command, it issues the following message:

```
WARNING - CONVERT PROCEDURE TO CCL
```

NOS will drop support of KCL at its next release.

CONTROL LANGUAGE FUNCTIONS

Control language functions are described in the following paragraphs. Separators and terminators must be used as shown in the function formats.

FILE FUNCTION

The FILE function is used to determine the status of any file assigned to the job and is used in conjunction with the SET, IF, and DISPLAY control language commands.

The format of the function is:

```
FILE(ifn,expression)
```

ifn File name.

expression Any legal expression; however, FILE expressions cannot include functions. In addition, FILE expressions use different symbolic names, as follows:

Symbolic names:

Names with values:

EQ Equipment status table (EST) ordinal † (0 through 778).

† Contact installation personnel for a list of EST ordinals.

Symbolic names:

File characteristics:

MS File is on mass storage.
LK File is locked.
OP File is opened.
EX Execute-only file.
AS File is assigned to user's control point.

File types:

LO Local.
IN Input.
QF Queued output.
LI Library.
PM Direct access permanent file.
PT Primary.

Device types:

DE Extended core storage.
DI 844-21 Disk Storage Subsystem (half track).
DJ 844-41 or 844-44 Disk Storage Subsystem (half track).
DK 844-21 Disk Storage Subsystem (full track).
DL 844-41 or 844-44 Disk Storage Subsystem (full track).
DM 885 Disk Storage Subsystem (half track).
DP Distributive data path to Extended Memory.
DQ 885 Disk Storage Subsystem (full track).

MS	Mass storage.
MT	Magnetic tape drive (seven-track).
NE	Null equipment.
NP	Host communications processor.
NT	Magnetic tape drive (nine-track).
TT	Interactive terminals.

Examples:

```
SET(R1=FILE(TAPE,MT))
```

If TAPE is a file on a seven-track magnetic tape drive, R1 is set to 1; otherwise, it is set to 0.

```
IF(FILE(BETA,DI.AND.PM))GOTO,200.
```

If BETA is a file on an 844-21 Disk Storage Subsystem and it is a direct access permanent file, control skips to the command at 200.

NUM FUNCTION

The NUM function is used to determine if the specified parameter name has a numeric value. It is used in conjunction with the SET, IF, and DISPLAY control language command.

The format of the function is:

```
NUM(name)
```

name	Parameter name. If the name is numeric, the statement is true; otherwise, it is false.
------	--

Example:

If the CALL command

```
CALL(A,RENAME(2XY=2,T=TAPE))
```

is used to call procedure file A, the IF command in A

```
IF(NUM(2XY))GOTO,1S.
```

is evaluated as true, and control transfers to 1S.

However, the command

```
IF(NUM(T))GOTO,1S.
```

is evaluated as false, and control passes to the next command in A.

PROCEDURE FILES

Procedure files are source files consisting of commands, control language commands, or both. The first statement of a procedure file may be the file name. If the first statement is the same as the file name used in the CALL command, the first statement is ignored. Procedure files are activated by the CALL command or by using the name of the procedure file, if the file is in the system.

Example 1:

The procedure file in this example is an indirect access file called COMPARE. This routine copies an input file and compares it with an existing direct access file. In the procedure file, these two files are called DUPL and MASTER. When the procedure file is inserted into the command record during job processing, the name of DUPL is changed to NEWFILE.

Original Input File

```
JOBAAA.  
USER(EFD2501,PASS)  
CHARGE,*.  
CALL(COMPARE(DUPL=NEWFILE))  
-EOR-
```

```
      .  
      input file  
      that is to  
      be compared  
      .
```

```
-EOI-
```

Procedure File COMPARE

```
COMPARE  
COPYBR(,DUPL)  
ATTACH(MASTER)  
VFYLIB(MASTER,DUPL)
```

After the CALL command is processed, the command record is as follows:

```
JOBAAA.  
USER(EFD2501,PASS)  
CHARGE,*.  
CALL(COMPARE(DUPL=NEWFILE))  
COPYBR(,NEWFILE)  
ATTACH(MASTER)  
VFYLIB(MASTER,NEWFILE)  
-EOR-
```

Example 2:

This is an example of nested calls. It illustrates the use of one procedure file to skip a specified number of files on a tape (contents of R1) and to copy source data to the tape. The other procedure file retrieves source data from the OPL (old program library) and calls the first procedure file to place that source data on the tape.

Input Deck

```
JOBAAA.  
USER(USERNAM,PASSWRD)  
CHARGE,*.  
ATTACH(OPL/UN=LIBRARY)  
REQUEST(TAPE)  
MODIFY(S,Z)/*EDIT,CPM  
SET(R1=0)  
CALL(PROC,RENAME(A=TAPE,B=SOURCE,2=2A,3=3A)  
SET(R1=R1+1)  
CALL(PROB)  
-EOR-
```

Procedure File PROB

```
PROB  
MODIFY(S=NEW,Z)/*EDIT,MTR  
CALL(PROC,RENAME(A=TAPE,B=NEW)  
RETURN,NEW.
```

Procedure File PROC

```
PROC  
REWIND(A,B)  
SET(R2=0)  
2,IF(R1=R2)GOTO,3.  
SKIPF(A)  
SET(R2=R2+1)  
GOTO,2.  
3,COPYBF,B,A.
```

NOTE

On job initiation, your input file is a locked file. If you wish to call procedure files that write data on the input file, you should enter the RETURN(INPUT) command before attempting to write on INPUT. For further information, refer to Input File Control in section 3.

LINE PRINTER CARRIAGE CONTROL

This appendix briefly describes the format and processing of print files.† It lists the carriage control for the programmable format control (PFC) and non-PFC 580 line printers.

PRINTED DATA

All data to be printed is in coded format in a print file within the print queue. The data consists of either 6-bit or 12-bit codes. Data recorded using the 6/12 display code set (refer to appendix A) should be converted to the 12-bit ASCII code set (refer to the FCOPY command in section 9) before being routed to a line printer.

The system extracts data until an end-of-line occurs or until 137 characters are retrieved. End-of-line is 12 or more zero bits in the rightmost byte of a central memory word.

CARRIAGE CONTROL

The system interprets the first character in a line as the carriage control character†† and that character is not printed (table I-1). The remainder of the line is then printed, except when the Q, R, S, or T carriage control characters are specified. The Q, R, S, T, and V format controls remain in effect until changed; all other carriage control characters must be supplied for each line they control. Line spacing is normally done in auto eject mode; that is, creases in the paper are skipped by the line printer's automatic line spacing mechanism if the paper is loaded properly. Auto eject mode must be turned off if you want to select format channels to advance printing from a position above the bottom of form to a position beyond the next top of form.

During the printing of each file, the system maintains a count of the number of lines printed or skipped for the file. If the number exceeds the limit for which you are validated, printing of the file is terminated. The informative diagnostic LINE LIMIT EXCEEDED is printed. If a job's dayfile is part of the terminated print file, the dayfile is subsequently printed.

The installation can impose an implied page control by setting a certain number of default lines for each page. If less than the default number of lines is printed or skipped on a page, the line limit is still decremented by the default number of lines.

†To print a file in which the first character of each line is not a carriage control character, refer to the COPYSEF command in section 9.

††The information in this appendix does not apply to remote batch line printers.

Table I-1. Carriage Control Characters

Character	Action
SPACE	Single space.
1	Eject page before print.
0	Skip one line before print (double space).
-	Skip two lines before print (triple space).
+	Suppress space before print.
/	Suppress space after print.
2	Skip to last line of form before print.
Q	Clear auto eject; † remainder of line is not printed.
R	Set auto eject; remainder of line is not printed.
S	Select 6 lines/inch; remainder of line is not printed.
T	Select 8 lines/inch; remainder of line is not printed.
V	Eject page before print on a 580 PFC printer, V loads a user-supplied PFC array (validated users only).
8	Skip to next punch in format channel 1 before print. ††
7	Skip to next punch in format channel 2 before print. ††
6	Skip to next punch in format channel 3 before print. ††
5	Skip to next punch in format channel 4 before print. ††
4	Skip to next punch in format channel 5 before print. ††
3	Skip to next punch in format channel 6 before print. ††
H	Skip to next punch in format channel 1 after print.
G	Skip to next punch in format channel 2 after print.
F	Skip to next punch in format channel 3 after print.
E	Skip to next punch in format channel 4 after print.
D	Skip to next punch in format channel 5 after print.
C	Skip to next punch in format channel 6 after print.
†The deselection of auto eject mode on a 580 line printer results in the deselection of 8 lines per inch, if previously selected. ††No space after print. For all other control characters, a line feed is issued after print.	

CARRIAGE CONTROL USING FORMAT CHANNEL SELECTION

You can use carriage control characters (table I-1) that refer to channels (tracks) on a punched carriage control tape or in a programmable format control (PFC) array. The carriage control character beginning a print file line determines where on the printer form the line is printed.

After reading the carriage control character for a print line, the printer checks the format channel the character references. If a punch exists at that frame in that channel of the carriage tape or if a bit is set in that frame and channel of the PFC array, the printer prints the line. If not, the printer advances the carriage tape or PFC array and advances the printer paper until a punch or set bit is found in that channel. It then prints the line.

As listed in table I-1, some carriage control characters name an action to be taken while others directly name a format channel. When specifying characters that name an action, you indirectly name the format channel that performs the named action. To use the characters that directly name a format channel, you need to know the contents of the channel.

The format of the carriage control tape recommended for use on a 580 non-PFC line printer is listed in table I-1. Lines 132 through 134 are identical to lines 0 through 2 because they overlap when the punched tape is glued together to form a continuous loop. Therefore, you can disregard lines 132 through 134.

Selecting format channels on the carriage tape illustrated in figure I-1 produces the following actions.

<u>Carriage Control Character</u>	<u>Format Channel Selected</u>	<u>The line printer advances to:</u>
8 or H	1	Line 0 or 66.
7 or G	2	First line of the next two-line group.
6 or F	3	First line of the next three-line group.
5 or E	4	First line of the next four-line group.
4 or D	5	First line of the next five-line group.
3 or C	6	Line 0.

If the numeric carriage control character is specified, the printer advances before printing. If the alphabetic character is specified, the printer advances after printing.

Frame	Channels											
	1	2	3	4	5	6	7	8	9	10	11	12
0	x	x	x	x	x	x	x	x	x	x	x	x
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41												
42												
43												
44												

Frame	Channels											
	1	2	3	4	5	6	7	8	9	10	11	12
45												
46												
47												
48												
49												
50												
51												
52												
53												
54												
55												
56												
57												
58												
59												
60												
61												
62												
63												
64												
65												
66												
67												
68												
69												
70												
71												
72												
73												
74												
75												
76												
77												
78												
79												
80												
81												
82												
83												
84												
85												
86												
87												
88												
89												

Frame	Channels											
	1	2	3	4	5	6	7	8	9	10	11	12
90												
91												
92												
93												
94												
95												
96												
97												
98												
99												
100												
101												
102												
103												
104												
105												
106												
107												
108												
109												
110												
111												
112												
113												
114												
115												
116												
117												
118												
119												
120												
121												
122												
123												
124												
125												
126												
127												
128												
129												
130												
131												
132												
133												
134												

Figure I-1. Carriage Control Tape Format

CARRIAGE CONTROL ARRAYS FOR 580 PFC PRINTERS

Line spacing on a 580 programmable format control (PFC) line printer is controlled by a PFC array that acts as a software version of the carriage control tape in non-PFC printers. The printer has four released PFC arrays, two for six-line-per-inch print density and two for eight-line-per-inch print density. You can specify one of the released arrays with the SC parameter on the ROUTE command. The released arrays are listed in the NOS 2 System Maintenance Reference Manual. If validated (refer to LIMITS command, section 7), you can include a PFC array in the print file to control its spacing until another PFC array or the end of the print file is reached. This user-supplied PFC array begins with the carriage control character V.

NOTE

The PFC array does not change the print density. Print density is selected by an S or T carriage control character.

Upon reading a V carriage control character, the line printer page ejects. If the printer is not a PFC printer, the rest of the line is ignored. If you are not validated to use the V carriage control character, the print file terminates, and you are informed by a message in your output file.

NOTE

The V carriage control character is ineffective when a print file is routed from an interactive job.

PFC ARRAY SYNTAX

Carriage control character V is in column 1 of the line. The character in column 2 determines which PFC array is changed. Its legal values are the following:

<u>Value</u>	<u>Description</u>
6	Six-line-per-inch spacing. The entire array specification is on this line.
8	Eight-line-per-inch spacing. The entire array specification is on this line.
C	Eight-line-per-inch spacing. The array specification is continued on a second line. The second line begins in column 3. Columns 1 and 2 are ignored.

The PFC array specifications start in column 3. They are alphabetic characters A through L, O, and blank and have the following significance.

<u>Character</u>	<u>Significance</u>
A	Specifies top of forms code. This must be the first character in the array.
B through K	Specifies format channels 2 through 11,† respectively.
L	Specifies bottom of forms code.
O	Signals the end of the array specification. This must be the last character in the array. It has no effect on printing.
Blank	No channel specified.

NOTE

You must specify each channel referenced in the print file in the PFC array. A channel specification can be repeated.

A six-line-per-inch array specification may be a maximum of 132 characters plus the array terminator. An eight-line-per-inch array specification may be 176 characters plus the array terminator.

If the array contains an illegal character, the array line is printed and the print file is terminated. Other invalid arrays are ignored and the file is printed using the carriage control array previously loaded into the printer.

Examples:

The following arrays are invalid.

<u>Invalid Array</u>	<u>Reason</u>
V6BCD O	Does not begin with an A.
VBA C DEO	Second character is not 6, 8, or C.
V8ABWCO	Contains an illegal character (W).

The following example uses a PFC array to print a short special form.

Suppose you want to print on the top, fifth, and bottom lines of an eight-line form (eight-line-per-inch print density). Select the print density with the T carriage control character and then load the following PFC array.

```
Columns:  1  2  3  4  5  6  7  8  9 10 11
          V  8  A          B      L  O
```

†The A character refers to channel 1, and L refers to channel 12. In the maximum length array specification (132 for six lines per inch and 176 for eight lines per inch), the J character can be specified only in the last position (immediately before the O).

The eight-line-per-inch carriage control array is changed as follows (x denotes a bit set):

	Channels											
	1	2	3	4	5	6	7	8	9	10	11	12
1	x											
2												
3												
4												
5			x									
6												
7												
8											x	

Because only 3 of the 12 format channels are specified in the array, the other 9 channels (in the eight-line-per-inch array) remain unchanged from their last setting.

After loading the array, output lines beginning with an 8 (format channel 1) are printed at the top of form, lines beginning with a 7 (format channel 2) are printed at the fifth line, and lines beginning with a 2 (format channel 12) are printed at the bottom of the form.

OBSOLETE TAPE FORMATS

J

The B, E, and X format coded tapes are no longer supported by NOS. The following description of the physical and logical characteristics of the obsolete tapes is included as a reference for you if you use the TCOPY command.

B (BLOCKED) FORMAT

<u>Characteristics</u>	<u>Description</u>
Header	Unlabeled.
Mode	Coded.
Block size (PRU size)	The block size cannot exceed 5120 frames. If the tape unit will not allow an odd number of frames to be written, the system will append a space. Unless you specify otherwise when you request a tape, the system will assume the maximum block size is 150 frames.
Logical end-of-record	For a write operation, there is no logical end-of-record. For a read operation, end-of-record status is returned when a tape mark is encountered. An additional read operation returns end-of-file status.
Logical end-of-file	Tape mark.
Logical end-of-information	None.
End-of-reel	Refer to option 2 under End-Of-Tape/End-Of-Reel Conditions.
Noise	Any block containing fewer than 18 frames is considered noise, and is, therefore, ignored.
Special considerations	<ul style="list-style-type: none">● B-formatted tapes cannot be labeled.● A write operation will stop either at a zero byte (end-of-line) in byte 4 of a CM word or at a multiple of CM words (rounded up) based on the frame or character count.● For control word reads, byte count and unused bit count will be set appropriately. For regular reads, EOL is guaranteed.● For a control word write operation, no end-of-line processing is done. Data is blocked on tape using the specified frame count. Likewise for a control word read operation, no end-of-line processing is done; data is transferred to you as it is read.

E (LINE IMAGE) FORMAT

<u>Characteristics</u>	<u>Description</u>
Header	Unlabeled.
Mode	Coded.
Block size (PRU size)	The block size cannot exceed 5120 frames. If the tape unit will not allow an odd number of frames to be written, the system will append a space. Unless you specify otherwise when you request the tape, the system assumes the maximum block size is 136 frames.
Logical end-of-record	For a write operation, there is no logical end-of-record. For a read operation, end-of-record status is returned when a tape mark is encountered. An additional read operation returns end-of-file status.
Logical end-of-file	Tape mark.
Logical end-of-information	None.
End-of-reel	Refer to option 2 under End-Of-Tape/End-Of-Reel Conditions.
Noise	Same as for B-formatted tapes.
Special considerations	<ul style="list-style-type: none">● E-formatted tapes cannot be labeled.● For a write operation, a block of data will stop either at a zero byte (end-of-line) in byte 4 of a CM word or at the multiple of CM words (rounded up) based on the frame or character count. The system will then space-fill the buffer to the number of frames specified. Thus, the amount of data written will exactly equal the amount specified.● For a read operation, if there is an odd number of characters, the system will space-fill the last 6 bits of the last byte and delete all trailing spaces. For control word reads, byte count and unused bit count will be set appropriately. For regular reads, EOL is guaranteed.● For a control word write operation, no end-of-line processing is done. Data is blocked on tape using the specified frame count. Likewise for a control word read operation, no end-of-line processing is done; data is transferred to you as it is read.

X (EXTERNAL) FORMAT

<u>Characteristics</u>	<u>Description</u>
Header	Unlabeled.
Mode	Binary.
Block size (PRU size)	Actual data block size can range from 0 to 512 (1000g) CM words in exact multiples of CM words.
Logical end-of-record	Any block containing fewer than 512 CM words represents a logical end-of-record. If a logical record consists of an exact multiple of 512 words, the block that denotes the logical end-of-record consists solely of a 48-bit block terminator.
Logical end-of-file	Tape mark.
Logical end-of-information	None.
End-of-reel	Refer to option 1 under End-Of-Tape/End-Of-Reel Conditions.
Noise	Any block containing fewer than eight frames for seven-track tapes or six frames for nine-track tapes is considered noise, and is, therefore, ignored.
Special considerations	<ul style="list-style-type: none">● X-formatted tapes cannot be labeled.● All nine-track tapes are written in an even multiple of bytes.

END-OF-TAPE/END-OF-REEL CONDITIONS

The following is a description of the processing options for end-of-tape conditions. You can select one of these options by default by specifying the data format or you can specify an option via the PO keyword on a LABEL, ASSIGN, or REQUEST command.

<u>Option</u>	<u>PO= Option</u>	<u>Description</u>
1	I	Rewrites the block on which the end-of-tape condition is sensed as the first block on the next volume, if the system senses the EOT during a write operation. During a read operation, the block on which the EOT occurred is ignored and reading continues on the next volume. If a tape mark and the EOT are sensed at the same time, the EOT is ignored.

<u>Option</u>	<u>PO= Option</u>	<u>Description</u>
2	P	Writes a trailer sequence following the block on which the EOT is sensed, if the system senses the EOT during a write operation. Any data that occurs following the block on which EOT is sensed, yet before the tape mark, is ignored. During a read operation, the system transfers the block on which the EOT is sensed to your job. The read operation resumes on the next reel. If a tape mark and the EOT are sensed at the same time, the EOT is ignored.

The system is concerned only with the block on which the EOT is sensed. If tapes written using these options are transferred to another system, any data that occurs on the reel after this block should be ignored.

INTRODUCTION

The network, which IAF uses as a terminal interface, can support many types of terminals. All supported terminals are grouped by the network into 15 terminal classes. Each terminal class has specific operating characteristics. These are defined by a set of parameters referred to as terminal definitions. These terminal definitions are predefined by the network to match closely the operating characteristics of actual terminals. Table K-1 lists these terminal definitions, the default values, and possible entries (in parentheses) for each of the 15 classes. To use the terminal definition defaults, your terminal should correspond to the terminal class (TC) in use. If you do not know the terminal class in use, you can enter the TC command specifying the terminal class.

When a terminal logs in to the network, either it is assumed by the network to be of a certain terminal class, or it is assigned a terminal class by the network that resembles its actual characteristics. In either case, if the characteristics of the terminal do not match those of its assigned terminal class, you can change the values of the terminal definitions or even change the terminal class, using the terminal definition commands described in this appendix or the TRMDEF command described in section 8.† Terminal definition commands can be entered any time after the terminal is connected to the system (refer to login/logout procedures in Volume 1, Introduction to Interactive Usage), including during IAF operations. The TRMDEF command can be entered only when IAF is being used.

When the TRMDEF command or a terminal definition command is used to change a value, that change remains in effect until the terminal is disconnected from the network or another TRMDEF or terminal definition command is used to change the value. Even application switching and IAF logout do not change the values of the terminal definitions if the terminal has not been disconnected from the network. Only by disconnecting the phone and redialing (on dial-up terminals) or by entering the TC terminal definition (described in this section) can the values be reset to their default values.

The TRMDEF command is described under Terminal Control Commands in section 8. All of the terminal definition parameters in table K-1 can be used with the TRMDEF command unless stated otherwise.

TERMINAL DEFINITION COMMAND FORMAT

The terminal class or terminal definition values can be changed by entering terminal definition commands. These commands should be entered in the following format.††

control terminal definition Ⓞ

<u>Parameter</u>	<u>Description</u>
control	Terminal control character defined for the terminal in use (refer to CT in table K-1).

†It is also possible to change terminal definitions using control byte 16. Refer to Volume 4, Program Interface for information on control bytes.

††Spaces are included for clarity. They should not be included when entering the command.

<u>Parameter</u>	<u>Description</u>
terminal definition	A two-character mnemonic shown in table K-1, followed by = and the desired terminal class number or parameter value. The range of values is shown in table K-1.
Ⓞ	Message terminator character for the terminal in use.

As an example, to set the cancel character on a 713 terminal to a ?, press the ESCAPE key and enter:

CN=? Ⓞ

On terminals from which it is possible to input multiple logical lines in a single transmission block (more than one line can be entered before transmitting), the terminal definition command must be the first and only input line in the transmission block. If the input is from paper tape, the tape reader must stop. Unpredictable results may occur if more input follows the terminal definition command in the transmission block.

If the user makes a format error while entering a terminal definition command, or enters a command which is invalid for the terminal, the message

xxERR..

is printed at the terminal. xx is the terminal definition mnemonic in error.

In addition to the terminal definition commands, a message command (MS) and a character code set command (CD), which have the same format, are available to you. These commands are described at the end of this appendix. MS and CD cannot be used as parameters in the TRMDEF command.

TERMINAL DEFINITION PARAMETERS

The following descriptions explain the terminal definition parameters used in the terminal definition commands and the TRMDEF command. Default values are not shown since they vary with the terminal class. Refer to table K-1 for the default values.

TERMINAL CLASS (TC)

The TC parameter specifies a value from 1 to 15 that associates the terminal type with a set of terminal characteristics parameters. Any terminal being used in the network belongs to one of the following 15 terminal classes.

<u>TC</u>	<u>Terminal Type</u>
1	M33, M35, M37, M38 teletypewriters
2	CDC 713-10
3	Reserved for future use
4	IBM 2741

<u>TC</u>	<u>Terminal Type</u>
5	M40
6	Hazeltine 2000
7	CDC 751-1, 752, 756
8	Tektronix 4014
9	HASP Protocol
10	200 User Terminal
11	CDC 714-30
12	CDC 711-10
13	CDC 714-10/20
14	CDC 731-12, 732-12
15	CDC 734

A terminal that is not shown as belonging to a terminal class may still be operational. It can be assigned to a terminal class having similar characteristics and whose parameters have been changed as necessary by site personnel to define correctly the operational characteristics of the terminal. The terminal class which supports the new terminal type has operating parameter values different from the default values shown in table K-1.

Terminals connected to the network through autorecognition lines can be assigned to terminal class 1, 4, 10, or 13 (unless site personnel have configured the network otherwise). When the terminal being used on an autorecognition line is other than terminal class 1, 4, 10, or 13, the TC parameter must be changed to correctly identify the terminal class.

CONTROL CHARACTERS (CT)

The CT parameter establishes the character to be used to identify terminal definition commands. When this character is entered as the first character in a logical line (or transmission block), the data following is assumed to be a terminal definition command. Any character in the ASCII 128-character set can be selected to function as the control character.† This character must differ from the characters currently defined for the BS, CN, AL, B1, and B2 parameters. The character specified by the CT parameter is the character represented by control in the terminal definition command format.

BACKSPACE CHARACTERS (BS)

The BS parameter specifies the character to be used to delete the previous input character. Any character in the ASCII 128-character set can be selected to function as the backspace character. This character must differ from the characters currently defined for the CT, CN, AL, B1, and B2 parameters. It is possible to backspace only to the beginning of the current physical line; additional backspaces are disregarded. When a page width of zero is selected, the characters are sent to IAF in multiples of 150 characters.

†Refer to table K-1 for the exceptions.

CANCEL CHARACTERS (CN)

The CN parameter specifies the character to be used to cancel the logical line currently being input. Any character in the ASCII 128-character set can be selected to function as the cancel character. This character can be the same as the AL (abort output line) character but must differ from the characters currently defined for the CT, BS, B1, and B2 parameters. When the cancel character is entered as the last character before Ⓜ is pressed, the entire logical line in progress is cancelled. If part of the current logical line has already been transmitted to the application, a flag is set to inform the application that the cancel character has been entered. The system responds to a cancel line character by printing *DEL* on the next line and positioning the carriage to the beginning of a new line.

ABORT OUTPUT LINE CHARACTER (AL)†

The AL parameter specifies the character to be used to abort an output logical line. Any character in the ASCII 128-character set can be specified as the abort output line character.†† This character can be the same as the CN (cancel) character but must differ from the characters currently defined for the CT, BS, B1, and B2 parameters. The current output line is discarded when the abort line character is entered as the only character in a logical line (entering the abort output line character followed by Ⓜ) while output is in progress.

INTERRUPTION CHARACTER (B1)

The B1 parameter specifies the character that, when entered as the only character in a logical line (interruption character followed by Ⓜ), causes program interruption. This process is discussed in section 17. The interruption sequence is also called the user break 1 sequence. Any character in the ASCII 128-character set can be specified as the interruption character.†† This character must differ from the characters currently defined for the CT, BS, CN, AL, and B2 parameters.

TERMINATION CHARACTER (B2)

The B2 parameter specifies the character that, when entered as the only character in a logical line (termination character followed by Ⓜ), causes program termination. This process is discussed in section 17. The termination sequence is also called the user break 2 sequence. Any character in the ASCII 128-character set can be specified as the termination character.†† This character must differ from the characters currently defined for the CT, BS, CN, AL, and B1 parameters.

CARRIAGE RETURN IDLE COUNT (CI)

The CI parameter specifies the number of idle characters to be inserted into the output stream after a carriage return. This command is necessary since the length of time for the return operation varies depending on the type of terminal being used. When a carriage return is entered, the network outputs the specified number of idle characters before outputting the next line. This allows time for the carriage return function and ensures that characters are not lost because of printing attempts during the carriage return operation. The CI parameter can be assigned any value from 0 through 99, or CI=CA can be specified to restore the carriage return idle count to the default value shown in table K-1.

†Not valid for PSN terminals.

††Refer to table K-1 for the exceptions.

LINE FEED IDLE COUNT (LI)

The LI parameter specifies the number of idle characters to be inserted into the output stream following a line feed. Its purpose is similar to the carriage return idle count (CI) except that the idle characters are output after the line feed instead of after the carriage return. The LI parameter can be assigned any value from 0 to 99, or LI=CA can be specified to restore the line feed idle count to the default value shown in table K-1.

PAGE WIDTH (PW)

The PW parameter establishes the maximum number of characters that the system will print on one output line. Page width can be set to any decimal value from 0 to 255. When output occurs at the terminal, the carriage is advanced to the beginning of the next physical line when the number of characters displayed equals the page width. When a page width of zero is selected, the carriage is never advanced to a new physical line because of page width; PW = 0 selects an infinitely long line.

NOTE

Only specify the PW parameter to establish a page width equal to the physical page width of the terminal being used. Setting page width to other values may cause unpredictable results.

PAGE LENGTH (PL)

The PL parameter establishes the maximum number of physical lines that can be printed as one page. Page length can be set to any value from 0 to 255. PL=0 selects an infinitely long page. During output at the terminal, when the number of lines output is one less than the PL parameter selected, a page boundary is reached. If page waiting has been selected (refer to the PG parameter) output stops until an entry is made at the terminal (ⒸR) to continue output of the next page. Even if page waiting is not enabled, the network performs several line feed functions upon reaching a page boundary. This can be useful when separating printout paper into individual pages.

PAGE WAIT (PG)

The PG parameter specifies whether page waiting is to be performed for the terminal. Page waiting is selected by entering PG=Y and cancelled by entering PG=N. Page waiting is in effect only when the output device is defined to be a CRT display. When page waiting is selected, output stops at each page boundary, and terminal input (ⒸR) is required before the next page is displayed. When the current output page is not full, and another page is available for output, the terminal displays:

OVER..

at the end of displayed output. When page waiting is in effect, a null input line is used to display the next page (entering ⒸR only); it has no other meaning. Functions normally performed when ⒸR only is entered (for example, continuing after an interruption, causing normal abort processing after a time limit) are not performed when page wait is in effect and a page is available for output. To perform these functions, a space followed by ⒸR should be entered.

In character mode, a page is determined by the number of physical lines specified by the PL parameter. During transparent mode output, the end of transparent output is interpreted as a page boundary.

PARITY SELECTION (PA)†

The PA parameter specifies the type of parity that the terminal generates on input and expects on output. Parity can be odd (O), even (E), zero (Z), or none (N). In zero parity (Z), the network does not check for parity and automatically clears the eighth bit. In all cases, even or odd parity, the network checks (or generates) the eighth bit of each character as a parity bit. The network clears the bit before transmitting the character to IAF. If character mode is set and no parity (N) is specified, the network does not check for parity and automatically clears the eighth bit. If transparent mode is set and no parity is specified, the network transmits all eight bits as data without checking parity.

SPECIAL EDITING (SE)†

The SE parameter specifies whether the terminal is in special editing mode. Special editing mode is selected by entering SE=Y and cancelled by entering SE=N. When the terminal is in special editing mode, the cancel input, backspace, and line feed characters are sent to IAF as part of input data. If a backspace followed by a line feed is entered, the system responds with a caret and line feed.

Availability of special editing depends on the site. If special editing is not supported at the site, the network responds

ERR..

to entry of the SE parameter.

If you have entered the APL command, APL has automatically set SE=Y.

TRANSPARENT INPUT MODE DELIMITER (DL)

The DL parameter specifies transparent input mode text delimiters. DL is valid only in a terminal definition command; it is not allowed as a parameter in the TRMDEF command. Three types of delimiters can be selected: characters, character count, and timeout.†† Each is optional, but at least one must be selected. A delimiter character can be any character and is specified with Xcc, where cc is a two-hexadecimal-digit representation of the delimiter character in the terminal's character set code. Character count delimiter is specified with Cvalue, where value can be any decimal value from 1 to 4096.††† The timeout delimiter is selected by including T0, which indicates a timeout of from 200 to 400 milliseconds. Delimiters can be entered in any order; trailing commas can be deleted. Terminal class 4 (IBM 2741) is configured with the RETURN key as the transparent input mode delimiter; terminal classes 12 and 13 are configured with the ETX key as the transparent input mode delimiter. These configurations cannot be changed.

When transparent input mode is initiated, using IN=XK, IN=XP, or IN=X, characters entered are sent to the system as 8-bit characters (if PA=N is specified; otherwise as 7-bit characters with the eighth bit zeroed) without translation, until one of the delimiter values is encountered. Normally, characters entered from the terminal are translated and stored as 6-bit or 12-bit internal display codes (refer to appendix A).

† Not valid for PSN terminals.

†† Terminals connected to a PSN can only use the character delimiter.

††† These values are valid within the network. However, unless changed during installation, the maximum logical input line length in IAF (160 characters) limits you.

INPUT DEVICE (IN)

The IN parameter specifies the input device as a keyboard in character mode (KB), a keyboard in transparent input mode (XK), a paper tape reader in character mode (PT), a paper tape reader in transparent input mode (XP), or the current input device in transparent input mode (X). The TRMDEF command allows character mode (KB and PT) only; values of XK, XP, and X are illegal.† Paper tape input is allowed in keyboard mode, but X-ON characters are not sent to start the paper tape reader. For transparent input mode, the transparent input delimiter (DL) character must have been established before transparent input mode is entered or the default delimiter for the terminal class is used.

OUTPUT DEVICE (OP)††

The OP parameter specifies the output device as a printer (PR), CRT display (DI), or paper tape punch (PT). It switches between a paper tape punch (PT) and the default device (DI or PR). DI cannot be specified if PR is the default device, and vice versa. Otherwise, printer and CRT display are functionally equivalent except for the page wait feature (page wait applies only for OP=DI). The terminal user can punch a paper tape in any mode, but the proper X-OFF characters are provided only if OP=PT is selected and the terminal is not in transparent mode.

ECHOPLEX MODE (EP)††

The EP parameter selects input character echoing. If EP=Y is specified, each character received by the network is echoed to the terminal just as it was received. This mode is effective only for terminals which have echoplex capability. For terminals having a HALF/FULL duplex switch, if this switch is in the HALF position when EP=Y is specified, each subsequent character entered is double printed (initially when it is typed and again when it is echoed by the system). Placing the switch in the FULL position allows only the character echoed to the terminal to be printed.

EP=N clears the EP=Y setting. Characters received by the network after EP=N is specified are not echoed to the terminal. Characters entered at the terminal are not printed when the HALF/FULL duplex switch is in the FULL position. Only system-generated output appears at the terminal. Placing the switch in the HALF position allows keyboard entry to be printed.

MESSAGE COMMAND (MS)

The same network mechanism that allows terminal users to perform the functions provided with the terminal definition commands allows them to send messages to the local operator. The local operator is responsible for controlling the communications elements of the network. Functions of the local operator include enabling and disabling terminals and applications, determining the status of the various communications elements, and sending messages to terminals (refer to the NOS 2 Operator/Analyst Handbook for a thorough description of local operator functions). If you are having problems with the network (such as difficulty connecting to IAF), you can communicate these problems to the local operator using the message command. The format of this command is as follows:

control MS=message (CR)

†Terminals connected to a PSN can only select the KB, XK, or X parameter.

††Not valid for PSN terminals.

<u>Parameter</u>	<u>Description</u>
control	This is the same terminal control character used in the terminal definition commands.
message	Message of no more than 50 characters (including spaces) to be sent to the local operator.

The format of this command is exactly the same as that of the terminal definition commands; spaces are included for clarity only.

Any response from the local operator is printed in the following format.

```
FROM THE LOP:
  message
```

CHARACTER CODE SET COMMAND

The character code set command changes a terminal's character set, usually by means of a switch on the terminal or a removable type ball.† This command can be entered any time after the terminal identification part of login is complete. The format of the command is as follows:

```
control    CD=A  Ⓞ
```

<u>Character</u>	<u>Description</u>
control	This is the same terminal control character used in the terminal definition commands.

The format of this command is the same as that of the terminal definition commands; spaces are included for clarity only.

After entering the character code set command, you have 60 seconds to physically select the terminal character set (for example, change the type ball) and enter one of the following:

<u>Character</u>	<u>Description</u>
) Ⓞ	The system determines the terminal's new character set.
Ⓞ	The system bypasses character set recognition (valid for standard ASCII terminals only). The terminal's new character set is assumed to be ASCII.

When the system recognizes the new character set, it issues two line feeds to indicate you can continue. This is the same procedure used to connect the terminal to the system (refer to the login procedure in Volume 1, Introduction to Interactive Usage). If no entry is made within 60 seconds after entering the character code set command, the network disconnects the terminal. If the character code set of the terminal is not supported at the site, the network issues the message:

```
UNSUPPORTED CODE SET.
```

† Terminals connected to a PSN use the ASCII code set.

Table K-1. Terminal Definition Parameters (Sheet 1 of 4)

Parameter	Mnemonic	Terminal Type			
		M33, M35, M37, M38	CDC 713-10	Reserved	IBM 2741
Terminal class	TC	1	2	3	4
Control character	CT	ESC (any) †	ESCAPE (any) †		ATTN % ÷ for APL (any) †
Backspace character	BS	CTRL/H (any) †	← (any) †		BACKSPACE (any) †
Cancel character	CN	CTRL/X (any) †	CTRL/X (any) †		ATTN ((any) †
Abort output line character	AL	CTRL/X (any) †	CTRL/X (any) †		ATTN ((any) †
User break 1 (interruption character)	B1	CTRL/P (any) †	CTRL/P (any) †		ATTN : (any) †
User break 2 (termination character)	B2	CTRL/T (any) †	CTRL/T (any) †		ATTN) (any) †
Carriage return idle count	CI	2 (0-99, CA)	0 (0-99, CA)		CA (0-99, CA) ††
Line feed idle count	LI	1 (0-99, CA)	0 (0-99, CA)		1 (0-99, CA)
Page width	PW	72 (0-255)	80 (0-255)		132 (0-255)
Page length	PL	0 (0-255)	0 (0-255)		0 (0-255)
Page wait	PG	N (Y,N)	N (Y,N)		N (Y,N)
Parity	PA	E (Z,O,E,N)	E (Z,O,E,N)		∅ (Z,O,E,N)
Special editing	SE	N (Y,N)	N (Y,N)		N (Y,N)
Transparent input mode delimiter	DL †††	X 0D C2043 (X any C1-C4095 TO)	X 0D C2043 (X any C1-C4095 TO)		X 6D (C1-C4095)
Input device	IN ††††	KB (KB,PT,XK,XP,X)	KB (KB,PT,XK,XP,X)		KB (KB,PT,XK,XP,X)
Output device	OP	PR (PR,PT)	DI (DI,PT)		PR (PR,PT)
Echoplex mode	EP	N (Y,N)	N (Y,N)		N/A

†Any ASCII character except NUL, STX, EOT, LF, CR, DEL, =, and space.
 ††The carriage return idle count is automatically incremented for every 10 characters after the first 40 characters in a line.
 For example, if a line had 72 characters, the idle count would be 7.
 †††DL is not a legal parameter in the TRMDEF command.
 ††††XK, XP, and X are not legal values for the IN parameter in the TRMDEF command.

Table K-1. Terminal Definition Parameters (Sheet 2 of 4)

Parameter	Mnemonic	Terminal Type			
		M40	Hazeltine 2000	CDC 751-1	Tektronix 4000
Terminal class	TC	5	6	7	8
Control character	CT	CTRL/P (any) †	ESC (any) †	ESC (any) †	ESC (any) †
Backspace character	BS	No default (any) †	CTRL/H (any) †	← (any) †	CTRL/H (any) †
Cancel character	CN	CTRL/X (any) †	CTRL/X (any) †	CTRL/X (any) †	CTRL/X (any) †
Abort output line character	AL	CTRL/X (any) †	CTRL/X (any) †	CTRL/X (any) †	CTRL/X (any) †
User break 1 (interruption character)	B1	CTRL/F (any) †	CTRL/P (any) †	CTRL/P (any) †	CTRL/P (any) †
User break 2 (termination character)	B2	CTRL/T (any) †	CTRL/T (any) †	CTRL/T (any) †	CTRL/T (any) †
Carriage return idle count	CI	1 (0-99, CA)	0 (0-99, CA)	0 (0-99, CA)	0 (0-99, CA)
Line feed idle count	LI	3 (0-99, CA)	3 (0-99, CA)	0 (0-99, CA)	0 (0-99, CA)
Page width	PW	74 (0-255)	74 (0-255)	80 (0-255)	74 (0-255)
Page length	PL	0 (0-255)	0 (0-255)	0 (0-255)	0 (0-255)
Page wait	PG	N (Y,N)	N (Y,N)	N (Y,N)	N (Y,N)
Parity	PA	E (Z,O,E,N)	E (Z,O,E,N)	E (Z,O,E,N)	E (Z,O,E,N)
Special editing	SE	N (Y,N)	N (Y,N)	N (Y,N)	N (Y,N)
Transparent input mode delimiter	DL ††	X 0D C2043 (X any C1-C4095) TO	X 0D C2043 (X any C1-C4095) TO	X 0D C2043 (X any C1-C4095) TO	X 0D C2043 (X any C1-C4095) TO
Input device	IN †††	KB (KB,PT,XX,XP,X)	KB (KB,PT,XX,XP,X)	KB (KB,PT,XX,XP,X)	KB (KB,PT,XX,XP,X)
Output device	OP	DI (DI,PT)	DI (DI,PT)	DI (DI,PT)	DI (DI,PT)
Echoplex mode	EP	N (Y,N)	N (Y,N)	N (Y,N)	N (Y,N)

† Any ASCII character except NUL, STX, EOT, LF, CR, DEL, =, and space.
 †† DL is not a legal parameter in the TRMDEF command.
 ††† XK, XP, and X are not legal values for the IN parameter in the TRMDEF command.

Table K-1. Terminal Definition Parameters (Sheet 3 of 4)

Parameter	Mnemonic	Terminal Type			
		HASP Protocol	200UT	CDC 714-30	CDC 711-10
Terminal class	TC	9	10	11	12
Control character	CT	% (any)†	% (any)†	% (any)†	% (any)†
Backspace character	BS	N/A	N/A	N/A	N/A
Cancel character	CN	((any)†	((any)†	((any)†	((any)†
Abort output line character	AL	N/A	N/A	N/A	N/A
User break 1 (interruption character)	B1	: (any)†	: (any)†	: (any)†	: (any)†
User break 2 (termination character)	B2) (any)†) (any)†) (any)†) (any)†
Carriage return idle count	CI	N/A	N/A	N/A	N/A
Line feed idle count	LI	N/A	N/A	N/A	N/A
Page width	PW	80 (0-255)	80 (0-255)	80 (0-255)	80 (0-255)
Page length	PL	0 (0)	13 (0-255)	16 (0-255)	16 (0-255)
Page wait	PG	N/A	Y (Y,N)	Y (Y,N)	Y (Y,N)
Parity	PA	N/A	0	0	0
Special editing	SE	N/A	N/A	N/A	N/A
Transparent input mode delimiter	DL††	N/A	N/A	N/A	N/A
Input device	IN†††	KB	KB	KB (KB,XK,X)	KB (KB,XK,X)
Output device	OP	DI	DI	DI	DI
Echoplex mode	EP	N/A	N/A	N/A	N/A

† Any ASCII character except NUL, STX, EOT, LF, CR, DEL, =, and space.
 †† DL is not a legal parameter in the TRMDEF command.
 ††† XK, XP, and X are not legal values for the IN parameter in the TRMDEF command.

Table K-1. Terminal Definition Parameters (Sheet 4 of 4)

Parameter	Mnemonic	Terminal Type		
		CDC 714-10/20	CDC 731-12 CDC 732-12	CDC 734
Terminal class	TC	13	14	15
Control character	CT	% (any) †	% (any) †	% (any) †
Backspace character	BS	N/A	N/A	N/A
Cancel character	CN	((any) †	((any) †	((any) †
Abort output line character	AL	N/A	N/A	N/A
User break 1 (interruption character)	B1	: (any) †	: (any) †	: (any) †
User break 2 (termination character)	B2) (any) †) (any) †) (any) †
Carriage return idle count	CI	N/A	N/A	N/A
Line feed idle count	LI	N/A	N/A	N/A
Page width	PW	80 (0-255)	80 (0-255)	80 (0-255)
Page length	PL	16 (0-255)	13 (0-255)	13 (0-255)
Page wait	PG	Y (Y,N)	Y (Y,N)	Y (Y,N)
Parity	PA	0	0	0
Special editing	SE	N/A	N/A	N/A
Transparent input mode delimiter	DL††	N/A	N/A	N/A
Input device	IN†††	KB (KB,XK,X)	KB	KB
Output device	OP	DI	DI	DI
Echoplex mode	EP	N/A	N/A	N/A

†Any ASCII character except NUL, STX, EOT, LF, CR, DEL, =, and space.
††DL is not a legal parameter in the TRMDEF command.
†††XK, XP, and X are not legal values for the IN parameter in the TRMDEF command.

LINE FEED AND TRANSMISSION KEYS

L

Table L-1 shows the keys used by the supported terminal classes to advance the carriage or cursor to the beginning of the next line and/or transmit messages from the terminal. The terminal classes are described in appendix K. The keys listed in table L-1 are divided functionally into three types called carriage return, new line, and line feed.

Table L-1. Terminal Key Equivalences

Terminal Class	Function		
	Carriage Return	New Line	Line Feed
1	RETURN	N/A	LINE FEED
2	RETURN	N/A	↓
3			
4	RETURN	N/A	ATTN
5	RETURN	N/A	NEW LINE
6	CR	N/A	LF
7	CARRIAGE RETURN†	N/A	LINE FEED
8	RETURN	N/A	LF
9	Varies ††	Varies	Varies
10	SEND	RETURN	N/A
11	SEND	RETURN	N/A
12	ETX	NEW LINE	N/A
13	ETX	NEW LINE	N/A
14	ETX	NEW LINE	N/A
15	SEND	NEW LINE	N/A

†If in block mode, refer to terminal documentation for terminal key equivalences. Those listed are for character and line modes.
 ††Terminals operating under HASP protocol use different keys for this purpose.

The functions each of the three types of keys provides for you at a terminal, and how IAF interprets each type are described in the following paragraphs.

Terminal user functions are:

<u>Type of Key</u>	<u>Function</u>
Carriage return	Terminates a message, advances the cursor or carriage to the beginning of the next line, and transmits the message. It also transmits any other messages which may be stored in the terminal's buffer.
New line	Terminates the message and advances the cursor or carriage, but does not transmit the message. Rather, it causes the message to be stored in the terminal's buffer until the carriage return function is used. Many terminals cannot activate the new line function and on other terminals the new line key does not activate the new line function. Refer to terminal documentation for information concerning the keys on the terminal.
Line feed	Terminates a physical line of input and advances the cursor or carriage to the beginning of the next line.

IAF interprets these functions as:

<u>Type of Key</u>	<u>IAF Interpretation</u>
Carriage return	Terminates a message.
New line	Terminates a message. Subsequent messages in the same transmission may not be accepted.
Line feed	Terminates a physical line of input and advances the cursor or carriage to the beginning of the next line. IAF concatenates information entered after a line feed with the previous line to form a single logical line.

PASSIVE PROCEDURES

M

Passive procedures are procedures that do not have the parameter-prompting capability. If the procedure call contains errors, the job step terminates and the procedure is not executed. In most respects, however, the structure and behavior of passive and parameter-prompting procedures are highly similar. This appendix shows only the areas in which passive procedures differ from the parameter-prompting variety. Section 4 contains a detailed description of parameter-prompting procedures.

.PROC DIRECTIVE

This format of the .PROC directive does not provide for the interactive entry of parameters nor will the system give parameter descriptions or other help.

Format:

.PROC, pname, p₁, p₂, ..., p_n.

<u>Parameters</u>	<u>Description</u>
pname	Name of the procedure; one to seven alphanumeric characters. As with file names, the name of a procedure should begin with a letter.
p _i	Optional formal parameters whose keywords are used in the body of the procedure. Depending on the parameters specified in the procedure call, keywords in the procedure body can be removed, left as they are, replaced by a value specified in the procedure call, or replaced by first or second default values as specified on the procedure header parameter (refer to keyword substitution in this section).

The maximum number of procedure header keywords is defined by the installation. The default is 50.

The following are the acceptable formats for p_i.

<u>Format</u>	<u>Example</u>
keyword	FILE1
keyword=	FILE1=
keyword=default1	FILE1=LGO
keyword=default1/default2	FILE1=LGO/OLD
keyword=/default2	FILE1=/OLD
keyword=#DATA (CDC graphics: keyword= ≡ DATA)	FILE1=#DATA
keyword=#FILE (CDC graphics: keyword= ≡ FILE)	FILE1=#FILE
keyword	A 1- to 10-character keyword.

Parameters

Description

default1 A 1- to 40-character first default value. If default1 contains special characters, it must be \$-delimited. This default value replaces the keyword in the procedure body if this parameter is omitted from the procedure call.

default2 A 1- to 40-character second default value. If default2 contains special characters, it must be \$-delimited. This default value replaces the keyword if the procedure call specifies a parameter value identical to the keyword.

default1 and default2 could be either of the following special values.

#FILE Special default value used for keyword if an overriding value is not specified on the procedure call. If this default value is used, keyword within the procedure body references the record that immediately follows the procedure record on the file.

#DATA Special default value used for keyword if an overriding value is not specified on the procedure call. If this default value is used, keyword within the procedure body references a data file created within the procedure by the .DATA directive (refer to .DATA directive in section 4).

PROCEDURE AND PARAMETER DESCRIPTIONS

You cannot include the .HELP or .ENDHELP directive in your procedure body. Hence, you cannot provide the procedure caller with a description of the procedure or any of its parameters unless the caller has already entered an error-free procedure call.

PARAMETER MATCHING

When a procedure is called, the system must match each parameter on the call command to a parameter on the procedure header command. The system uses two methods of parameter matching; order-dependent and order-independent.

ORDER-DEPENDENT PARAMETER MATCHING MODE

Parameter matching always begins in order-dependent mode (refer to Order-Independent Parameter Matching Mode for information on changing parameter matching modes). The system compares, in order, each parameter on the BEGIN command with the parameter in that position on the procedure header directive. The system then substitutes the selected parameters into the procedure body.

All possible parameter substitutions in order-dependent mode are summarized in table M-1. The table shows each parameter format on the BEGIN command, each parameter format on the procedure header, and the substitution resulting from each combination. In the table the word value indicates that the parameter in the BEGIN command (called value) is different

than the corresponding keyword and/or defaults on the procedure header. Keyword in the BEGIN Command Parameter Format heading means the keyword in the BEGIN command is identical to the keyword in the procedure header parameter.

Assuming that all parameter matches between the BEGIN command and the procedure header are valid for order-dependent mode (table M-1), the system completes parameter matching in order-dependent mode.

In order-dependent mode, the system ignores excess parameters on the BEGIN command.

Use table M-1 with the following examples to clarify the meaning of the table entries (keyword, default1, default2, value, and null).

Examples: Parameter Matching in Order-Dependent Mode.

<u>Procedure on File Named MYFILE</u>	<u>Call and Substitution</u>	<u>Explanation</u>
.PROC,SAMPL1,L,M,N=XY. REWIND,L,A,M,N.	BEGIN,SAMPL1,MYFILE. yields REWIND,L,A,M,XY.	When parameters are omitted on the BEGIN command, the system uses the defaults from the procedure header (L, M, and XY).
	BEGIN,SAMPL1,MYFILE,,,N. yields REWIND,L,A,M,N.	Omitted parameters indicate use of the procedure header defaults (L and M). N overrides the procedure header default (XY).
	BEGIN,SAMPL1,MYFILE,\$*\$,C. yields REWIND,*A,C,XY.	Special characters must be \$-delimited. The asterisk (*) replaces M. The system uses the procedure header default (XY) for the omitted parameter.
.PROC,SAMPL2,LFN1=,LFN2, SBF=/SBF. COPY_SBF,LFN1,LFN2.	BEGIN,SAMPL2,MYFILE. yields COPY,,LFN2.	Omitted parameters indicate use of procedure header defaults (LFN2 and null substitution for LFN1 and SBF).
	BEGIN,SAMPL2,MYFILE,,,SBF. yields COPYSBF,,LFN2.	Omitted parameters indicate use of procedure header defaults (null and LFN2). The BEGIN command parameter, SBF, indicates use of the second default of the SBF procedure header parameter (SBF). The linking character (__) connects COPY and SBF to make COPYSBF.
	BEGIN,SAMPL2,MYFILE,FORMS,OUTPUT. yields COPY,FORMS,OUTPUT.	FORMS replaces LFN1 and OUTPUT replaces LFN2. Since the third parameter is omitted, the system uses the procedure header default (null).
.PROC,SAMPL3,PFN, P1=/\$M=W\$. ATTACH,PFN/P1.	BEGIN,SAMPL3,MYFILE,TAXES,P1. yields ATTACH,TAXES/M=W.	TAXES replaces PFN and the character string M=W replaces P1.

Table M-1. Parameter Substitution In Order-Dependent Mode

		BEGIN Command Parameter Format			
		omitted	keyword	value	\$value\$
Procedure Header	keyword	keyword	keyword	value	value
Parameter Format	keyword=	null	keyword	value	value
	keyword=default1	default1	keyword	value	value
	keyword=default1/default2†	default1	default2	error	error

† Switches keyword substitution to order-independent mode for all subsequent parameters.

Example 1:

The following procedure is on file PROCFIL. It prepares a file for processing. If the file is local, it is rewound. If it is not local, the system searches for the file in your permanent file catalog. If the file is not found, the procedure reverts and aborts.

```
.PROC,PREPARE,FNAME=M,R.
IFE,FILE(FNAME,AS),PREP1.
REWIND,FNAME.
REVERT,FNAME PREPARED.
ENDIF,PREP1.
ATTACH,FNAME/#M=M,NA.
IFE,FILE(FNAME,.NOT.AS),PREP2.
GET,FNAME/NA.
IFE,FILE(FNAME,.NOT.AS),PREP3.
REVERT,ABORT.FNAME NOT FOUND.
ENDIF,PREP3.
ENDIF,PREP2.
REVERT,FNAME PREPARED.
EXIT.
REVERT,ABORT.PREPARE ERRORS.
```

Prepare file TEST with the following command.

```
BEGIN,PREPARE,,TEST.
```

Since PROCFIL is the default file, it does not have to be specified and is noted by successive commas.

The following is a segment of the dayfile that results when the BEGIN command is processed.

```
08.26.45.$BEGIN,PREPARE,,TEST.
08.26.46.IFE,FILE(TEST,AS),PREP1.
08.26.46.ENDIF,PREP1.
08.26.46.ATTACH,TEST/M=R,NA.
08.26.46. TEST IS INDIRECT ACCESS, AT 000121.
08.26.46.IFE,FILE(TEST,.NOT.AS),PREP2.
08.26.46.GET,TEST/NA.
08.26.47.IFE,FILE(TEST,.NOT.AS),PREP3.
08.26.47.ENDIF,PREP3.
08.26.47.ENDIF,PREP2.
08.26.47.REVERT. TEST PREPARED.
```

Example 2: Parameter Matching in Nested Procedures (Order-Dependent Parameter Matching Mode)

The substitutions made in a procedure that calls a second procedure are shown in figure M-1. The resultant dayfile is shown on the right side of the figure.

GET,PROGRAM1. BEGIN,EXECUTE,PFILE1,PROGRAM1,PRINT.	
PFILE1	RESULTANT DAYFILE
.PROC,EXECUTE,NAME,OUT. FTN5,I=NAME,L=OUT. LGO. IFE,EF=0,DROP. BEGIN,LISTING,PFILE2,OUT. ENDIF,DROP.	16.01.08.GET,PROGRM1. 16.01.08.BEGIN,EXECUTE,PFILE1,PROGRM1,PRINT. 16.01.09.FTN5,I=PROGRM1,L=PRINT. 16.01.10. .043 CP SECONDS COMPILATION TIME 16.01.10.LGO. 16.01.11. STOP 16.01.11. .038 CP SECONDS EXECUTION TIME 16.01.12.IFE,EF=0,DROP. 16.01.12.BEGIN,LISTING,PFILE2,PRINT. 16.01.12.REWIND,PRINT. 16.01.13.COPYSBF,PRINT,OUTPUT. 16.01.13. END OF INFORMATION ENCOUNTERED. 16.01.13.\$REVERT.CCL 16.01.13.ENDIF,DROP. 16.01.13.\$REVERT.CCL
PFILE2 .PROC,LISTING,OUTFILE=OUT. REWIND,OUTFILE. COPYSBF,OUTFILE,OUTPUT.	

Figure M-1. Keyword Substitution in Two Procedures

ORDER-INDEPENDENT PARAMETER MATCHING MODE

The system switches to order-independent mode to match the remainder of the parameters if in comparison of a BEGIN command parameter and a procedure header parameter one of the following occurs.

- A BEGIN command parameter is in the format keyword= or keyword=value.
- A procedure header command parameter is in the format keyword=default/default2.

For each BEGIN command parameter, matching always begins in order-dependent mode. Once in order-independent mode, the system matches each keyword of the BEGIN command to the identical keyword in the procedure header command, regardless of order.

The following commands illustrate the parameter combinations that result in switching from order-dependent mode to order-independent mode.

<u>Procedure on Default</u> <u>File PROCFIL</u>	<u>Call and Substitution</u>	<u>Explanation</u>
.PROC,SALES,TAX,TOTAL=, FLAG=A. COPYL,TAX,TOTAL,HOLD,,FLAG. REPLACE,HOLD=TAX.	BEGIN,SALES,,TAX,TOTAL=SUM,FLAG. yields COPYL,TAX,SUM,HOLD,,A. REPLACE,HOLD=TAX.	Parameter matching starts in order-dependent mode. The BEGIN parameter TOTAL=SUM switches the mode to order-independent mode. FLAG is then matched in order-independent mode, which yields A.
.PROC,TAXES,TAX=FED/MN, DEDUCT,FLAG=A. COPYL,TAX,DEDUCT,HOLD,,FLAG. REPLACE,HOLD=TAX.	BEGIN,TAXES,,TAX,DEDUCT. yields COPYL,MN,DEDUCT,HOLD,,A. REPLACE,HOLD=MN.	The TAX=FED/MN procedure parameter switches the mode to order-independent mode. All parameters will be matched in order-independent mode.

All possible parameter substitutions in order-independent mode are summarized in table M-2. The table shows each parameter format on the BEGIN command, each parameter format on the procedure header command, and the substitution resulting from each combination.

In the table the word value indicates that the parameter in the BEGIN command (called value) is different than the keyword and/or defaults on the procedure header command. Use table M-2 with the following examples to clarify the meaning of the table entries (keyword, default1, default2, value, and null).

Examples of Parameter Matching:

<u>Procedure on Default File PROCFIL</u>	<u>Call and Substitution</u>	<u>Explanation</u>
.PROC,SAMPL1,L,M,N=XY. REWIND,L,A,M,N.	BEGIN,SAMPL1,,L=SWITCH. yields REWIND,SWITCH,A,M,XY.	The L=SWITCH BEGIN parameter switches parameter matching mode to order-independent mode. Order-independent mode uses the procedure header defaults for omitted BEGIN parameters. Order-dependent and order-independent modes work identically for omitted BEGIN parameters.
	BEGIN,SAMPL1,,L=CHANG,M,N. or BEGIN,SAMPL1,,L=CHANG,N,M. yields REWIND,CHANG,A,M,XY.	The L=CHANG parameter switches parameter matching to order-independent mode. In order-independent mode the order of the BEGIN parameters does not matter. M matches with M, and the N BEGIN keyword indicates substitution of the procedure header default (XY).
	BEGIN,SAMPL1,,L=FLIP,M=B,N=Z. yields REWIND,FLIP,A,B,Z.	BEGIN parameters in the form keyword=value always override procedure header parameters. FLIP replaces L, B replaces M, and Z replaces XY.
.PROC,TRACE,IN,OUT, TC=EOI,N=1. COPY,IN,OUT,,,TC,N.	BEGIN,TRACE,,IN=,OUT=HOLD,N=. yields COPY,,HOLD,,,EOI,.	The IN= parameter switches parameter matching to order-independent mode. All BEGIN commands in the form keyword= use null substitution.
.PROC,SAMPL4,FILE1, EC=B6/A6,DC=LP,REP=0. REWIND,FILE1. ROUTE,FILE1,#DC=DC, #EC=EC,#REP=REP.	BEGIN,SAMPL4,,COIN. yields REWIND,COIN. ROUTE,COIN,DC=LP,EC=B6,REP=0.	COIN is substituted in order-dependent mode. EC=B6/A6 switches the mode to order-independent mode. The omitted parameters indicate use of the procedure header defaults (order-dependent and order-independent modes work alike for omitted BEGIN parameters).
	BEGIN,SAMPL4,,COIN,EC,DC,REP. yields REWIND,COIN. ROUTE,COIN,DC=LP,EC=A6,REP=0.	EC=B6/A6 switches the mode to order-independent mode. Specifying the keyword on the BEGIN command produces the same result as omitting them (refer to previous example) except for the double default procedure parameter, EC=B6/A6. If EC is omitted, B6 is used. If EC is specified, A6 is used.

Table M-2. Parameter Substitution In Order-Independent Mode

		BEGIN Command Parameter Format				
		omitted	keyword	keyword=	keyword=value	value or \$value\$ †
Procedure Header Parameter Format	keyword	keyword	keyword	null	value	error
	keyword=	null	null	null	value	error
	keyword=default1	default1	default1	null	value	error
	keyword=default1/ default2†	default1	default2	null	value	error
† Assumes the parameter is entered under order-independent mode.						

Example 1:

The following procedure is the same procedure as Example 2 in Order-Dependent Parameter Matching Mode. It resides on file PROCFIL. It routes a specified file (FNAME) to the specified equipment (default is any CDC-graphics line printer).

```
.PROC,PRINTR,FNAME,REP=0,DC=LP,EC=B6.
REWIND,FNAME.
ROUTE,FNAME,#DC,#REP=REP,#EC=EC.
REVERT. FNAME ROUTED.
EXIT.
REVERT,ABORT. PRINTR PARAMETER ERRORS.
```

The following command calls the procedure PRINTR. The system matches COLOR in order-dependent form. DC=SB switches the mode to order-independent mode. SB indicates the file is to be punched.

```
BEGIN,PRINTR,,COLOR,DC=SB,EC=SB.
```

The following is a segment of the dayfile that results when the BEGIN command is processed.

```
15.27.26.$BEGIN,PRINTR,,COLOR,DC=SB,EC=SB.
15.27.27.REWIND,COLOR.
15.27.27.ROUTE,COLOR,DC=SB,REP=0,EC=SB.
15.27.27.ROUTE COMPLETE.
15.27.28.REVERT. COLOR ROUTED.
```

Example 2: Parameter Matching in Nested Procedures (Order-Dependent and Order-Independent Parameter Matching Modes)

As shown in figure M-2, procedures ROUT and PREPARE reside on the default file PROCFIL. A BEGIN command within ROUT calls PREPARE. In procedure ROUT the substitution for the FNAME parameter (TEST) is passed to procedure PREPARE by the BEGIN command. The resulting dayfile is on the right side of figure M-2.

PROCFIL	Resultant Dayfile
BEGIN,ROUT,,TEST,EC=A9,SBF.	15.27.50.\$BEGIN,ROUT,,TEST,EC=A9,SBF.
	15.27.50.BEGIN,PREPARE,,FNAME=TEST.
.PROC,ROUT,FNAME=L,SBF=/SBF,REP=0, EC=B6,DC=LP.	15.27.51.IFE,FILE(TEST,AS),PREP1.
BEGIN,PREPARE,,#FNAME=FNAME.	15.27.51.REWIND,TEST.
COPY SBF,FNAME,HOLD.	15.27.51.REVERT. TEST PREPARED.
REWIND,HOLD.	15.27.51.COPYSBF,TEST,HOLD.
IFE,\$EC\$.EQ.\$A9\$,JUMP.	15.27.52. EOI ENCOUNTERED.
FCOPY,P=HOLD,N=TEMP.	15.27.52.REWIND,HOLD.
RENAME,HOLD=TEMP.	15.27.52.IFE,\$A9\$.EQ.\$A9\$,JUMP.
ENDIF,JUMP.	15.27.52.FCOPY,P=HOLD,N=TEMP.
REWIND,HOLD.	15.27.52. FCOPY COMPLETE.
ROUTE,HOLD,#DC=DC,#EC=EC,#REP=REP.	15.27.52.RENAME,HOLD=TEMP.
REVERT. FNAME -> PRINTER.	15.27.52.ENDIF,JUMP.
EXIT.	15.27.52.REWIND,HOLD.
REVERT,ABORT. ROUT ERRORS.	15.27.52.ROUTE,HOLD,DC=LP,EC=A9,REP=0.
-EOR-	15.27.53. ROUTE COMPLETE.
.	15.27.53.REVERT. TEST -> PRINTER.
-EOR-	
.PROC,PREPARE,FNAME=M,R.	
IFE,FILE(FNAME,AS),PREP1.	
REWIND,FNAME.	
REVERT. FNAME PREPARED.	
ENDIF,PREP1.	
ATTACH,FNAME/#M=M,NA.	
IFE,FILE(FNAME,.NOT.AS),PREP2.	
GET,FNAME/NA.	
IFE,FILE(FNAME,.NOT.AS),PREP3.	
REVERT,ABORT. FNAME NOT FOUND.	
ENDIF,PREP3.	
ENDIF,PREP2.	
REVERT. FNAME PREPARED.	
EXIT.	
REVERT,ABORT. PREPARE ERRORS.	

BEGIN calls procedure ROUT. The SBF=/SBF parameter switches parameter matching to order-independent mode. The first command of ROUT is a BEGIN command that calls procedure PREPARE. The parameters are matched in order-independent mode. PREPARE reads a file for processing. If the file is local, it is rewound. If it is not local, the system searches for the file in your permanent file catalog. If the file is not found, the procedure reverts and aborts. If the file is found, processing continues with the second command in procedure ROUT. The file is prepared for printing. Since the file is to be printed with the ASCII graphic 95-character set (EC=A9), the file must be changed to a 12-bit ASCII code file (FCOPY). The procedure then routes the file to the printer and reverts to the command following the BEGIN command.

Figure M-2. Keyword Substitution in Nested Procedures

PAPER TAPE OPERATIONS

N

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 you 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 you 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>
ⒸⓇ	Message terminator. This is the RETURN key on the model 33. It moves the print head to the beginning of the line and informs IAF 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, IAF 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 and to allow time for the carriage to return to the left margin. While it is possible that adequate separation is provided with less than three rubouts, this is the recommended number.

PROGRAM LINE

100 PROGRAM TEST ⒸⓇ LF RO(3)

DATA LINE

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

(The ? is supplied by IAF. 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 ⒸⓇ 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.)

PUNCHING 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, you enable IAF to synchronize tape input with its processing.†

You enter tape mode by using the TRMDEF command (section 8) or a terminal definition command (appendix K) to change the input device to a paper tape reader. To do this with the TRMDEF command, enter:

```
TRMDEF,IN=PT
```

To perform the same operation with a terminal definition command, press the ESC key‡ (assuming a model 33 teletypewriter is being used) and enter

```
IN=PT
```

The TRMDEF command can specify the input device to be in character mode (PT or KB) only; to specify the input device is in transparent input mode (XP, XK, or X), a terminal definition command must be used. For all other paper tape operations, the TRMDEF command and terminal definition commands can be used interchangeably. The remainder of this section will use the TRMDEF command in examples.

The network acknowledges entry into tape mode with the response "IN ACCEPTED." If the reader switch is in the AUTO position, reading begins automatically; if the reader switch is in the OFF position, reading does not begin until it is manually initiated by momentarily moving the switch to START.

†You cannot use paper tape mode from terminals connected to a packet switching network (PSN).

‡‡The ESC key is the default control character for terminal class 1. It must be pressed to signal the entry of a terminal definition command. Because the control character differs for some terminals, refer to appendix K for the default values for all terminal classes.

You can exit from tape mode by using the TRMDEF command or a terminal definition command to change the input device back to the keyboard. This can be done with TRMDEF by entering:

```
TRMDEF,IN=KB
```

The paper tape punch can also be defined as the output device using the TRMDEF command or a terminal definition command. This can be done by entering:

```
TRMDEF,OP=PT
```

You specify this command to enter tape mode for outputting information to paper tape. This mode provides the proper X-OFF characters when punching the tape. You can exit this mode by using the TRMDEF command or a terminal definition command to change the output device back to a printer. This can be done by entering:

```
TRMDEF,OP=PR
```

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, enter TRMDEF,OP=PT.
3. Type in LIST if the primary file is to be punched; type in LIST,F=lfm if a local file other than the primary file is to be punched. Do not press **CR**.
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 network 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 the printout, IAF 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 before and after the tape is read.

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.

100 LET FACTOR=1	Ⓞ	LF	RO(3)
110 INPUT N	Ⓞ	LF	RO(3)
120 FOR I=1 TO N	Ⓞ	LF	RO(3)
130 LET FACTOR=FACTOR*I	Ⓞ	LF	RO(3)
140 PRINT FACTOR	Ⓞ	LF	RO(3)
150 NEXT I	Ⓞ	LF	RO(3)
160 END	Ⓞ	LF	RO(30)

To demonstrate the use of this tape, you must be logged in under the null subsystem. You position the tape in the reader and set the reader switch to AUTO. The following is the printout that results. The right half of the page contains explanations of the action.

basic	←	By typing this command, you enter the BASIC subsystem.
READY.		
new, fact1	←	You type in this line to establish a new primary file with the name FACT1.
trmdef, in=pt		As soon as this command is typed in and 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.
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		
READY.		
trmdef, in=kb	←	You change the input device back to the keyboard.
READY.		
run	←	You type in RUN and receive the diagnostics shown. BASIC variables cannot be more than two characters.
ILLEGAL STATEMENT AT 100		
ILLEGAL STATEMENT AT 130		
ILLEGAL STATEMENT AT 140		
BASIC COMPILATION ERRORS		
100 let f=1	←	You type in the corrective coding shown.
130 let f=f*i		
140 print f		

<pre>run</pre>	<p>← You initiate another run with the RUN command. IAF replies with ?. You enter a 5 and receive the desired printout.</p>
<pre>? 5 1 2 6 24 120</pre>	
<pre>RUN COMPLETE. trmdef,op=pt</pre>	<p>← You want to punch a new tape containing this corrected program. You select paper tape as your output mode.</p>
<pre>READY. list</pre>	<p>← You type LIST but do not press the Ⓞ. You turn on the paper tape punch and run off a leader of blank tape. You then press the Ⓞ. This punches a CR on the tape and the network responds with LF and NUL punches. These three punches are not part of the file, and you must replace them with rubouts if you use the tape for subsequent input.</p>
<pre>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</pre>	<p>← IAF begins the listing with a sequence of CR, NUL(n), LF, and NUL(m).† BRIEF mode suppresses this sequence.</p> <p>IAF punches the tape as it lists the file. IAF terminates each line with CR, NUL(n), LF, NUL(m), DC3, and NUL(2).††</p>
<pre>READY.</pre>	<p>← IAF punches READY. after it has punched the contents of the file. A CR-NUL(n)-LF-NUL(m) punch sequence precedes and follows this message.</p>

†If you enter this tape as input, the CR and LF should be rubbed out, so that the system does not read them as an empty line. The system ignores all NULs.

††Refer to table K-1 in appendix K to find n, the carriage return idle count, and m, the line feed idle count, for your terminal class.

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.

FORTRAN	Ⓞ	LF	CTRL/X-OFF	RO(3)
NEW, DEMO	Ⓞ	LF	CTRL/X-OFF	RO(3)
100 PROGRAM DEMO	Ⓞ	LF	RO(3)	
110 DIMENSION II(5)	Ⓞ	LF	RO(3)	
120 READ 10, (II(J), J=1,5)	Ⓞ	LF	RO(3)	
130 10 FORMAT (515)	Ⓞ	LF	RO(3)	
140 PRINT 20, (II(J), J=1,5)	Ⓞ	LF	RO(3)	
150 20 FORMAT(5I10)	Ⓞ	LF	RO(3)	
160 END	Ⓞ	LF	RO(3)	
RUN	Ⓞ	LF	CTRL/X-OFF	RO(3)
111112222333334444455555	Ⓞ	LF	CTRL/X-OFF	RO(3)
SAVE, DEMO=TAPE1	Ⓞ	LF	CTRL/X-OFF	RO(3)
135 DO 1 I=1,5	Ⓞ	LF	RO(3)	
137 II(I)=II(I)+4444	Ⓞ	LF	RO(3)	
139 1 CONTINUE	Ⓞ	LF	RO(3)	
RESEQ, 100,5	Ⓞ	LF	CTRL/X-OFF	RO(3)
LIST	Ⓞ	LF	CTRL/X-OFF	RO(3)
RUN	Ⓞ	LF	CTRL/X-OFF	RO(3)
111112222333334444455555	Ⓞ	LF	CTRL/X-OFF	RO(3)
REPLACE, DEMO=TAPE1	Ⓞ	LF	CTRL/X-OFF	RO(30)

To execute this tape, you log in to IAF, position the tape in the reader, set the reader switch to AUTO, type the command TRMDEF, IN=PT and, upon pressing Ⓞ, start the reading of the tape. You then observe 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.

FORTRAN ← When this line is read, the reader stops. As soon as FORTRAN is established as the current subsystem, IAF turns the reader back on.

NEW, DEMO ← When this line is read, the reader stops. After a new primary file called DEMO is established, IAF turns the reader back on.

100 PROGRAM DEMO ←
 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

The seven lines of the program are read without interruption.

RUN ← After the command to run the program is read, the reader is turned off.

? 11112222333344445555 ← After processing reaches the READ statement (line 120) and after IAF prints ?, IAF 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
 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.


```

LIST ←
00100 PROGRAM DEMO
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

```

After the LIST command is read from tape, the tape reader is turned off. Then IAF lists the primary file which contains the previous modifications in resequenced format. When the listing is completed, the reader is turned on again.

```

RUN ←
? 1111122222333334444455555

```

After the RUN command is read in, the reader is turned off, and the modified program is executed. IAF 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.

```

15555    26666    37777    48888    59999

```

```

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, you have no more tapes to run, exit tape mode by entering the TRMDEF, IN=KB command.

MASS STORAGE DEVICE STATISTICS

O

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

<u>Device Code</u>	<u>Device</u>	<u>PRUs</u>	<u>Block Size</u>	
			<u>CM Words</u>	<u>Characters</u>
DE	Extended Memory	16	1024	10240
DIn	844-2l 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	844-2l 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 Extended Memory	16	1024	10240
DQn	885 Disk Storage Subsystem (full track; 1<n<3)	n*640	n*40960	n*409600
DV††	819 Disk Storage Subsystem (single density)	400	25600	256000
DW††	819 Disk Storage Subsystem (double density)	800	51200	512000

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

If your validation permits (refer to LIMITS Command, section 7), 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 and by your validation. There are no other system limitations.

†Half-track is a recording mode that accesses alternate PRUs during a disk revolution; full-track recording mode accesses consecutive PRUs. Half-track mode needs two revolutions to access all PRUs on a physical track; full-track mode needs only one revolution.

††Model 176 only.

INDEX

- A registers 14-4
AAM (refer to Advanced Access Methods)
Abbreviated job status command 8-13
Abort C-1
Abort line character 17-6
Abort output line character K-4
Aborting output lines 17-6
ABS record type 16-3
Absolute binary card output F-7
Access category 10-2
ACCESS command 8-8
Access mode 10-7
Access subsystem 8-8,15,24
Accessing files 8-16; 10-6,18,19
Account block 7-2,35,37; C-1
Account dayfile 3-13; 7-2
Accounting 3-7
Accounting activity 7-2
ACF record type 16-3
ADD directive 16-23
Adder activity 3-13
Address registers 14-4
Advanced Access Methods (AAM) 1-6
AFTER directive 16-27
Allocatable device C-1
Alphanumeric C-1
ALTER command 8-29
ANSI-labeled tape 12-13,19
ANSI labels 2-8; 12-7,8,16; G-1
ANSI standards 12-2
APL command K-6
APPEND command 10-6
Append mode 2-14
Appending files 10-6
Appending records 16-11
Application 8-14; C-1
Application activity D-4
Application switching 3-3; C-1
Applications 7-21
Argument C-1
Arithmetic operator 6-2
ASCII C-1
ASCII code terminal 9-5
ASCII command 8-2
ASCII graphic character sets A-1
ASCII mode 8-2,4; 9-5; E-1
ASCII 128-character set A-1
Assembler language 1-5
ASSIGN command 9-1; 12-5
Asynchronous terminals 17-6,8
ATTACH command 10-6
ATTN key 17-9
AUTO command 8-2
Auto-drop status 9-46
Auto eject mode I-2
Auto mode 8-2,12
Autorecognition line K-3
Auxiliary device 2-16; 10-3,20; C-1

B format J-1
B registers 14-4
Backspace character K-3
Backup copy 2-16; 10-2
BAM (refer to Basic Access Methods)
Banner page 3-6; D-2
BASIC C-1
Basic Access Methods (BAM) 1-6
BASIC command 8-8
BASIC subsystem 8-8
BATCH command 8-9
Batch job 3-1
Batch subsystem 8-9
BCD code A-11
BEFORE directive 16-24
BEGIN command 4-19
Beginning-of-information (BOI) 2-2,3,4; C-1
Binary card F-1,5
Binary card output F-6
Binary dump 11-1
Binary file 16-14
Binary mode 2-7
Bit C-1
BKSP command 9-3
BLANK command 12-8
Block C-1
Block count G-8,9
Block size 9-8,10; 12-13; O-1
Block transmission terminal 9-5
Blocking type 2-3
BOI (refer to Beginning-of-information)
BREAK key 17-6
BRIEF command 8-4
BUILD directive 16-24
BYE command 3-3; 8-13
Byte C-2

CALL command H-5
Cancel line character 17-5,6; K-4
CAP record type 16-3
Card file 2-6

Card file data conversion F-1
 Card formats F-2
 Card punch 1-4
 Card reader 1-4
 Cards punched, total 3-14
 Carriage control I-1
 Carriage control character 9-23; I-1
 Carriage control tape I-3
 Carriage return L-2
 Carriage return idle count K-4
 Carriage return key 17-3
 Catalog C-2
 CATALOG command 16-4
 Catalog tracks 2-7
 Cataloging files 16-4
 CATLIST command 10-9
 CATLIST output 10-9
 CCCCCCO file 13-1
 CDC 711-10 terminal 17-1; K-3
 CDC 713-10 terminal 17-1; K-2
 CDC 714-10/20 terminal 17-1; K-3
 CDC 714-30 terminal 17-1; K-3
 CDC 731-12 terminal 17-1; K-3
 CDC 732-12 terminal 17-1; K-3
 CDC 734 terminal 17-1; K-3
 CDC 751-1 terminal 17-1; K-3
 CDC 752 terminal 17-1; K-3
 CDC 756 terminal 17-1; K-3
 CEJ/MEJ (refer to Central/monitor
 exchange jump)
 Central exchange jump/monitor exchange
 jump (CEJ/MEJ) 14-1
 Central library directory 5-5
 Central memory allocation 1-3,6
 Central memory (CM)
 Central memory dump 11-1,4,5,8,9
 Central memory field length (FL) 3-8
 Central memory resident (CMR) 1-3
 Central processor time 3-13; D-4
 Central processor unit (CPU) 1-2; 7-46
 CFO command 7-2
 CHANGE command 10-14
 Character C-2
 Character code conversion A-1,9,11
 Character code set 9-22
 Character code set command K-8
 Character code set conversion 9-22
 Character conversion E-1
 Character echoing K-7
 Character set K-8
 Anomalies A-2
 ASCII A-1
 ASCII graphic 63-character set A-1
 ASCII graphic 64-character set A-1
 ASCII graphic 95-character set A-1
 ASCII 128-character set A-1
 Batch jobs A-2
 Conversion for magnetic tapes A-11
 Definition A-1
 Interactive jobs A-2
 Line printers A-3
 NOS-supported A-1
 Tables A-2
 63-character set A-1
 64-character set A-1
 Character set conversion 9-4
 Character set mode 8-4
 CHARGE command 7-2
 Charge number 7-2
 Checklist 4-3
 Checkpoint 9-32; 13-1; C-2
 Checkpoint dump 13-1
 Checkpoint file 9-2,32; 12-8,14; 13-1; C-2
 Checksum 16-6
 Record 16-15
 Circumflex character (^) 9-5
 CKP command 13-1
 CLEAR command 9-4
 Clock cycle 7-14
 CM (refer to Central memory)
 CMR (refer to Central memory resident)
 Code set 9-22
 ASCII A-1
 Definition A-1
 Display code A-1
 NOS-supported A-1
 12-bit ASCII code A-1
 6/12 display code A-1
 Coded card F-1,3
 Coded card conversion F-4
 Coded card output F-6
 Coded file 9-13,43
 Coded mode 2-7
 Coded record 9-15
 Colon (:) 9-5
 Colon representation A-2; F-1
 Command
 Continuation line 5-2; 6-2; 12-3
 Definition C-2
 Execution control 6-1
 Expression 6-2,4
 File management 9-1
 Flow 5-5; 6-1
 Format 5-2; 6-2
 Global library file 5-1
 Interactive-only 8-1
 Interactive status 8-1,12
 Job control 7-1
 Job processing 8-1,13
 Label Field 5-2
 Limit 3-10
 Load/dump memory 3-12; 11-1
 Local file 5-1
 Loop 6-1
 Permanent file 10-1
 Prefix character 5-2

Primary file editing 5-2; 8-1,26
 Processing 5-1
 Product set 5-1
 Protected 3-12
 Skipping 6-1,15,16,17
 Subsystem selection 8-1,6
 Syntax 6-2
 System 5-1
 System utility 15-1
 Tape management 12-1
 Terminal control 8-1,2
 Terminator 5-2
 Command processing flow 5-5
 Command record 3-1; 4-1; C-2
 Commands
 Terminal definition K-1
 COMMENT command 7-3
 COMMENT directive 16-25
 Comments field 16-5,6,15
 COMMON command 10-15
 COMPASS assembler 9-24
 Compilation error 14-1
 Compile C-2
 COMPILE file 15-5,11
 Compiler 1-5; 8-23
 Compiler diagnostic 14-1
 Compiler language 1-5
 Concatenation symbol A-3
 Connect time 3-4
 Constant 6-5
 Continuation line 5-2; 6-2; 12-3
 Continuation lines 4-2
 Control byte 17-10
 Control character A-1; C-2; K-1,3
 Control language 6-2; H-1
 Control point C-2
 Control register 6-6,22,24; 7-9
 Control statement C-2
 Conversion mode 12-7,9,14; F-3
 CONVERT command 9-4
 COPY command 9-6
 COPY directive 16-25
 COPYBF command 9-11
 COPYBR command 9-12
 COPYCF command 9-13
 COPYCR command 9-15
 COPYEI command 9-16
 Copying
 Additional copy specifications 9-18
 Binary mode 9-7
 Block size 9-8
 Coded files 9-13,43
 Coded mode 9-7
 Coded records 9-15
 Copy count 9-8
 Error limit 9-8,51
 File 9-18
 Files 8-36,38,39; 9-6,16,17
 Multifile files 9-10
 COPYING
 Multifiles 9-11
 Copying
 Octal copy 9-52
 Processing options 9-9,10,51
 Records 9-12
 Tapes 9-49
 Termination 9-7,10
 Termination condition 9-50
 Copying files 16-7,17
 Copying records 16-7
 COPYL command 16-7
 COPYLM command 16-7
 COPYSBF command 9-17; F-2
 COPYX command 9-18
 Correcting input lines 17-4
 Correspondence code 8-2
 Correspondence code terminal 9-5
 CPU mode error 6-19; 14-1
 CPU priority 7-38
 CPU (refer to Central processor unit)
 CPU time 7-3,10
 CPU time limit 7-38
 CPUMTR 1-3
 Creation date 12-16; 16-6,15; G-4,8,9
 CRM (refer to CYBER Record Manager)
 Cross-reference listing 15-2
 Cross-references 16-20,32,33
 CSET command 8-4
 CTIME command 7-3
 CTRL/F character 17-7
 CTRL/P character 17-7
 CTRL/T character 17-7
 CYBER Interactive Debug 14-1
 CYBER Loader 1-5,6; 16-32,33; C-2
 CYBER Record Manager (CRM) 1-5,6; 2-2; 9-1;
 16-1; C-2

 D command 8-12
 #DATA 4-3,14
 Data conversion F-1
 DATA directive 4-2,14
 Data format 12-7
 Data formats 12-2
 Database file 2-15
 DATE directive 16-25
 Dayfile 7-4; C-3; D-4
 Account 3-13
 Job 3-13
 DAYFILE command 7-4
 DDP (refer to Distributive data path)
 Deadlock 7-28,29
 Deadstart C-3
 Debug mode 3-12; 12-5
 Debug utility 14-1
 Debugging 14-1

Default C-3
 Default subsystem 8-11
 Deferred batch jobs 7-40
 Deferred routing 9-29,37,39
 DEFINE command 10-15
 DELETE command 8-31
 DELETE directive 16-26
 Deleting input lines 17-5
 Density identifier 7-30
 Detach command 8-12
 Detached job 8-12,19; C-3
 Device C-3
 Auxiliary 2-16
 Family 2-15
 Mass storage 2-15
 Permanent file 2-15
 Device type 6-12; 9-1; 12-6; 0-1
 Device type mnemonics 6-12; 7-28; 9-2;
 10-4; 0-1
 Diagnostic messages B-1
 Diagnostics
 Compiler 14-1
 DIAL command 8-15
 Direct access 10-15
 Direct access file 2-14; C-3
 Direct access file type 2-11
 Direct access files 10-6
 * directive 4-2,17
 Directive C-3
 Directives
 GTR 16-12
 LIBEDIT 16-20
 Procedure 4-2
 PROFILE 15-9
 SUBMIT 7-41
 Directory
 File 16-1
 Random Access 16-11
 Random access 16-17,24
 Disk storage subsystems 1-4; 0-1
 Display code C-3
 DISPLAY command 6-14
 Disposition 2-11; 9-37
 Disposition code 9-37; C-3
 Distributive data path (DDP) 1-4; 0-1
 DMB command 11-1
 DMB output 11-3
 DMD command 11-4
 DMDECS command 11-4
 DMP command 11-5
 DMPECS command 11-6
 DOCUMENT command 9-19
 DROP command 7-6
 Dropping jobs 7-6
 DT function 6-12
 Dual processor 3-13
 Dump 9-32,52
 Binary 11-1
 Central memory 11-1,4,5,8,9
 Checkpoint 13-1
 Display code 11-4
 Exchange package 11-1,4,5; 14-1
 Extended memory 11-1,4,6
 Dump memory commands 11-1
 DUP command 8-32

 E command 8-13
 E format J-2
 EBCDIC C-3
 EBCDIC character code A-11
 EC directive 7-44
 Echoplex mode K-7
 EDIT command 15-1
 Edit file 8-26
 Editing files 15-1,14; 16-17
 Editing program libraries 15-3,6,10
 Editing responses 8-28
 EEA (refer to Error exit address)
 ELSE command 6-15
 EM field 14-3
 Empty PRU/record C-3
 Empty record 9-58
 End-of-file (EOF) 2-3,4; 4-16; 7-41; 9-43;
 C-3; F-2
 End-of-file label G-1,8,10
 End-of-information (EOI) 2-2,3,4; C-3; F-2
 End-of-line F-2; I-1
 End-of-line byte A-2; F-1
 End-of-record (EOR) 2-3,4; 4-16; 7-41;
 9-43; C-4; F-2
 End-of-reel J-3
 End-of-tape (EOT) 12-3; C-4; G-1; J-3
 End-of-volume label G-1,9,10
 ENDHELP directive 4-2,12
 ENDIF command 6-16
 ENDW command 6-16
 ENQUIRE command 7-7
 ENTER command 7-13
 Entry point 16-5,15,33; C-4
 EOF directive 4-2,16
 EOF directive 7-41
 EOF (refer to End-of-file)
 EOFn label G-1,10
 EOF1 label 2-8; G-8
 EOI (refer to End-of-information)
 EOR directive 4-2,16; 7-41
 EOR (refer to End-of-record)
 EOT (refer to End-of-tape)
 EOvn label G-1,10
 EOv1 label 2-8; G-9
 Equipment status table (EST) 3-14
 Error blocks 9-9
 Error codes 6-8
 Error control 3-11
 Error exit address (EEA) 14-4

Error exit mode 6-19
 Error flag 6-6,22; C-4
 Error messages B-1
 Error processing 3-11; 5-7; 6-19; 14-5
 Error types 6-8
 Escape code A-1
 Escape code character 7-40,44
 EST ordinal 9-2; 12-6; C-4; D-4
 EST (refer to Equipment status table)
 Even parity 2-7; K-6
 EVICT command 9-21
 Exchange package 3-11; 14-1; C-4
 Exchange package dump 11-1,4,5; 14-1
 Executable object code C-4
 EXECUTE command 8-9
 Execute subsystem 8-9
 Execution time 1-2
 EXIT command 5-7; 6-17
 Exit mode 14-3
 Exit processing 5-7; 6-8; 9-42
 Expiration date 12-16; G-4,8,9
 Expression 6-2,4
 Extended instruction stack purging C-4
 Extended memory 1-3; C-4; O-1
 Extended memory dump 11-1,4,6
 Extended memory field length (FLE) 3-8;
 14-4
 Extended memory preservation 7-27
 Extended memory reference address (RAE)
 14-4
 Extended stack purging 7-23
 External characteristics 9-39
 External reference 16-33

F tape format 2-9
 Family 2-15; D-2
 Family device C-5
 Family devices 2-15
 Family name 7-46; C-5
 FCOPY command 9-22
 FET (refer to File environment table)
 Field length (FL) 1-2; 3-8; 7-15,23,33;
 14-1; C-5
 #FILE 4-3
 #File 4-15
 File
 Access 10-6,18,19
 Access category 10-2
 Access methods 16-1
 Access mode 2-14; 10-7
 FILE
 Accsss 8-16
 File
 Appending 10-6
 Appending lines 8-38
 Appending records 16-11
 Assignment to device 9-1

Attribute 6-11
 Backspacing 9-3
 Backup copy 2-16; 10-2
 Binary 16-14
 Block count G-4,8,9
 Card 2-6
 Catalog 2-13; 10-9
 Cataloging 16-4
 Changing lines 8-29
 Checkpoint 9-32; 13-1
 Code set conversion 9-22
 Coded 9-13,43
 Copying 8-36,38; 9-6,16,17,18,52;
 16-7,17
 Copying lines 8-39
 Creation date G-4,8,9
 Database 2-15
 Definition 2-1; C-5
 Deleting lines 8-31
 Device residence 2-7,15,16
 Direct access 2-13,14; 10-6,15
 Directory 16-1
 Duplicating lines 8-32
 Editing 8-26; 15-1,3,10,14
 Expiration date G-4,8,9
 External characteristics 9-39
 External documentation 9-19
 File space 9-21,35
 Formats 9-6
 Generation number G-4,8,9
 Identifier G-4,8,9
 Indirect access 2-14
 Inserting lines 8-29,36
 Internal characteristics 9-40
 Internal documentation 9-19
 Itemizing record information 16-14
 Library 2-13,16; 10-15; 16-1,32
 Listing lines 8-18
 Listing lines selectively 8-33
 Local 2-12
 Locked 9-24,54
 Logical 2-1,2
 Logical structure
 CYBER Record Manager 2-3
 Diagram of 2-2
 NOS 2-3
 Magnetic tape 2-7
 Mass storage 2-6,15
 Moving lines 8-35
 Multifile 2-3,4
 Multifile set 2-8
 Name 2-1
 Organization 2-3
 Packing 9-30
 Password 10-4
 Permanent 2-13; 10-9,24,25
 Permission mode 10-5,21
 Physical structure 2-1,4

Positioning 9-3,36,46,47,48
 Preferred residence 10-4
 Primary 2-13; 9-29,30
 Private 10-3
 Profile 15-8
 Program library 2-17
 Public 10-3
 Purging 10-22,23
 Queued 7-27; 9-29,37
 Read-only 2-13
 Reformatting 9-25
 Releasing 9-4,21,35,46,53
 Replacement 10-24
 Resequencing 9-34
 Residence 10-4,20
 Retrieval 8-16; 10-6,18,19
 Rewinding 9-36
 Routing 9-37
 Section number G-4,8,9
 Semi-private 10-3
 Sequence number G-4,8,9
 Set identification G-4,8,9
 Sorting 8-18; 9-34,48
 Space 10-17
 Status 7-8; 9-46
 Storing 10-24,25
 Structure 2-1; 16-2
 Structure reporting 9-43
 Subdivisions 2-1,2,3
 Subsystem association 8-8,11; 10-5
 Temporary 2-12
 Terminology 2-3
 Types 2-11
 User library 2-17
 Verification 16-31,34
 Verifying 9-54
 File accessibility character 12-9,15
 File attribute 6-11
 File attribute mnemonics 6-11
 FILE command 2-7
 FILE directive 16-26
 File environment table (FET) C-5
 FILE function 6-10
 File header label G-1,4,10
 File information table (FIT) C-5
 File management commands 9-1
 File mark 2-4; 9-30,43,58
 File name table (FNT) 5-5
 File residence 2-15
 File sequence number 12-15,19
 File set C-5
 File Space 9-53
 File/Status 7-18
 FIT (refer to File information table)
 FL (refer to Central memory field length)
 Flag C-5
 FLE (refer to Extended memory field length)
 FNT (refer to File name table)

Format channel selection I-3
 Forms code 9-40
 FORTRAN C-5
 FORTRAN command 8-10
 FORTRAN subsystem 8-10
 Frame C-6
 FTNTS C-6
 FTNTS command 8-11
 FTNTS subsystem 8-11
 Full-track 0-1
 Function 6-10

 Generation C-6
 Generation number 12-15; G-4,8,9
 Generation version number 12-16
 GET command 10-18
 Global error flag 6-6,22
 Global library directory 5-5
 Global library set C-6
 GO command 7-13
 GOODBYE command 3-3; 8-15
 GOTO command H-4
 Graphic C-6
 Graphic character A-1
 GTR command 16-11

 Half-track 0-1
 Hardware 1-1
 Hardware-detected error 14-5
 HASP protocol terminal 17-1; K-3
 Hazeltine 2000 terminal 17-1; K-3
 HDRn label G-1,10
 HDRl label 2-8; G-4
 Headers 8-4
 HELLO command 3-4; 8-15
 HELP command 8-16
 HELP directive 4-2,11
 Hollerith punch code F-3
 Hollerith punch output F-5
 HTIME command 7-14

 I/O mode 2-9
 I tape format 2-9
 IAF (refer to Interactive Facility)
 IBM 2741 terminal 17-1,9; K-2
 Idle character K-4,5
 IF command H-8
 IFE command 6-17
 IGNORE directive 16-27
 Immediate job status command 8-13
 Implicit remote routing 9-41
 Implicit routing 9-42
 Increment registers 14-4
 Indefinite operand 6-20
 Indirect access file 2-14; C-6

Input conventions 17-2
 Input data E-1
 Input device K-7
 Input disposition 2-11; 9-37
 Input file 2-11; 3-9; 7-17,40; 9-37
 INPUT file 3-9; 13-1
 Input file type 2-11; C-6
 Input line
 Correction 17-4
 Deletion 17-5
 Delimiter 17-3
 Entry before prompt 17-5
 Length 17-3
 Termination 17-4
 Input mode 8-12
 Input prompt 17-2,5
 INSERT directive 16-27
 Instruction stack purging 7-23; C-6
 Interactive Facility (IAF) 1-1; 3-2; 17-1;
 C-6; K-1
 Interactive job 3-1
 Interactive-only commands 8-1
 Interactive origin job 3-5
 Interactive status commands 8-1,12; 17-5
 Internal characteristics 9-40
 Interrecord gap C-6
 Interrupting jobs 17-7
 Interruption character 17-7; K-4
 Interruption sequence 8-24; 17-7,8
 ITEMIZE command 16-14
 ITEMIZE output 16-15
 Itemizing record information 16-14

 Job C-6
 Accumulated time 7-3
 Batch 3-1
 Checkpoint 13-1
 Command record 4-1
 Comment 7-3
 Communication area 5-4
 Completion 3-12
 Control 3-8
 Control registers 7-9
 Dayfile 3-13,14
 Detached 3-5
 Detachment 8-12
 Dropping 7-6
 Entry 3-1; 7-17,40
 Error processing 6-19; 14-5
 Exchange package 14-1
 Execution 3-1
 Exit processing 5-7; 6-8
 Field length 3-8; 7-15,23,33
 Flow 3-1
 Initiation 3-1
 Input file 3-9
 Interactive 3-1
 Interruption sequence 17-7
 Name 3-6
 Origin type 3-5; 6-9
 Output 3-14; D-1
 Output processing 2-12; 7-17,36,40
 Ownership 7-6
 Priority 3-7; 7-38
 Priority level 7-15
 Recoverable 3-3
 Recovery 8-19
 Resource demand 7-8,28
 Resource usage 3-13; 7-9,10,35,37,39
 Restart 13-1
 Rolled out 3-11
 Scheduling 3-7
 Scheduling priority 3-5
 Security 3-12
 Service class 3-5
 SRU limit 3-10
 Status 7-7; 8-12,13,20
 Status of files 7-8,18
 Summation of activity 3-13
 Suspension 7-36,38; 17-8
 Termination 3-1; 5-7; 7-36; 9-29
 Termination sequence 17-7
 Time limit 3-10; 7-15,38
 Timeout period 3-4; 8-19
 Type 3-1
 Job card 7-15
 Job command 7-15
 Job control
 Command limit 3-10
 Error 3-11
 Field length 3-8
 Input file 3-9
 Rollout 3-11
 SRU limit 3-10
 Time limit 3-10
 Job control commands 7-1
 JOB directive 7-41
 Job entry 3-1; 7-40
 Job execution 3-1
 Job field length 1-2
 Job file 3-1
 Job flow 3-1
 Job initiation 3-1
 Job origin D-2
 Job processing commands 8-1
 Job scheduler 7-34
 Job sequence name (JSN) 3-6; C-6; D-2
 Job step 3-10; C-6
 Job termination 3-1
 Job type 3-1
 JSN (refer to Job sequence name)

KCL control language 6-2; H-1
Keypunch conversion mode 2-6; 7-16; F-3
Keyword 5-4
KRONREF command 15-2

L tape format 2-9
Label 12-19; C-7
 ANSI G-1
 Creation 12-11
 Identifier G-10,11
 Magnetic tape 2-8; 12-8,11; G-1
 Number G-10,11
 Optional G-1,10
 Required G-1
 User G-11
LABEL command 12-11
Label field 5-2
Label standard level 12-10
Labeled tape 12-7,13
LBC command 11-7
LDI command 7-17
LENGTH command 7-18
Level number C-7
LGO command 14-6
LGO file 13-1; 16-11,18,26
LIB command 8-16
LIBEDIT command 16-17
LIBGEN command 16-32
LIBGEN directive 16-28
LIBLOAD command 2-17
Library 2-16; 16-1; C-7
LIBRARY command 2-17
Library directory 16-20
Library editing 16-17
Library file 2-13; 10-15; C-7
Library file type 2-11
Library maintenance 16-1
Library processing examples 16-36
Library record type 16-1,2
LIBRARY user name 2-16; 8-16
LIMITS command 7-19
Line C-7
 Length 17-3
 Logical 17-3
 Physical 17-3
Line feed L-2
Line feed idle count K-5
Line feed key 17-3; L-1
Line number 7-41; 8-2,26; 9-34,48
Line number overlap 8-28
Line printer 1-4
Line printer carriage control I-1
Line printer usage A-3
Line range 8-26
Lines printed, total 3-14
LIST command 8-18,33
LIST directive 16-28

Listing labels 12-19
LISTLB command 12-19
LIST80 command 9-24
Literal 6-5; 12-3
Literal card input F-3
Load/dump memory commands 3-12; 11-1
LOAD file 7-17
Load map 14-1
Load memory commands 11-1
Load point C-7
Loader information 7-9
Loading (also refer to CYBER Loader)
 Binary data 11-7
 Central memory 1-6; 11-8
 Octal lines 11-7
LOC command 11-7
Local applications 7-22
Local batch job 3-5
Local file 2-12; C-7
Local file name C-7
Local file name table (FTN) C-7
Local file type 2-11
Local operator (LOP) 7-21; 17-3; C-7
Local operator message K-7
LOCK command 9-24
Locked file C-7
Logical device 2-6; C-7
Logical file C-7
Logical line 17-3
Logical operator 6-4
Logical record C-7
Logical track 2-6
Login 3-2; 8-12,15; C-7
LOGIN command 3-4; 8-18
Logout 3-3
LOGOUT command 3-3; 8-18
Logout display 3-3; 8-14
Loop 6-1
LOP (refer to Local operator)
L072 9-25

MA (refer to Monitor address)
MACHINE command 7-23
Machine identifier 6-7; D-4
Machine language 1-5
Macro C-7
Magnetic set 12-15
Magnetic tape 1-4
 Accessability 12-9,10,15
 ANSI standards 12-2
 Assignment 12-1,5,11,21
 Block size 12-7,13
 Checkpoint file 12-8,14
 Command rules 12-2
 Commands 12-1
 Conversion mode 12-7,9,14
 Creating files 12-5

Creation date 12-16
 Data formats 2-9; 12-7,13
 Density 2-7; 12-6,9,13
 Density specification 12-2
 Device type 12-6
 Error processing 12-3
 Expiration date 12-16
 File 2-7
 File sequence number 12-15,19
 File set configurations 2-8
 Generation number 12-15
 Generation version number 12-16
 Label 2-8; 12-8,19; G-1
 Label standard level 12-10
 Labeled 12-7,13
 Listing labels 12-19
 Load point 12-2
 Multifile file 12-1
 Multifile set 2-8; 12-15,17
 Multifile set positioning 12-18
 Multivolume set 12-25
 Noise size 12-8,14
 Parity 2-7
 Parity Error 12-3
 Processing option 12-3
 Recording mode 2-7
 Section number 12-15
 Set identifier 12-15,19
 Tape mark 2-8
 Unlabeled 12-7,13
 Volume association 12-24
 Write ring 12-4
 Writing labels 12-8
 Magnetic tape activity 3-13
 Magnetic tape units 7-30
 Map
 Load 14-1
 Symbolic reference 14-1
 MAP (refer to Matrix Array Processor)
 Mass storage C-7
 Mass storage activity 3-13; D-4
 Mass storage device 1-4; 2-15; O-1
 Mass storage device statistics O-1
 Mass Storage Facility (MSF) 2-16
 Mass storage file 2-6
 Master device C-7
 Master user 15-8
 Matrix Array Processor (MAP) 3-7,13
 Maximum line length 17-3
 Memorex 1240 terminal 9-5
 MERGE file 15-12
 Message command K-7
 MFL command 7-23
 MODE command 6-19
 Modification decks 15-6
 MODIFY command 15-3
 Modify mode 2-14
 Monitor address (MA) 14-4
 MOVE command 8-35
 MSF (refer to Mass Storage Facility)
 Multifile file 2-3,4; 12-1; C-8
 Multifile set 12-17; C-8
 Multimainframe configuration 1-3
 Multivolume set 12-25
 M33 TTY 17-1; K-2
 M35 TTY 17-1; K-2
 M37 TTY 17-1; K-2
 M38 TTY 17-1; K-2
 M40 terminal 17-1; K-3
 NAM (refer to Network Access Method)
 NAME directive 16-31
 Nested procedure M-9
 Network 3-4; 17-1,3; C-8; K-1
 Network Access Method (NAM) 1-1
 Network applications 7-21
 Network/Local operator (NOPLP) 7-21
 Network messages 17-3
 Network operator (NOP) 7-21
 Network processing modes 1-1
 Network processing unit 1-4
 NEW command 9-29
 NEW directive 16-28
 New line L-2
 NEWPL file 15-12
 No abort option 10-3
 No-auto-drop status 9-46
 NOEXIT command 5-7; 6-21
 NOINS directive 16-28
 Noise C-8
 Noise blocks 9-9
 Noise size 2-9; 12-8,14
 Nonallocatable device C-8
 NOP (refer to Network operator)
 NOPACK directive 7-42
 NOPLP (refer to Network/Local operator)
 NOREP directive 16-29
 NORERUN command 7-24
 NOREW directive 16-29
 NORMAL command 8-4
 Normal mode 8-2,4; 9-5; E-1
 NOSEQ directive 7-42
 NOSORT command 8-18
 NOSTEXT file 15-2
 NOTE command 7-24
 NOTRANS directive 7-43
 NPC= entry point 5-2
 NPL file 15-4,6
 NULL command 8-11
 NUM function 6-13
 Number sign, procedures 4-17
 Numeric string 6-5

Object code 8-23; C-8
 Object code program 8-9
 Obsolete tape formats J-1
 Odd parity 2-7; K-6
 OFFSW command 7-25
 OLD command 10-19
 OLD directive 16-29
 OLDPL file 15-13
 On-line assistance 8-16
 ONEXIT command 5-7; 6-21
 ONSW command 7-25
 Operand 6-2,4
 Operand registers 14-4
 Operating system (OS) 1-5
 Operating system version D-2
 Operator 6-2
 OPL file 15-2,6
 OPL record type 16-3
 OPLC record type 16-3
 OPLD directory 16-20,24,33
 OPLD record type 16-3
 OPLEDIT command 15-6
 Optional label G-1,10
 Order-dependent C-8
 Order-independent C-8
 Order of evaluation 6-4
 Origin type 6-9; C-8
 OS (refer to Operating system)
 OUT command 9-29
 Output conventions 17-2
 Output data E-2
 Output device K-7
 OUTPUT file 2-11; 3-14; 9-29,38; 13-1
 Output line
 Aborting 17-6
 Length 17-3
 Output processing 2-12; 7-17,36,40
 Overload condition 17-3
 OVL record type 16-3

 P field 14-1
 PACK command 9-30
 PACK directive 7-42
 Pack name 10-3,20
 Packet switching network (PSN) 3-4; 17-1;
 C-8
 PACKNAM command 10-20
 Page boundary 17-6,7
 Page length K-5
 Page wait 17-6; K-5
 Page width K-5
 Paper tape
 Control characters N-2
 Corrections N-4
 Input lines N-2
 Operation N-1
 Punch N-1
 Punching off-line N-3
 Punching on-line N-4
 Reader N-1
 Paper tape mode N-3
 Parameter C-8
 Keyword 5-4
 Operating system format 5-2
 Order-dependent 5-4
 Order-independent 5-4
 Product set format 5-2
 Parameter matching mode 4-27; M-2
 Parameter prompting 4-2
 Parity 2-7; C-8
 Parity errors 9-9,51; 12-3
 Parity selection K-6
 Partition 2-3
 Passive procedure C-8; M-1
 PASSWORD command 7-26
 Password 7-26,46; 10-4; C-8
 Pause bit 7-2,13,26
 PAUSE command 7-26
 PBC command 11-8
 Peripheral equipment 1-4
 Peripheral processor library 5-5
 Peripheral processor library directory C-9
 Peripheral processor (PP) 1-4; C-9
 Permanent file 2-13; 10-9,24,25; C-9
 Permanent file activity 3-13; D-4
 Permanent file commands 10-1
 Permanent file device C-9
 Permanent file family C-9
 Permission mode 10-5,21
 PERMIT command 10-21
 PFC array 9-41; I-3,5
 PFC printer I-1
 Physical line 17-3
 Physical record unit (PRU) 2-4; C-9; O-1
 Plot disposition 2-11; 9-38
 Positioning files 9-3,36,46,47,48
 Positioning multifile set 12-18
 Post radix C-9
 PP configurations 1-4
 PP record type 16-3
 PP (refer to Peripheral processor)
 PPU record type 16-3
 Preferred residence 10-4
 Prefix character C-9
 Prefix table 16-2,15,25,34
 Presetting memory 7-35
 PRIMARY command 9-30
 Primary file 2-13; 9-29; C-9
 Primary file editing commands 5-2; 8-1
 General description 8-26
 Parameter format 8-26
 Primary file type 2-11
 Print disposition 2-11; 9-38
 Print trains A-3
 Printer carriage control I-1

Priority 3-7
Priority level 7-15
Private auxiliary device C-9
Private file 10-3
Privileged nnetwork users 7-21
PROC directive 4-2; M-1
PROC record type 16-3
Procedure C-9
 Body 4-1
 Call 4-1,19
Call-by-name 4-20
Comments 4-17
Concatenation symbol 4-17
Data file 4-14,16
Definition 4-1
Directives 4-2
Header 4-1
Help information 4-23
Inhibiting substitution 4-17
Interactive processing 4-23
Keywords 4-17
Nesting level 6-7
Parameter descriptions 4-23
Parameter matching mode 4-27; M-2
Parameter prompting 4-2,23
Parameter substitution 4-17; M-2
Parameters 4-17
Passive M-1
Structure 4-1
Termination 4-25
Procedure call 4-1,19
Procedure file C-9
PROCFIL file 4-20
Products 1-5,6; 3-4
PROFILB file 15-8
PROFILE command 15-8
Program address (P) 14-1
Program library 2-17; 15-3,10
PROGRAM statement 8-23
Program status designator (PSD) 14-3
Programmable format control (PFC) 9-41
Project number 7-2
Project profile file 15-8
Prologue 7-20,45; C-9
Prompts 17-5
PROTECT command 7-27
Protected command 3-12
PRU device C-9
PRU (refer to Physical record unit)
PSD field 14-3
PSN (refer to Packet switching network)
Public auxiliary device C-10
Public file 10-3
Punch card format F-1
Punch code 7-16
Punch disposition 2-11; 9-38
PUNCH file 2-11; 3-14; 9-29,38; 13-1; F-6
Punch file format F-5
PUNCHB file 2-11; 3-14; 9-29,38; 11-8;
 13-1; F-6
PURGALL command 10-22
PURGE command 10-23
Purging files 10-22,23
P8 file 2-11; 3-14; 9-29,38; 13-1; F-7

QGET command 7-27
Question mark, procedures 4-21,23
Queue 2-11
Queue priority C-10
Queued file 2-11; 7-27; 9-37; C-10
Queued file disposition 2-11; 9-37
Queued file type 2-11
Queued files 9-29

RA field 14-1
RA (refer to Reference address)
RAE (refer to Extended memory reference
 address)
Random access 16-1; C-10
Random access directory 16-11,17
Random access file C-10
RBF (refer to Remote Batch Facility)
READ command 8-36
READ directive 7-43
Read-only file 2-13
Read-only permission C-10
Reading central memory 11-8,9
Real-time clock 7-34
Real-time processing option 10-8
Record C-10
 Checksum 16-6,15
 Comments field 16-5,6,15
 Copying 16-7
 Creation date 16-6,15
 File subdivision 2-3
 Group identifier 16-22
 Length 16-6,15
 Name 16-5,15
 Number 16-5
 Reference point identifier 16-22
 S-type 16-1
 Type 2-3; 16-3,15,31
 U-type 16-1
Recording mode
 Binary 2-7
 Coded 2-7
RECOVER command 8-19
Recoverable job 3-3; 8-19
Recovery C-10
Recovery processing 8-19
Reel number G-2
Reference address (RA) 1-2; 7-2; 14-1
Reference address (RA and RAE) C-10
Reformatting 9-25

Reformatting directives 7-40
 REL record type 16-3
 Relational operator 6-3
 Releasing file space 9-21
 Releasing files 9-4,35,46,53
 Remote Batch Facility (RBF) 1-1; 3-1
 Remote batch job 3-5
 Removable packs 7-33
 RENAME command 9-31
 RENAME directive 16-29
 REPLACE command 10-24
 REPLACE directive 16-30
 Replacement file 16-18,26
 Replacing files 10-24
 Replacement file 16-7
 REQUEST command 9-32; 12-21
 Required label G-1
 RERUN command 7-28
 RESEQ command 9-34
 Reservation block 2-7
 RESOURC command 7-28
 Resource commitment 7-29
 Resource demand 7-8,28; 9-36
 Resource limits 7-20
 Resource overcommitment 7-31
 Resource usage 3-13; 7-9,10,35,37,39; D-4
 Restart 13-1
 RESTART command 13-2
 Retrieving files 8-16; 10-6,18,19
 RETURN command 9-35
 Reverse slant (\) 4-3,27
 REVERT command 4-25
 REWIND command 9-36
 REWIND directive 7-43; 16-30
 Rewinding files 9-36
 RFL command 7-33
 Rolled out C-10
 ROLLOUT command 7-34
 Rollout control 3-11
 ROUTE command 9-37
 Routing files 9-37
 RTIME command 7-34
 Rubout N-2
 Rubout characters C-10
 RUN command 8-23

S command 8-13
 S tape format 2-9
 S-type record 9-1; 16-1
 SATISFY command 2-17
 SAVE command 10-25
 SC directive 5-2
 Scheduler 3-7; 7-34
 Scheduling priority 2-12; 3-5; C-10
 SCOPY command 9-43
 Scratch files 2-1
 Section 2-3

Section number 12-15; G-4,8,9
 Security 3-12; 7-26
 Security count 7-47; 9-42; C-11
 Semi-private file 10-3
 Sense switch 6-10; 7-25,45
 Separator 5-2
 SEQ directive 7-41
 Sequence number 9-34; C-11; G-4,8,9
 Sequential access 16-1; C-11
 Sequential (SQ) file C-11
 Service class 2-12; 3-5; C-11; D-2
 SET command 6-22
 Set identification G-4,8,9
 Set identifier 12-15,19
 SETASL command 7-35
 SETCORE command 7-35
 SETFS command 9-46
 SETJOB command 7-36
 SETJSL command 7-37
 SETPR command 7-38
 SETTL command 7-38
 Short PRU C-11
 SI tape format 2-9
 SKIPEI command 9-46
 SKIPF command 9-47
 SKIPFB command 9-47
 Skipped commands 6-6,22
 Skipping commands 6-1,15,16,17
 SKIPR command 9-48
 Slant (/) 4-3,27; 5-2
 Software 1-5
 SORT command 9-48
 Sort flag 8-18
 Sorting files 8-18; 9-34,48
 Source code 9-34; C-11
 Source code program 8-9
 SOURCE file 15-5,13
 Source program 8-23
 Spacing code 9-41
 Special editing mode 8-2; C-11; K-6
 SRU (refer to System resource unit)
 Stack purging 7-23
 Standard labeled tape C-11
 STIME command 7-39
 Storing files 10-24,25
 String C-11
 Structure reporting 9-43
 SUBMIT command 7-40
 Submit file 7-40
 Subsystem 8-6; 10-5
 Subsystem flag C-11
 Subsystem selection commands 8-1,6
 Subsystem selection parameter 8-8
 SUMMARY command 7-44
 Suspending jobs 17-8
 Suspending output 17-6
 SWITCH command 7-45
 Symbolic name 6-6,22

Symbolic reference map 14-1
 Synchronous terminals 17-6
 System C-11
 System code G-4,8,9
 System console 3-5
 System file C-11
 System library 5-2
 System monitor 1-3; 14-1
 System origin job 3-5
 System prologue 7-20
 System resource unit (SRU) 3-7,10,13;
 7-10,35,37,39; C-11; D-4
 System symbols 15-2
 System text 15-2
 System utility commands 15-1
 System version number 6-7
 SYSTEXT file 15-3

TAF (refer to Transaction Facility)
 Tape assignment 12-5,11,21
 Tape density 12-9
 Tape formats 2-9; 12-2,7,13; C-11; J-1
 Tape management 12-1
 Tape marks 2-8; 12-2; C-11
 Tape owner identification G-2
 Tape (refer to Magnetic tape)
 TAPEn file 11-9; 12-19
 TCCOPY command 9-49
 TDUMP command 9-52
 Tektronix 4014 terminal 17-1; K-3
 Teletypewriter (TTY) 17-1
 Temporary file C-11
 Terminal
 Character conversion E-1
 Characteristics 8-5; K-1
 Class 17-1; K-2
 Control 8-5; 17-10
 Control character K-1
 Input 17-1,2
 Name 17-1; C-12
 Output 17-1,6
 Session 3-3
 Type K-2
 Terminal control commands 8-1,2
 Terminal definition commands 8-5; 17-5; K-1
 Terminal definition parameter K-2
 Terminal identifier (TID) 2-12
 Terminating input lines 17-4
 Termination character 17-7; K-4
 Termination sequence 8-24; 17-7
 Terminator 5-2
 TEXT command 8-24
 Text editors 15-1,14
 Text mode 8-12,24
 TEXT record type 16-3
 TID (refer to Terminal identifier)
 Time limit 3-10; 7-15,38

Time slice 3-11; C-12
 Timed/Event rollout C-12
 TIMEOUT command 8-5
 Timeout period 3-4; 8-19; K-6
 Timeout status 8-5
 Tracks 2-7
 TRANS directive 7-42
 Transaction Facility (TAF) i-1
 Transmission block K-2,3
 Transmission key L-1
 Transparent input mode C-12; K-6
 Transparent input mode delimiter K-6
 Transparent submit mode 7-40,42; C-12
 TRMDEF command 8-5
 TTY (refer to Teletypewriter)
 TYPE directive 16-31

U-type record 9-1; 16-1
 UCF record type 16-4
 UHLA G-11
 UHLA label G-1
 UJN (refer to User job name)
 ULIB directory 16-32,33
 ULIB record type 16-3
 Underflow error mode 6-20
 Underline character 4-17
 Unit assignments 7-33
 Unlabeled tape 12-7,13
 UNLOAD command 9-53
 UNLOCK command 9-54
 Unprintable characters A-3
 UPDATE command 15-10
 UPLx record type 16-4
 UPROC command 7-45
 USECPU command 7-46
 User break 8-24
 User break processing 8-12; 17-5
 User break 1 sequence 17-7; C-12; K-4
 User break 2 sequence 17-7; C-12; K-4
 USER command 7-46
 USER directive 7-41
 User header label G-1
 User index 7-20; C-12
 User index hash 3-6; 7-7; C-12; D-2
 User job name (UJN) 3-6; 7-15; C-12; D-2
 User label G-11
 User library 2-17; 16-32
 User name 7-46; C-12
 User number C-12
 User permissions 7-21
 User program 1-5
 User prologue 7-3,20,45
 User trailer label G-1
 User validation 7-19,46
 User volume label G-1
 UTLA G-11
 UTLA label G-1

UVLa G-11
UVLa label G-1

Validation 3-6
Validation file C-12
Validation limits 7-19
Verifying files 9-54; 16-31,34
Version number 6-7
VFYLIB command 16-34
VFYLIB directive 16-31
Volume C-13
Volume accessibility character 12-10
Volume association 12-24
Volume header label G-1,2
Volume serial number (VSN) 12-1,6,9,12,24;
C-13; G-2
VOL1 label 2-8; 12-1; G-1,2
VSN command 12-24
VSN (refer to Volume serial number)

Wait disposition 2-11; 9-38
Wait-if-busy option 10-5
WBR command 11-9
WHATJSN command 8-24
Word C-13
WRITE command 8-38
Write interlock 2-15; 9-24,54; C-13
Write mode 2-14
Write ring 12-4; C-13
WRITEF command 9-58
WRITEN command 8-39
WRITER command 9-58
Writing labels 12-8

X command 8-25
X format J-3
X registers 14-4
XEDIT command 15-14

YANK status 16-6,15

Zero byte terminator C-13
Zero length PRU C-13
Zero parity K-6
ZZCCLxx file 4-14
ZZZDUMP file 3-11; 11-4,5,6

ZZZZDMB file 11-1,3
ZZZZZC0 file 9-46
ZZZZZC1 file 9-46
ZZZZZC2 file 9-46
ZZZZZLD file 9-46; 13-1
ZZZZZxx files 13-1

026 mode 7-16; 9-29
026 punch code A-3; F-3
029 mode 7-16; 9-29
029 punch code A-3; F-3

12-bit ASCII code 9-22

200 User Terminal 17-1; K-3

5/7/9 multipunch F-3,5
580 PFC printer 9-41; I-5

6/12 display code 8-2; 9-22
6/7/8/9 multipunch 2-6; C-13; F-2
6/7/9 multipunch 2-6; C-13; F-2
6PP record type 16-4
63-character set A-1
64-character set A-1

7/8/9 multipunch 2-6; C-13; F-2
7/9 multipunch F-5
7PP record type 16-4
77 table (refer to Prefix table) 16-2
7700 table (refer to Prefix table) 16-2

95-character set A-1

/ 4-3,27
4-17
_ 4-17
? 4-21,23
\ 4-3,27

COMMENT SHEET

MANUAL TITLE: CDC NOS Version 2 Reference Set, Volume 3, System Commands

PUBLICATION NO.: 60459680

REVISION: A

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).

Please Reply

No Reply Necessary

CUT ALONG LINE

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

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

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



CUT ALONG LINE

OLD

FOLD



CONTROL DATA CORPORATION