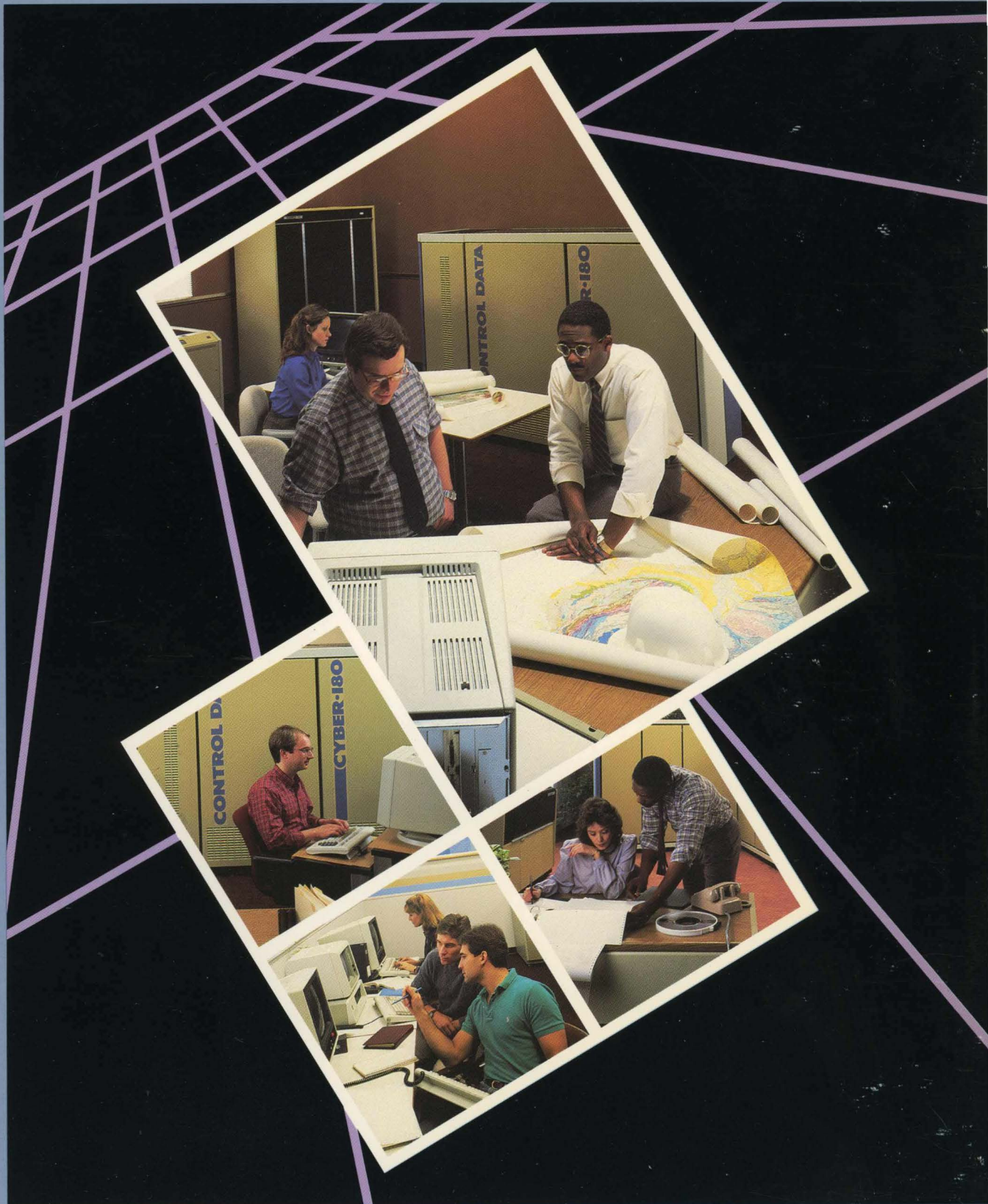


NOS 2 Reference Set Volume 3 System Commands


CONTROL
DATA



60459680

PROCEDURE DIRECTIVES

<u>Directive</u>	<u>Page</u>
.CC	4-33
.CORRECT	4-23
.DATA	4-34.2
.ELSE	4-38
.ENDHELP	4-24
.ENDIF	4-39
.ENTER	4-24
.EOF (or .EOP)	4-38
.EOR (or .EOS)	4-38
.EX	4-39
.EXPAND	4-33
.Fn	4-24.1
.F7	4-24.2
.HELP	4-25
.IC	4-34
.IF (or .IFE)	4-40
.NOCLR	4-31
.NOTE	4-31
.PAGE	4-32
.PROC	4-11, L-1
.PROMPT	4-32
.SET	4-34
.*	4-40.1



**NOS VERSION 2
REFERENCE SET**

Volume 3

SYSTEM COMMANDS

**CDC® COMPUTER SYSTEMS:
CYBER 180
CYBER 170
CYBER 70
MODELS 71, 72, 73, 74
6000**

REVISION RECORD

REVISION	DESCRIPTION
A (04-26-82)	Manual released. This manual reflects NOS Version 2 at PSR level 562. This manual documents parameter-prompting procedures and the following new commands: ALTER, CFO, DELETE, Detach, DMB, DROP, DUP, GO, LIBTASK, MOVE, PAUSE, QGET, READ, SCOPY, SETFS, SETJOB, UPROC, WHATJSN, WRITE, WRITEN.
B (01-27-83)	This manual reflects NOS 2.1 at PSR level 580. This revision incorporates the MFLINK, MFQUEUE, SHELL, and new terminal definition commands; menu-generating procedures and new procedure directives; support of 885-42 disk drives; as well as numerous usability changes. Because of extensive changes to this manual, change bars and dots are not used. This manual obsoletes all previous editions.
C (10-11-83)	This manual reflects NOS 2.2 at PSR level 596. This revision incorporates the following features: enhanced system security; service class assignment by users; a multihost network; full screen editing; screen mode input for procedures and other applications; an expansion of online help, including online manuals; equipment status table expansion; project prologues and epilogues; SCOPE 2 station; and the Interactive Transfer Facility (ITF). Because of extensive changes to this manual, change bars and dots are not used. This revision obsoletes all previous editions.
D (10-05-84)	This manual reflects NOS 2.3 at PSR level 617. This revision incorporates the following features: 834 disk subsystem support; 639 magnetic tape unit support; PSU printer support; FSE enhancements; NOS procedure enhancements, and the following new system commands: BLOCK, EFFECT, ERRMSC, LISTLID, RECLAIM, REDO, SHOW, and TDU. This manual obsoletes all previous editions.
E (03-28-85)	This manual reflects NOS 2.4.1 at PSR level 630. This revision documents the KEY command and the following new features: support of CYBER 180 Models 840, 850, and 860; 895 disk subsystem support; 5870 Non-Impact Printer support; NOS/VE dual-state support; and NOS procedure enhancements.
F (12-16-85)	This manual reflects NOS 2.4.3 at PSR level 647. This revision documents the APPSW command, the RC terminal definition command, and the following features: support of CYBER 180 model 990; support of 836 disk subsystem; enhancements to 533/536 printer support; support of CDCNET version 1; disk error recovery enhancements (ENQUIRE output); AUTO command internal 0003 byte processing.
G (09-30-86)	This manual reflects NOS 2.5.1 at PSR level 664. This revision documents the XMODEM and DROPDS command and the following features: support of print train images, security enhancements, and miscellaneous technical corrections.
H (04-23-87)	This manual reflects NOS 2.5.2 at PSR level 678. This revision documents support for the Televideo terminal and the 585 printer, new user messages related to enhancements to tape error recovery, changes to several procedure directives, and the addition of the .SET directive and the STR, STRB, and STRD functions.
J (09-23-87)	This manual reflects NOS 2.5.3 at PSR level 688. This revision documents support of magnetic tape as an alternate storage medium for permanent files, the 537 printer, and drops documentation of the Mass Storage Subsystem (MSS).
K (04-05-88)	This manual reflects NOS 2.6.1 at PSR level 700. This revision documents CHARGE enhancements, a new PASSWOR parameter, the LVL function, the .F7 directive, the LIBEDIT directive (*/*), new RECLAIM options, and STORNET, a new type of extended memory.
L (12-08-88)	This manual reflects NOS 2.7.1 at PSR level 716. This revision documents support of CYBER 180 Models 960 and 994, support of the 9853 disk subsystem, enhancements to the CATLIST, PURGALL, TDUMP, PDU, and TDU commands, new RECLAIM options, new permanent file command (FTP), new diagnostic messages, and miscellaneous technical corrections.
Publication No. 60459680	

REVISION LETTERS I, O, Q, S, X AND Z ARE NOT USED.

Address comments concerning this manual to:

Control Data
Technical Publications
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983

© 1982, 1983, 1984, 1985, 1986, 1987, 1988
by Control Data Corporation
All rights reserved
Printed in the United States of America

or use Comment Sheet in the back of this manual.

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	-	3-7	D	4-40.1/4-40.2	H	6-31	K	7-55	D
Commands and Procedure Directives Index	L	3-8	C	4-41	D	7-1	L	7-56	J
Title Page	-	3-9	C	4-42	K	7-2	H	7-57	D
2	L	3-10	C	4-43	G	7-3	K	7-58	D
3	L	3-11	E	4-44	H	7-4	K	7-59	L
4	L	3-12	C	4-45	H	7-5	K	7-60	F
5/6	L	3-13	G	4-46	D	7-6	K	7-61	D
7	L	3-14	G	4-47	E	7-7	D	7-62	D
8	L	3-15	G	4-48	H	7-8	D	7-63	L
9	L	3-16	G	4-49	D	7-9	H	7-64	L
10	H	3-17	G	4-50	D	7-10	D	7-65	H
11	L	3-18	G	4-51	K	7-11	E	7-66	D
12	K	3-19	G	4-52	H	7-12	E	7-67	D
13	K	4-1	D	4-53	D	7-12.1/7-12.2	E	7-68	E
14	K	4-2	D	4-54	H	7-13	D	7-69	K
15/16	K	4-3	J	5-1	D	7-14	D	7-70	J
17	K	4-4	H	5-2	E	7-15	G	7-71	J
18	K	4-5	E	5-3	E	7-16	D	7-72	J
19	K	4-6	J	5-4	E	7-16.1/7-16.2	H	7-73	K
20	L	4-7	J	5-5	K	7-17	H	7-74	K
21	L	4-8	J	5-6	K	7-18	D	7-75	H
22	L	4-9	J	5-7	E	7-19	D	7-76	D
23	L	4-10	F	6-1	D	7-20	D	8-1	L
24	L	4-11	E	6-2	L	7-21	D	8-2	F
1-1	E	4-12	K	6-3	H	7-22	E	8-3	C
1-2	C	4-13	K	6-4	L	7-22.1/7-22.2	E	8-4	F
1-3	J	4-14	K	6-5	H	7-23	F	8-5	J
1-4	L	4-15	K	6-6	L	7-24	L	8-6	F
1-5	G	4-16	H	6-7	H	7-25	H	8-6.1	F
1-6	C	4-17	H	6-8	H	7-26	D	8-6.2	H
2-1	H	4-18	F	6-8.1/6-8.2	L	7-27	D	8-7	G
2-2	C	4-19	K	6-9	L	7-28	D	8-8	G
2-3	C	4-20	K	6-10	L	7-29	D	8-8.1/8-8.2	L
2-4	C	4-20.1/4-20.2	K	6-10.1	L	7-30	D	8-9	G
2-5	E	4-21	H	6-10.2	G	7-31	K	8-10	L
2-6	C	4-22	H	6-11	G	7-32	K	8-11	J
2-7	E	4-22.1	H	6-12	K	7-33	D	8-12	D
2-8	H	4-22.2	G	6-13	K	7-34	D	8-13	D
2-9	J	4-23	H	6-14	K	7-35	F	8-14	F
2-10	J	4-24	H	6-15	L	7-36	E	8-15	D
2-10.1/2-10.2	J	4-24.1/4-24.2	K	6-16	K	7-37	D	8-16	J
2-11	D	4-25	K	6-16.1	K	7-38	D	8-17	K
2-12	E	4-26	G	6-16.2	K	7-39	D	8-18	L
2-13	E	4-27	H	6-16.3	K	7-40	K	8-19	H
2-14	C	4-28	E	6-16.4	L	7-41	K	8-20	G
2-15	J	4-29	H	6-17	K	7-42	K	8-20.1/8-20.2	F
2-16	J	4-30	H	6-18	K	7-43	K	8-21	F
2-17	G	4-31	K	6-19	K	7-44	K	8-22	D
3-1	H	4-32	H	6-20	K	7-45	J	8-23	E
3-2	L	4-33	K	6-21	F	7-46	D	8-24	D
3-3	H	4-34	K	6-22	D	7-47	D	8-25	D
3-4	H	4-34.1	K	6-23	D	7-48	D	8-26	H
3-5	H	4-34.2	K	6-24	D	7-49	D	8-27	D
3-6	G	4-35	E	6-25	J	7-50	L	8-28	D
3-6.1	G	4-36	E	6-26	D	7-50.1/7-50.2	K	8-29	D
3-6.2	G	4-37	D	6-27	D	7-51	L	8-30	D
		4-38	K	6-28	D	7-52	D	8-31	H
		4-39	K	6-29	H	7-53	D	8-32	D
		4-40	K	6-30	D	7-54	J	8-33	F

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
8-34	K	9-52	F	10-29	D	12-26	C	15-44	G
8-35	E	9-53	F	10-30	L	13-1	C	15-45	G
8-36	G	9-54	D	10-31	L	13-2	C	16-1	H
8-37	D	9-55	D	10-32	L	13-3	C	16-2	F
8-38	G	9-56	D	10-32.1/		13-4	C	16-3	F
8-39	D	9-57	D	10-32.2	L	14-1	K	16-4	C
8-40	D	9-58	D	10-33	D	14-2	D	16-5	C
8-41	D	9-59	D	10-34	H	14-3	D	16-6	C
8-42	D	9-60	H	10-35	L	14-4	E	16-7	C
8-43	D	9-61	D	10-36	L	14-5	F	16-8	C
8-44	D	9-62	L	10-36.1/		14-6	K	16-9	J
8-45	D	9-63	L	10-36.2	L	14-7	D	A-1	C
8-46	D	9-64	L	10-37	L	14-8	D	A-2	C
8-47	D	9-64.1/9-64.2	L	10-38	L	14-9	D	A-3	H
8-48	D	9-65	D	10-38.1	H	14-10	L	A-4	E
8-49	D	9-66	D	10-38.2	H	14-11	L	A-5	C
8-50	D	9-67	D	10-39	K	14-12	J	A-6	C
8-51	D	9-68	D	10-40	L	14-13	C	A-7	C
8-52	D	10-1	L	10-40.1/		14-14	C	A-8	C
9-1	E	10-2	J	10-40.2	L	14-15	C	A-9	C
9-2	L	10-3	L	10-41	H	14-16	C	A-10	C
9-3	K	10-4	L	10-42	L	14-17	C	A-11	C
9-4	C	10-4.1	L	10-43	L	14-18	D	A-12	D
9-5	C	10-4.2	L	10-44	H	14-19	L	A-13	E
9-6	C	10-4.3	L	10-44.1/		15-1	K	A-14	C
9-7	D	10-5	J	10-44.2	H	15-2	K	A-15	C
9-8	C	10-6	C	10-45	E	15-3	K	B-1	L
9-9	D	10-7	J	10-46	J	15-4	K	B-2	L
9-10	D	10-8	J	10-47	L	15-5	C	B-3	L
9-11	C	10-8.1/10-8.2	J	10-48	L	15-6	E	B-4	L
9-12	C	10-9	J	10-49	D	15-7	K	B-5	L
9-13	F	10-10	L	11-1	G	15-8	C	B-6	L
9-14	F	10-11	L	11-2	C	15-9	C	B-7	L
9-15	F	10-12	L	11-3	C	15-10	C	B-8	L
9-16	F	10-13	L	11-4	C	15-11	L	B-9	L
9-17	C	10-14	L	11-5	H	15-12	L	B-10	L
9-18	F	10-14.1	L	11-6	H	15-12.1/		B-11	L
9-19	H	10-14.2	L	11-6.1/11-6.2	H	15-12.2	L	B-12	L
9-20	K	10-14.3	L	11-7	H	15-13	C	B-13	L
9-21	K	10-14.4	L	11-8	C	15-14	C	B-14	L
9-22	K	10-15	J	11-9	C	15-15	E	B-15	L
9-23	L	10-16	G	11-10	C	15-16	G	B-16	L
9-24	K	10-17	F	12-1	C	15-17	K	B-17	L
9-25	L	10-18	J	12-2	C	15-18	K	B-18	L
9-26	D	10-19	D	12-3	E	15-19	K	B-19	L
9-27	E	10-20	J	12-4	L	15-20	K	B-20	L
9-28	D	10-20.1	L	12-5	H	15-20.1/		B-21	L
9-29	D	10-20.2	L	12-6	H	15-20.2	K	B-22	L
9-30	D	10-20.3	L	12-7	C	15-21	G	B-23	L
9-31	F	10-20.4	L	12-8	D	15-22	G	B-24	L
9-32	D	10-20.5	L	12-9	H	15-23	C	B-25	L
9-33	H	10-20.6	L	12-10	H	15-24	C	B-26	L
9-34	G	10-20.7	L	12-10.1/		15-25	C	B-27	L
9-35	D	10-20.8	L	12-10.2	H	15-26	C	B-28	L
9-36	D	10-20.9	L	12-11	H	15-27	K	B-29	L
9-37	D	10-20.10	L	12-12	H	15-28	K	B-30	L
9-38	D	10-20.11	L	12-13	C	15-29	K	B-31	L
9-39	D	10-20.12	L	12-14	D	15-30	K	B-32	L
9-40	D	10-20.13	L	12-15	H	15-31	K	B-33	L
9-41	D	10-20.14	L	12-16	H	15-32	J	B-34	L
9-42	D	10-20.15	L	12-16.1/		15-33	G	B-35	L
9-43	D	10-21	J	12-16.2	H	15-34	K	B-36	L
9-44	H	10-22	J	12-17	E	15-35	D	B-37	L
9-45	H	10-22.1/		12-18	C	15-36	J	B-38	L
9-46	J	10-22.2	J	12-19	H	15-37	J	B-39	L
9-47	F	10-23	F	12-20	C	15-38	J	B-40	L
9-48	J	10-24	E	12-21	K	15-39	J	B-41	L
9-48.1/9-48.2	K	10-25	J	12-22	C	15-40	J	B-42	L
9-49	H	10-26	J	12-23	H	15-41	J	B-43	L
9-50	K	10-27	J	12-24	D	15-42	L	B-44	L
9-51	F	10-28	J	12-25	C	15-43	G	B-45	L

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
B-46	L	H-7	H	Index-2	L				
B-47	L	H-8	J	Index-3	K				
B-48	L	H-9	J	Index-4	K				
B-49	L	I-1	C	Index-5	K				
B-50	L	I-2	C	Index-6	L				
B-51	L	I-3	C	Index-7	L				
B-52	L	I-4	C	Index-8	L				
B-53	L	J-1	G	Index-9	L				
B-54	L	J-2	D	Index-10	L				
B-55	L	J-3	C	Index-11	L				
B-56	L	J-4	D	Index-12	L				
B-57	L	J-5	D	Index-13	L				
B-58	L	J-6	H	Index-14	L				
B-59	L	J-7	H	Index-15	L				
B-60	L	J-8	F	Index-16	L				
B-61	L	J-9	H	Comment Sheet	L				
B-62	L	J-10	C	Mailer					
B-63	L	J-11	G	Back Cover	-				
B-64	L	J-12	H						
B-65	L	J-13	G						
B-66	L	J-14	H						
B-67	L	J-15	G						
B-68	L	J-16	H						
C-1	C	J-17	D						
C-2	G	J-18	H						
C-3	G	J-19	H						
C-4	G	J-20	H						
C-5	G	J-21	H						
C-6	C	J-22	H						
C-7	C	J-23	H						
C-8	C	J-24	L						
C-9	D	J-25	H						
C-10	C	J-26	H						
C-11	C	J-27	H						
C-12	F	J-28	H						
C-13	C	J-29	G						
C-14	C	J-30	G						
C-15	H	K-1	C						
D-1	C	K-2	D						
D-2	C	L-1	K						
D-3	C	L-2	K						
D-4	C	L-3	C						
E-1	H	L-4	C						
E-2	C	L-5	C						
F-1	C	L-6	C						
F-2	C	L-7	C						
F-3	C	L-8	J						
F-4	C	L-9	C						
F-5	C	M-1	C						
F-6	C	M-2	C						
F-7	C	M-3	C						
G-1	C	M-4	C						
G-2	C	M-5	C						
G-3	C	M-6	C						
G-4	C	M-7	C						
G-5	C	M-8	C						
G-6	C	M-9	C						
G-7	C	N-1	L						
G-8	L	N-2	J						
G-9	L	O-1	C						
G-10	C	O-2	H						
G-11	C	O-3	G						
H-1	H	P-1	G						
H-2	J	P-2	K						
H-2.1/H-2.2	J	P-3	G						
H-3	K	P-4	G						
H-4	J	P-5	G						
H-4.1	J	P-6	G						
H-4.2	J	P-7	G						
H-5	D	P-8	G						
H-6	D	Index-1	K						

PREFACE

This manual describes the CONTROL DATA® Network Operating System (NOS) Version 2. NOS can operate on the following computer systems:

CDC® CYBER 180 Computer Systems

Models 810, 830, 835, 840, 845, 850, 855, 860, 870, 960, 990, 994, and 995

CDC CYBER 170 Computer Systems

Models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 815, 825, 835, 845, 855, 865, and 875

CDC CYBER 70 Computer Systems

Models 71, 72, 73 and 74

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 (60459690), 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. 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 file loading. 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.

SUBMITTING COMMENTS

The last page of this manual is a comment sheet. Please use it to give your opinion on the manual's usability, to suggest specific improvements, and to report any errors. If the comment sheet has already been used, you can mail your comments to:

Control Data
Technical Publications ARH219
4201 Lexington Avenue North
St. Paul, MN 55126-6198

Additionally, if you have access to SOLVER, an online facility for reporting problems, you can use it to submit comments about the manual. Declare your problem type as DOC and use NS2 as the product identifier.

Address questions about physical packaging and/or distribution of printed materials to Literature and Distribution Services at the following address:

Control Data
Literature and Distribution Services
308 North Dale Street
St. Paul, MN 55103-2495

or you can call (612) 292-2101. If you are a Control Data employee, call (612) 292-2100.

CYBER SOFTWARE SUPPORT HOTLINE

Control Data's CYBER Software Support maintains a hotline to assist you if you have trouble using our products. If you need help beyond that provided in the documentation or find that the product does not perform as described, call us at one of the following numbers and a support analyst will work with you.

From USA and Canada: (800) 345-9903

From other countries: (612) 851-4131

CONVENTIONS

REFERENCES TO OTHER MANUALS IN THE NOS REFERENCE SET

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.

180-CLASS MAINFRAMES

Some of the CYBER 170 Computer Systems share many of the functional and architectural attributes of the CYBER 180 Computer Systems. Specifically, CYBER 170 Models 815, 825, 835, 845, and 855 fall into this category. It is sometimes convenient to refer to the CYBER 180 models and these CYBER 170 models collectively. This manual uses the term 180-class mainframes to refer to this collection.

EXTENDED MEMORY

Extended memory for the CYBER 180-class machines and models 865 and 875 is unified extended memory (UEM) and may also include extended core storage (ECS), extended semiconductor memory (ESM), or STORNET. Extended memory for model 176 is large central memory extended (LCME) and may include ECS, ESM, or STORNET. Extended memory for all other NOS computer systems is either ECS, ESM, or STORNET.

In this manual, ECS refers to ECS, ESM, and STORNET. Extended memory refers to all forms of extended memory unless otherwise noted. However, when referencing extended memory in the context of an ECS multiframe complex, distributive data path (DDP), or low-speed port (LSP) access, UEM and LCME are excluded. ECS, ESM, and STORNET are the only forms of extended memory that can be shared in ECS multiframe complex and can be accessed by a DDP or LSP. For CYBER 180-class machines, only UEM may be accessed directly from CPU programs as extended field length; similarly, on the model 176, only LCME may be accessed directly from CPU programs. The other forms of extended memory are supported only as mass storage devices on these machines.

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 display 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 an opening parenthesis for their initial delimiter and a closing parenthesis for their terminator.

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 K 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.
- \leftarrow 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

You may want to consult the following Control Data manuals for information related to topics discussed in this manual. For your convenience, these manuals are listed by category: CDCNET manuals, hardware manuals, NOS 2 manuals, and optional product manuals.

If you need a brief description of the NOS product commands and their parameters, refer to the NOS Version 2 Applications Programmer's Instant (60459370). In addition, brief descriptions of all NOS operating system and NOS product set manuals are available in the NOS System Information manual. You access this online manual by logging into NOS and entering the EXPLAIN command.

Control Data manuals are available through Control Data sales offices or through Literature Distribution Services:

Control Data
Literature and Distribution Services
308 North Dale Street
St. Paul, MN 55103-2495

CDCNET MANUALS

The following manuals describe the Control Data Distributed Communications Network (CDCNET).

<u>Control Data Publication</u>	<u>Publication Number</u>
CDCNET Access Guide†	CDCNETA
CDCNET Batch Device User Guide	60463863
CDCNET Conceptual Overview	60461540
CDCNET Configuration and Site Administration Guide	60461550
CDCNET Terminal Interface Usage	60463850
CDCNET TCP/IP Usage	60000214

† Online manual only. Enter EXPLAIN,M>manual.

HARDWARE MANUALS

The following manuals describe Control Data computer systems and related equipment.

<u>Control Data Publication</u>	<u>Publication Number</u>
CYBER 70 Model 71 Computer System Hardware Reference Manual	60453300
CYBER 70 Model 72 Computer System Hardware Reference Manual	60347000
CYBER 170 Computer Systems Models 171 through 175 (Levels A, B, C) Model 176 (Level A, B, C) Hardware Reference Manual	60420000
CYBER 170 Computer Systems Models 720, 730, 740, 750, and 760 Model 176 (Level B/C) Hardware Reference Manual	60456100
CYBER 170 Computer Systems Models 815 and 825 Hardware Reference Manual	60469350
CYBER 170 Computer Systems Models 835, 845, and 855 CYBER 180 Computer Systems Models 835, 840, 845, 850, 855, 860, and 990 CYBER 990E and 995E Computer Systems CYBER 170 State Hardware Reference Manual	60469290
CYBER 170 Computer Systems Models 835, 845, and 855 CYBER 180 Computer Systems Models 835, 845, and 855 Hardware Operator's Guide	60458390
CYBER 170 Computer Systems Models 865 and 875 Hardware Reference Manual	60458920
CYBER 180 Models 810 and 830 Computer Systems Hardware Reference Manual	60469420
CYBER 840A, 850A, 860A, and 870A Computer Systems Hardware Reference Manual	60463560
5870 Printer User's Reference Manual	60462720

NOS 2 MANUALS

The following manuals describe NOS 2.

<u>Control Data Publication</u>	<u>Publication Number</u>
Common Memory Manager Version 1 Reference Manual	60499200
COMPASS Version 3 Reference Manual	60492600
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
FORM Version 1 Reference Manual	60496200
Modify Version 1 Reference Manual	60450100
NOS Online Maintenance Software Reference Manual	60454200
NOS Version 2 Administration Handbook	60459840
NOS Version 2 Analysis Handbook	60459300
NOS Version 2 Applications Programmer's Instant	60459360
NOS Version 2 Diagnostic Index	60459390
NOS Version 2 Full Screen Editor User's Guide	60460420
NOS Version 2 Installation Handbook	60459320
NOS Version 2 Operations 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 Screen Formatting Reference Manual	60460430
NOS Version 2 Security Administrator's Handbook	60460410
NOS Version 2 System Overview	60459270
NOS Version 2 Systems Programmer's Instant	60459370
SYMPL Version 1 Reference Manual	60496400

OPTIONAL PRODUCT MANUALS

The following manuals describe optional products.

<u>Control Data Publication</u>	<u>Publication Number</u>
APEX IV Reference Manual	84002550
APL Version 2 Reference Manual	60454000
Automatically Programmed Tooling System (APT IV) Version 2 Reference Manual	17326900
BASIC Version 3 Reference Manual	19983900
COBOL Version 5 Reference Manual	60497100
CYBER Database Control System 2 Database Administrator Reference Manual	60485200
CYBER Database Control System 2 Database Application Programming Reference Manual	60485300
CYBER Interactive Debug Version 1 Reference Manual	60481400
Data Catalogue 2 Version 1 Reference Manual	60483200
Data Catalogue 2 Version 2 Reference Manual	60483350
DDL Version 3 Reference Manual, Volume 1	60481900
DDL Version 3 Reference Manual, Volume 2	60482000
DDL Version 3 Reference Manual, Volume 3	60482100
FORTRAN DataBase Facility Version 1 Reference Manual	60482200
FORTRAN Extended Version 4 Common Library Mathematical Routines Reference Manual	60498200
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

Optional Product Manuals (Continued)

<u>Control Data Publication</u>	<u>Publication Number</u>
Information Management Facility Version 1 Application Programming Reference Manual	60484500
Message Control System Version 1 Reference Manual	60480300
Network Access Method Version 1 Host Application Programming Reference Manual	60499500
Network Access Method Version 1 Network Definition Language Reference Manual	60480000
Network Access Method Version 1/Communications Control Program Version 3 Terminal Interfaces Reference Manual	60480600
NOS Version 2 Tape Management System (TMS) Site Operations Manual	60463350
NOS Version 2 Tape Management System (TMS) User Reference Manual	60463110
Pascal Version 1.1 Reference Manual	60497700
PERT/Time Version 2 Reference Manual	60456030
Query Update Version 3 Reference Manual	60498300
Remote Batch Facility Version 1 Reference Manual	60499600
Remote Host Facility Access Method Reference Manual	60459990
Remote Host Facility Usage	60460620
Sort/Merge Version 5 Reference Manual	60497500
TAF Version 1 Reference Manual	60459500
TAF Version 1 User's Guide	60459520
TAF/CRM Data Manager Version 1 Reference Manual	60459510
TOTAL-CDC Version 2 Reference Manual	76070300
Update Version 1 Reference Manual	60449900
XEDIT Version 3 Reference Manual	60455730
8-Bit Subroutines Version 1 Reference Manual	60495500

CONTENTS

<p>1. SYSTEM DESCRIPTION 1-1</p> <p>System Hardware 1-1</p> <p style="padding-left: 20px;">Central Processor Unit 1-2</p> <p style="padding-left: 20px;">Central Memory 1-2</p> <p style="padding-left: 40px;">Job Field Length 1-2</p> <p style="padding-left: 40px;">Central Memory Resident 1-3</p> <p style="padding-left: 20px;">Extended Memory 1-3</p> <p style="padding-left: 20px;">Peripheral Processors 1-4</p> <p style="padding-left: 20px;">Peripheral Equipment 1-4</p> <p>System Software 1-5</p> <p style="padding-left: 20px;">User Programs 1-5</p> <p style="padding-left: 20px;">Operating System 1-5</p> <p style="padding-left: 40px;">CYBER Loader 1-6</p> <p style="padding-left: 40px;">CYBER Record Manager 1-6</p> <p>2. FILES 2-1</p> <p>File Names 2-1</p> <p>File Structure 2-1</p> <p style="padding-left: 20px;">CYBER Record Manager File Structure 2-3</p> <p style="padding-left: 20px;">NOS File Structure 2-3</p> <p style="padding-left: 20px;">Physical File Structure 2-4</p> <p style="padding-left: 40px;">Card Files 2-6</p> <p style="padding-left: 40px;">Mass Storage Files 2-6</p> <p style="padding-left: 40px;">Magnetic Tape Files 2-7</p> <p>File Types 2-11</p> <p style="padding-left: 20px;">Input Files 2-11</p> <p style="padding-left: 20px;">Queued Files 2-11</p> <p style="padding-left: 20px;">Local Files 2-12</p> <p style="padding-left: 20px;">Primary Files 2-13</p> <p style="padding-left: 20px;">Direct Access Files 2-13</p> <p style="padding-left: 20px;">Library Files 2-13</p> <p>Permanent Files 2-13</p> <p style="padding-left: 20px;">Indirect Access Permanent Files 2-14</p> <p style="padding-left: 20px;">Direct Access Permanent Files 2-14</p> <p>Mass Storage File Residence 2-15</p> <p>Family Devices 2-15</p> <p>Auxiliary Devices 2-16</p> <p>Alternate Storage 2-16</p> <p>Libraries 2-17</p> <p style="padding-left: 20px;">User Name Library 2-17</p> <p style="padding-left: 20px;">Program Libraries 2-17</p> <p style="padding-left: 20px;">User Libraries 2-17</p> <p>3. JOB FLOW AND EXECUTION 3-1</p> <p>Job Initiation and Termination 3-1</p>	<p>1-1</p> <p>1-1</p> <p>1-2</p> <p>1-2</p> <p>1-2</p> <p>1-3</p> <p>1-3</p> <p>1-4</p> <p>1-4</p> <p>1-4</p> <p>1-5</p> <p>1-5</p> <p>1-5</p> <p>1-6</p> <p>1-6</p> <p>2-1</p> <p>2-1</p> <p>2-1</p> <p>2-3</p> <p>2-3</p> <p>2-4</p> <p>2-6</p> <p>2-6</p> <p>2-7</p> <p>2-11</p> <p>2-11</p> <p>2-11</p> <p>2-12</p> <p>2-13</p> <p>2-13</p> <p>2-13</p> <p>2-14</p> <p>2-14</p> <p>2-15</p> <p>2-15</p> <p>2-16</p> <p>2-16</p> <p>2-17</p> <p>2-17</p> <p>2-17</p> <p>2-17</p> <p>3-1</p> <p>3-1</p>	<p>Batch Jobs 3-1</p> <p>Interactive Jobs 3-2</p> <p style="padding-left: 20px;">Login 3-2</p> <p style="padding-left: 20px;">Logout and Application Switching Procedure 3-4</p> <p>Submitting Jobs to Remote Hosts 3-6</p> <p>Job Origin Types 3-6.2</p> <p>Job Service Classes 3-6.2</p> <p>Job Names 3-7</p> <p style="padding-left: 20px;">Job Sequence Name (JSN) 3-7</p> <p style="padding-left: 20px;">User Job Name (UJN) 3-8</p> <p>Validation 3-8</p> <p>Accounting 3-8</p> <p>Job Scheduling 3-9</p> <p>Job Control 3-9</p> <p style="padding-left: 20px;">Field Length Control 3-9</p> <p style="padding-left: 20px;">Input File Control 3-11</p> <p style="padding-left: 20px;">Time Limit Control 3-11</p> <p style="padding-left: 20px;">SRU Limit Control 3-11</p> <p style="padding-left: 20px;">Command Limit Control 3-11</p> <p style="padding-left: 20px;">Rollout Control 3-12</p> <p style="padding-left: 20px;">Error Control 3-12</p> <p>Security Features 3-13</p> <p style="padding-left: 20px;">Memory Protection 3-13</p> <p style="padding-left: 20px;">System and File Access Controls 3-13</p> <p style="padding-left: 20px;">Security Access Levels 3-14</p> <p style="padding-left: 20px;">Security Access Categories 3-16</p> <p style="padding-left: 20px;">Responsibilities for Data Security 3-16</p> <p style="padding-left: 20px;">Security Conflict Processing 3-17</p> <p>Job Completion 3-17</p> <p>4. PROCEDURES 4-1</p> <p>Procedure File Residence and Search Order 4-2</p> <p>Procedure Structure 4-2</p> <p>Procedure Processing 4-3</p> <p style="padding-left: 20px;">Interactive Parameter Entry 4-4</p> <p style="padding-left: 20px;">Procedure Expansion 4-4</p> <p style="padding-left: 40px;">.EXPAND Directive 4-6</p> <p style="padding-left: 40px;">Inhibit Character 4-6</p> <p style="padding-left: 40px;">Concatenation Character 4-8</p> <p style="padding-left: 40px;">.EX Directive 4-9</p> <p style="padding-left: 20px;">Execution of the Command Record 4-9</p> <p>Procedure Directives 4-9</p> <p style="padding-left: 20px;">Procedure Header (.PROC) Directive 4-11</p> <p style="padding-left: 40px;">Interactive Format 4-11</p> <p>Checklist Entries 4-13</p> <p style="padding-left: 20px;">Special Values 4-14</p>
--	---	---

Checklist Patterns	4-16	FILE Function	6-12
Menu Format	4-22	DT Function	6-14
Formatting and Help Directives	4-23	LEN Function	6-16
.CORRECT Directive	4-23	LVL Function	6-16
.ENDHELP Directive	4-24	NUM Function	6-16.1
.ENTER Directive	4-24	STR Function	6-16.1
.Fn Directive	4-24.1	STRB Function	6-16.2
.F7 Directive	4-24.1	STRD Function	6-16.3
.HELP Directive	4-25	Command Descriptions	6-16.3
.NOCLR Directive	4-31	BEGIN Command	6-16.3
.NOTE Directive	4-31	DISPLAY Command	6-16.4
.PAGE Directive	4-32	ELSE Command	6-17
.PROMPT Directive	4-32	ENDIF Command	6-18
Expansion Control Directives	4-33	ENDW Command	6-19
.CC Directive	4-33	EXIT Command	6-19
.EXPAND Directive	4-33	IF (or IFE) Command	6-20
.IC Directive	4-34	MODE Command	6-22
.SET Directive	4-34	NOEXIT Command	6-24
File Directives	4-34.1	ONEXIT Command	6-24
.DATA Directive	4-34.2	REVERT Command	6-24
.EOF (or .EOP) Directive	4-38	SET Command	6-25
.EOR (or .EOS) Directive	4-38	SKIP Command	6-30
Branching Directives	4-38	WHILE Command	6-31
.ELSE Directive	4-38		
.ENDIF Directive	4-39		
.EX Directive	4-39	7. JOB CONTROL COMMANDS	7-1
.IF (or .IFE) Directive	4-40		
.COMMENT (.*) Directive	4-40.1	BLOCK Command	7-2
Calling a Procedure (BEGIN Command)	4-41	CFO Command	7-3
Requesting Help on Procedure Calls	4-46	CHARGE Command	7-4
Ending a Procedure (REVERT Command)	4-51	CHVAL Command	7-5
Parameter Matching Modes	4-53	CLASS Command	7-7
Order-Dependent Mode	4-53	COMMENT Command	7-9
Order-Independent Mode	4-53	CTIME Command	7-10
		DAYFILE Command	7-10
		DROP Command	7-12.1
5. COMMAND PROCESSING	5-1	ENQUIRE Command	7-13
Command Format	5-2	ENTER Command	7-21
Command Processing Flow	5-5	ERRMSG Command	7-22
Exit Processing	5-7	GO Command	7-22
		HTIME Command	7-22.1
		Job Command	7-23
		LDI Command	7-26
6. FLOW CONTROL COMMANDS	6-1	LENGTH Command	7-27
Command Syntax	6-2	LIMITS Command	7-28
Operators	6-2	Resource Limits	7-29
Arithmetic Operators	6-2	Other Characteristics	7-30
Relational Operators	6-3	User Permissions	7-31
Logical Operators	6-4	Network Applications	7-32
String Operators	6-4	Local Applications	7-32
Order of Evaluation	6-5	Privileged Network Applications	7-32
Operands	6-5	Shell Permissions	7-32
Constants	6-5	Security Permissions	7-33
Numeric Strings	6-5	Security Access Categories	7-33
Literals	6-5	Security Access Levels	7-33
Symbolic Names	6-7	Service Classes	7-33
Functions	6-12	Default Service Classes	7-33
		LISTLID Command	7-36

MACHINE Command	7-39	BASIC Command	8-12
MFL Command	7-39	BATCH Command	8-13
NORERUN Command	7-40	EXECUTE Command	8-13
NOTE Command	7-40	FORTRAN Command	8-14
OFFSW Command	7-41	FTNTS Command	8-15
ONSW Command	7-42	NULL Command	8-15
PASSWOR Command	7-42	Interactive Status Commands	8-16
PAUSE Command	7-44	Detach Command (ctD)	8-16
PROTECT Command	7-44	Immediate Job Status	
QGET Command	7-45	Command (ctE)	8-17
QUEUE7 Command	7-46	Abbreviated Job Status	
REDO Command	7-50	Command (cts)	8-17
RERUN Command	7-50.1	Job Processing Commands	8-18
RESOURC Command	7-50.1	APPSW Command	8-18
Deadlock Prevention	7-52	BYE Command	8-19
Single Resource Use	7-54	DIAL Command	8-20.1
Tape Units	7-54	EXPLAIN Command	8-20.1
Resource Overcommitment	7-55	GOODBYE Command	8-21
Altering Resource Requirements	7-56	HELLO Command	8-21
Unit Assignment	7-57	HELLO7 Command	8-22
RFL Command	7-57	HELLO7 Directives	8-22
ROLLOUT Command	7-58	HELP Command	8-26
RTIME Command	7-58	HELPME Command	8-26
SETASL Command	7-59	LIB Command	8-26
SETCORE Command	7-60	LIST Command	8-28
SETJAL Command	7-61	LOGIN Command	8-28
SETJOB Command	7-62	LOGOUT Command	8-29
SETJSL Command	7-63	NOSORT Command	8-29
SETPR Command	7-64	RECOVER Command	8-30
SETTL Command	7-64	RUN Command	8-35
SHELL Command	7-66	Secure Login Command	8-36
STIME Command	7-67	SHOW Command	8-36
SUBMIT Command	7-68	TEXT Command	8-37
SWITCH Command	7-73	WHATJSN Command	8-37
UPROC Command	7-73	X Command	8-38
USECPU Command	7-74	XMODEM Command	8-38
USER Command	7-74	Primary File Editing Commands	8-39
		Parameter Format	8-40
		Suppressing Editing Responses	8-41
		Line Number Overlap	8-41
		Inserting Lines	8-42
		ALTER Command	8-42
		DELETE Command	8-44
		DUP Command	8-45
		LIST Command	8-46
		MOVE Command	8-48
		READ Command	8-49
		RESEQ Command	8-51
		WRITE Command	8-51
		WRITEN Command	8-52
8. COMMANDS FOR INTERACTIVE JOBS	8-1		
Terminal Control Commands	8-2		
ASCII Command	8-2		
AUTO Command	8-3		
BRIEF Command	8-4		
CSET Command	8-4		
EFFECT Command	8-4		
KEY Command	8-5		
LINE Command	8-6.1		
NORMAL Command	8-7		
SCREEN Command	8-7		
TDU Command	8-8.1		
TIMEOUT Command	8-9		
TRMDEF Command	8-9		
Using TRMDEF with NAM/CDCNET	8-10		
Examples	8-11		
Subsystem Selection Commands	8-11		
ACCESS Command	8-12		
		9. FILE MANAGEMENT COMMANDS	9-1
		ASSIGN Command	9-2
		BKSP Command	9-4

CLEAR Command	9-5	CATLIST Command	10-10
COPY Command	9-5	CHANGE Command	10-15
Copy Termination	9-10	COMMON Command	10-16
Block Sizes	9-10	DEFINE Command	10-17
Processing Options	9-10	DROPDS Command	10-20
COPYBF Command	9-11	FTP Command	10-20.1
COPYBR Command	9-12	FTP Directives	10-20.1
COPYCF Command	9-13	APPEND_FILE Directive	10-20.2
COPYCR Command	9-15	CHANGE_AUTO_LOGIN_MODE	
COPYEI Command	9-16	Directive	10-20.3
COPYSBF Command	9-17	CHANGE_BELL_MODE Directive	10-20.3
COPYX Command	9-18	CHANGE_DEBUG_MODE Directive	10-20.4
DOCUMENT Command	9-20	CHANGE_FILE_NAME Directive	10-20.4
EVICT Command	9-22	CHANGE_INPUT_FILE Directive	10-20.4
FCOPY Command	9-23	CHANGE_OUTPUT_FILE Directive	10-20.5
LIST80 Command	9-26	CHANGE_PORT_COMMAND_MODE	
LOCK Command	9-26	Directive	10-20.5
LO72 Command	9-27	CHANGE_TRANSFER_TYPE	
MFQUEUE Command	9-31	Directive	10-20.5
Parameter Descriptions	9-31	CHANGE_VERBOSE_MODE	
The Routing Directive	9-33	Directive	10-20.6
NEW Command	9-34	CREATE_HOST_CONNECTION	
OUT Command	9-35	Directive	10-20.6
OVWRITE Command	9-35	DEFINE_AUTO_LOGIN Directive	10-20.7
PACK Command	9-36	DELETE_FILE Directive	10-20.7
PRIMARY Command	9-37	DELETE_HOST_CONNECTION	
RENAME Command	9-37	Directive	10-20.8
REQUEST Command	9-38	DISPLAY_COMMAND_INFORMATION	
RESEQ Command	9-40	Directive	10-20.8
RETURN Command	9-42	DISPLAY_DIRECTORY Directive	10-20.8
REWIND Command	9-43	DISPLAY_FILE_NAMES Directive	10-20.9
ROUTE Command	9-44	DISPLAY_FTP_OPTIONS	
Parameter Descriptions	9-44	Directive	10-20.9
Implicit Routing	9-50	DISPLAY_LOCAL_HELP Directive	10-20.9
SCOPY Command	9-51	DISPLAY_REMOTE_HELP	
SECHDR Command	9-54	Directive	10-20.10
SETFAL Command	9-55	GET_FILE Directive	10-20.10
SETFS Command	9-56	LOGIN_USER Directive	10-20.12
SKIPEI Command	9-56	QUIT Directive	10-20.13
SKIPF Command	9-57	REPLACE_FILE Directive	10-20.14
SKIPFB Command	9-57	SEND_FTP_COMMAND Directive	10-20.15
SKIPR Command	9-58	GET Command	10-21
SORT Command	9-58	MFLINK Command	10-22
TCOPY Command	9-59	MFLINK Directives	10-23
TDUMP Command	9-62	APPEND Directive	10-24
UNLOAD Command	9-64	ATTACH Directive	10-24
UNLOCK Command	9-65	CHANGE Directive	10-24
VERIFY Command	9-65	CHARGE Directive	10-25
WRITEF Command	9-68	DEFINE Directive	10-25
WRITER Command	9-68	DROPDS Directive	10-25
		GET Directive	10-25
		PACKNAM Directive	10-25
		PERMIT Directive	10-26
10. PERMANENT FILE COMMANDS	10-1	PURGE Directive	10-26
		REPLACE Directive	10-26
Common Parameters	10-2	SAVE Directive	10-26
APPEND Command	10-7	USER Directive	10-26
ATTACH Command	10-8	Interactive Use of MFLINK	10-27

OLD Command	10-28	PROFILE Command	14-11
PACKNAM Command	10-29	UPDATE Command	14-14
PERMIT Command	10-30	XEDIT Command	14-18
PURGALL Command	10-31		
PURGE Command	10-32.1		
RECLAIM Command	10-33		
Magnetic Tapes	10-34	15. LIBRARY MAINTENANCE	15-1
Command Format	10-35		
RECLAIM Output	10-44	Library File Access Methods	15-1
REPLACE Command	10-45	Library Record Types	15-2
SAVE Command	10-46	CATALOG Command	15-5
SETPFAC Command	10-48	COPYL and COPYLM Commands	15-7
SETPFAL Command	10-49	GTR Command	15-11
		ITEMIZE Command	15-14
		LIBEDIT Command	15-17
		Command Format	15-18
11. LOAD/DUMP MEMORY COMMANDS	11-1	LIBEDIT Directives	15-20
		ADD Directive	15-23
DMB Command	11-1	BEFORE Directive	15-24
DMD Command	11-5	BUILD Directive	15-24
DMDECS Command	11-6	COMMENT Directive	15-25
DMP Command	11-6.1	COPY Directive	15-25
DMPECS Command	11-7	DATE Directive	15-25
LBC Command	11-8	DELETE Directive	15-26
LOC Command	11-8	FILE Directive	15-26
PBC Command	11-9	IGNORE Directive	15-27
RBR Command	11-9	INSERT or AFTER Directive	15-27
WBR Command	11-10	LIBGEN Directive	15-28
		LIST Directive	15-28
		NEW Directive	15-28
		NOINS Directive	15-28
		NOREP Directive	15-29
		NOREW Directive	15-29
		OLD Directive	15-29
		RENAME Directive	15-29
		REPLACE Directive	15-30
		REWIND Directive	15-30
		TYPE or NAME Directive	15-30
		VFYLIB Directive	15-31
		*/ Directive	
		(Comment Directive)	15-31
		LIBEDIT Output	15-31
		LIBGEN Command	15-32
		ULIB Command	15-34
		VFYLIB Command	15-35
		Library Processing Examples	15-37
		LIBRARY Command	15-43
		Maximum Size of Global Library	
		Set	15-43
		Library Search Order	15-44
		Examples	15-44
		16. TERMINAL INPUT/OUTPUT	16-1
		Terminals	16-1
		Input/Output Conventions	16-2
		Input From Terminal	16-2
		Length of Output/Input Lines	16-3
11. LOAD/DUMP MEMORY COMMANDS	11-1		
DMB Command	11-1		
DMD Command	11-5		
DMDECS Command	11-6		
DMP Command	11-6.1		
DMPECS Command	11-7		
LBC Command	11-8		
LOC Command	11-8		
PBC Command	11-9		
RBR Command	11-9		
WBR Command	11-10		
12. TAPE MANAGEMENT	12-1		
Tape Assignment	12-1		
Command Rules	12-2		
Processing Options	12-3		
ASSIGN Command	12-5		
BLANK Command	12-9		
LABEL Command	12-11		
LISTLB Command	12-19		
REQUEST Command	12-21		
VSN Command	12-25		
13. CHECKPOINT/RESTART	13-1		
CKP Command	13-1		
RESTART Command	13-2		
14. SYSTEM UTILITY COMMANDS	14-1		
EDIT Command	14-1		
FSE Command	14-2		
KRONREF Command	14-4		
MODIFY Command	14-5		
OPLEDIT Command	14-8		
PDU Command	14-10		

Terminating Input Line	16-3	Aborting Output Block	16-5
Correcting Input Line	16-3	Interrupting Job Step	16-6
Deleting Input Line	16-4	Terminating Job Step	16-6
Entering Input Before Prompts	16-4	Continuing a Suspended Job	16-7
Page Wait	16-5	Using ATTN on IBM 2741 Terminals	16-8
Suspending Output	16-5	Control Bytes	16-9

APPENDIXES

A. CHARACTER SETS	A-1	Optional Labels	G-10
Character Set Anomalies	A-2	HDR2 through HDR9 - Additional File Header Labels	G-10
Character Set Tables	A-2	EOF2 through EOF9 - Additional End-Of-File Labels	G-10
Interactive Jobs	A-3	EOV2 through EOV9 - Additional End-Of-Volume Labels	G-10
Batch Jobs	A-3	User Labels	G-11
Jobs Using Line Printers	A-3		
Jobs Using Magnetic Tapes	A-12		
B. DIAGNOSTIC MESSAGES	B-1	H. LINE PRINTER CARRIAGE CONTROL	H-1
C. GLOSSARY	C-1	Printed Data	H-1
D. SAMPLE JOB OUTPUT	D-1	Paper Length	H-1
E. TERMINAL CHARACTER CONVERSION	E-1	580 Line Printers	H-1
Data Input	E-1	533/536, 537, and 585 Printers	H-2
Data Output	E-2	Carriage Control	H-2
F. CARD FILE DATA CONVERSION	F-1	Format Channel Selection on Non-PFC Printers	H-4.2
Input Card File Formats	F-2	Format Channel Selection on 533/536, 537, and 585 Printers	H-4.2
Coded Cards	F-3	Carriage Control Arrays	H-7
Conversion Modes	F-3	Array Syntax	H-7
Literal Input	F-3	Reserved Format Channels for VFU Processing	H-9
Binary Cards	F-5	Example	
Summary	F-5	H-9	
Punch File Formats	F-5	I. OBSOLETE TAPE FORMATS	I-1
Coded Cards (Punch)	F-6	B (Blocked) Format	I-1
Binary Cards (Punch B)	F-6	E (Line Image) Format	I-2
Absolute Binary Cards (P8)	F-7	X (External) Format	I-3
G. ANSI TAPE LABEL FORMATS	G-1	End-of-Tape/End-of-Reel Conditions	I-3
Required Labels	G-1	J. NAM/CCP TERMINAL DEFINITION COMMANDS	J-1
VOL 1 - Volume Header Label	G-2	Introduction	J-1
HDR1 - First File Header Label	G-4	Terminal Definition Command Format	J-3
EOF1 - First End-Of-File Label	G-8	Restrictions	J-4
EOV1 - First End-Of-Volume Label	G-9	Terminal Definition Commands (AB) - Abort Output Block Character	J-4
			J-5

(AR) - Automatic Character Recognition	J-5	L. PASSIVE PROCEDURES	L-1
(BF) - Blocking Factor	J-6	.PROC Directive	L-1
(BR) - Break Key as User Break 1	J-6	Procedure and Parameter Descriptions	L-2
(BS) - Backspace Character	J-7	Parameter Matching	L-2
(B1) - Interruption or User Break 1 Character	J-7	Order-Dependent Parameter Matching Mode	L-2
(B2) - Termination or User Break 2 Character	J-8	Order-Independent Parameter Matching Mode	L-6
(CH) - Display Terminal Characteristics	J-8		
(CI) - Carriage Return Idle Count	J-9		
(CN) - Cancel Character	J-9		
(CP) - Cursor Positioning after Input	J-10	M. PAPER TAPE OPERATIONS	M-1
(CT) - Network Control Character	J-10	General Description	M-1
(DL) - Delimiters for Single-Message Transparent Input Mode	J-11	Teletypewriter	M-1
(EB) - End-of-Block Character	J-12	Control Characters	M-2
(EL) - End-of-Line Character	J-13	Input Lines	M-2
(EP) - Echoplex Mode	J-14	Punching Tape Off-Line	M-3
(FA) - Full-ASCII Input Mode	J-14	Tape Mode	M-3
(HC) - Host Connection	J-15	Punching a Tape On-Line	M-4
(HD) - Display of Host Nodes	J-15	Corrections	M-4
(HN) - Host Node Selection	J-17		
(HS) - Host Selection	J-17		
(IC) - Flow Control for Input Devices	J-18	N. MASS STORAGE DEVICE STATISTICS	N-1
(IN) - Input Device and Transmission Mode	J-18		
(LI) - Line Feed Idle Count	J-20		
(LK) - Lockout of Unsolicited Messages	J-20	O. INTERACTIVE TRANSFER FACILITY (ITF)	O-1
(MS) - Message to Network Operator	J-21	Selecting the ITF Application	O-1
(OC) - Flow Control for Output Devices	J-21	Selecting a Remote CYBER 200 System	O-1
(OP) - Output Device Selection	J-21	Processing Your Job on the Remote CYBER 200	O-2
(PA) - Parity Processing	J-22	Terminating Your ITF Session	O-2
(PG) - Page Waiting	J-23		
(PL) - Page Length	J-23		
(PW) - Page Width	J-24		
(RC) - Reset Terminal Characteristics	J-24	P. FILE TRANSFERS USING XMODEM	P-1
(SE) - Special Editing Mode	J-24	XMODEM Command	P-1
(TC) - Terminal Class Command	J-25	Using XMODEM in Prompting Mode	P-3
(TM) - Terminating a Terminal-Host Connection	J-26	Sending a File from a CYBER	P-3
(XL) - Multimessage Transparent Mode	J-27	Receiving a File from a Micro	P-4
		Configuration File	P-4
		File Transfer Notes	P-6
		Text Files	P-6
		CYBER Binary	P-6
		Micro Binary	P-6
		Limitations and Considerations	P-7
K. DELIMITING AND TRANSMITTING TERMINAL INPUT	K-1		
Physical End-of-Line (Line Feed)	K-1		
Logical End-of-Line (Carriage Return)	K-1		
Message Transmission	K-1		

INDEX

FIGURES

1-1 Central Memory Allocation	1-3	7-4 Resource Commitment Processing (Simplified)	7-53
2-1 Logical Structure of Files	2-2	15-1 Random Access File Structure	15-2
2-2 NOS and CRM File Terminology	2-3	15-2 LIBEDIT Input and Output	15-18
2-3 Sample Card File Structure	2-6	15-3 User Library Structure	15-33
2-4 Use of ANSI Labels	2-8	15-4 Maximum Size of Global Library Set	15-43
3-1 FORTRAN Compile and Execute Deck	3-2	F-1 Examples of Coded Card Conversion	F-4
3-2 Submitting Jobs to Remote Hosts	3-6.1	H-1 Carriage Control Tape Format (Standard Paper Length; 6LPI)	H-5
4-1 Procedure File Structure	4-3	H-2 Carriage Control Tape Format (Short Paper Length; 8LPI)	H-6
4-2 TESTPN with Valid Entry	4-5	L-1 Keyword Substitution in Two Procedures	L-5
4-3 TESTPN with Invalid Entry	4-5	L-2 Keyword Substitution in Nested Procedures	L-9
4-4 Summary of Checklist Entries	4-15		
4-5 Calling a Procedure	4-41		
5-1 NOS Command Processing Flow	5-6		
7-1 QUEUE7 Execute Queue Sample Listing	7-47		
7-2 QUEUE7 Input Queue Sample Listing	7-48		
7-3 QUEUE7 Output Queue Sample Listing	7-49		

TABLES

2-1 Physical File Structure on Different Devices	2-5	A-3 ASCII to 6/12-Bit Display Code Conversion	A-10
2-2 Data Representation on Magnetic Tape	2-7	A-4 Nine-Track ASCII Coded Tape Conversion	A-13
3-1 User Job Access Levels	3-15	A-5 Nine-Track EBCDIC Coded Tape Conversion	A-14
4-1 Procedure Directives	4-10	A-6 Seven-Track Coded Tape Conversions	A-15
4-2 Programmable Function Keys for Procedures	4-25	H-1 Carriage Control Characters for 512/580 and 5870 Printers	H-3
4-3 Help on Procedure Calls	4-48	H-2 Carriage Control Characters for 585 Printers	H-4
9-1 Range of Permissible Formats for the COPY Command	9-6	J-1 Default Terminal Definitions	J-28
9-2 Code Set-Line Terminator Reference Chart	9-24	J-2 Parameter Ranges for Terminal Definition Commands	J-29
9-3 Compatible File Structures for the VERIFY Command	9-67	K-1 Default Message Delimiters and Transmission Keys	K-2
10-1 Access Mode Granted When Attaching a Currently Attached Direct Access File	10-9	L-1 Parameter Substitution in Order-Dependent Mode	L-4
A-1 Character Sets for Interactive Jobs	A-5	L-2 Parameter Substitution in Order-Independent Mode	L-8
A-2 Character Sets for Batch Jobs	A-7		

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 you 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 180, 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. Each 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.

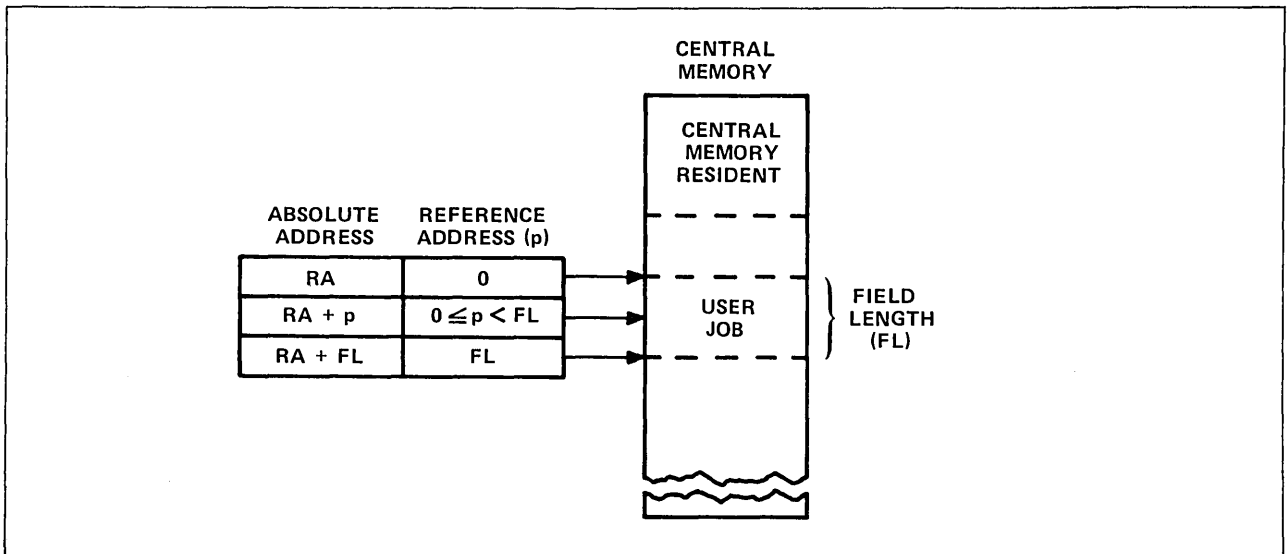


Figure 1-1. Central Memory Allocation

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 multmainframe configuration.[†]

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

[†]Model 176 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) or low-speed port (LSP).
- 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
- 533/536 Printer
- 537 Printer
- 580-12, 580-16, and 580-20 Line Printers
- 585 Printer
- 5870 Non-Impact Printer
- 834/836 Disk Storage Subsystem
- 844-21 Disk Storage Subsystem
- 844-41/44 Disk Storage Subsystems
- 885-11/12 Disk Storage Subsystems
- 885-42 Disk Storage Subsystem
- 887 Disk Storage Subsystem
- 895-1/2 Disk Storage Subsystem
- 819 Disk Storage Subsystem
- 9853 Disk Storage Subsystem
- 7990 Mass Storage Extended Subsystem (MSE)
- 639, 667, 669, 677, 679, and 698 Magnetic Tape Units

255x Network Processing Units

26xx Device Interfaces

380-170 Network Access Device

6683 Satellite Coupler

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.

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 180, CYBER 170, CYBER 70, and 6000 Computer Systems.
- Hardware instructions are interpreted directly by the computer. Each hardware instruction is a binary number. Most users are rarely concerned with instructions written at this level except when program debugging requires interpreting 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. 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.

† There are differences in logical structure that depend on device residence, but these differences are of interest only to a COMPASS programmer. These differences are described in Volume 4, Program Interface.

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.

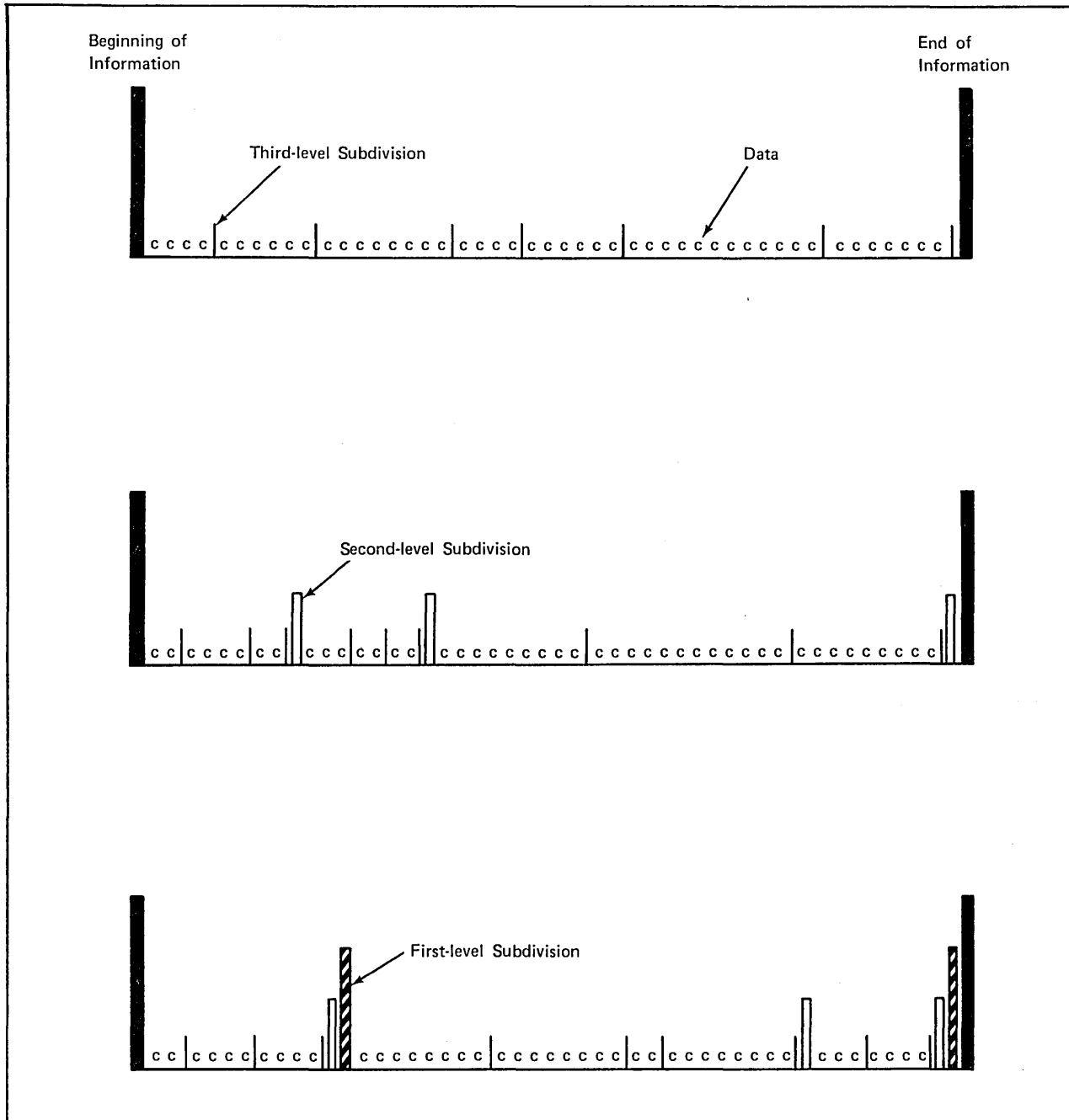


Figure 2-1. Logical Structure of Files

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

Record and File Mark Indicators						
Device	PRU Size	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.	
	I (Internal)	512 CM words; each PRU includes a 48-bit terminator.	If labeled, tape mark following HDR1 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 HDR1 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.
Magnetic tape††††	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 HDR1 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 HDR1 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 HDR1 label of the first volume. In the case of an ANSI multifile, BOI is the HDR1 label associated with the beginning of the multifile.
 ††††For more information, refer to section 12 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.†

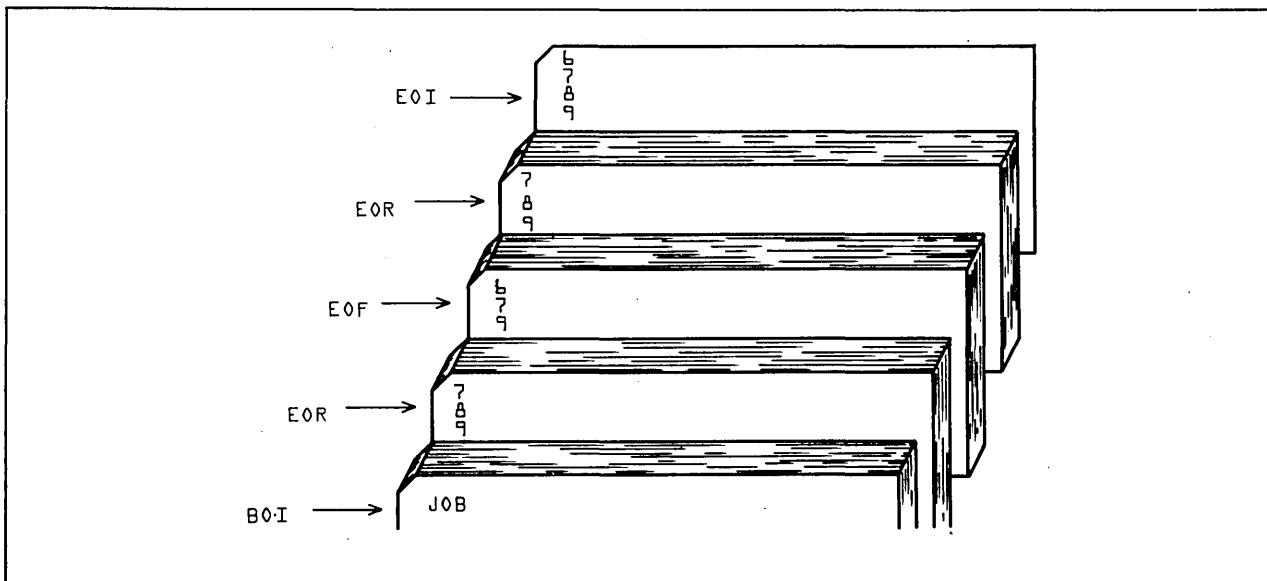


Figure 2-3. Sample Card File Structure

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 coded mode or binary mode. Coded mode operations involve data conversion from NOS-supported character sets to external character sets as shown in table 2-2. Coded mode is applicable largely for S and L tape formats. Binary mode operations copy files with no character set conversion and can be used for any tape format.

The term coded mode should not be confused with coded line. A coded line is defined as a line of 160 characters or less, terminated by a zero byte. Similarly, coded and binary mode should not be confused with coded and binary files. A coded file is a NOS system file coded in 6-bit display code, 6/12-bit display code, or 7-bit ASCII code. A binary file is simply a file containing binary information. Coded files can be copied in coded or binary mode; binary files can be copied only in binary mode.

The only NOS commands recommended for coded mode operations are COPY, FCOPY, and TCOPY. Any of the copy commands can be used for binary mode copies. For CYBER Record Manager (CRM) files, the CM parameter on the FILE command is used to select coded or binary mode. For FORTRAN programs, use the BUFFER OUT(n,1) statement to select binary mode and the BUFFER OUT(n,0) statement to select coded mode. For COBOL programs, use either the RECORDING MODE IS BINARY statement or the RECORDING MODE IS CODED statement.

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. Table 2-2 shows how the data representation varies with these factors.

Table 2-2. Data Representation on Magnetic Tape

Number of Tracks	Binary Mode	Coded Mode
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.

Tape Density

You can select 200- (read only), 556-, or 800-characters-per-inch (cpi) 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. Any tape marks occurring after the first tape mark are handled as appropriate for the tape format (refer to table 2-1).

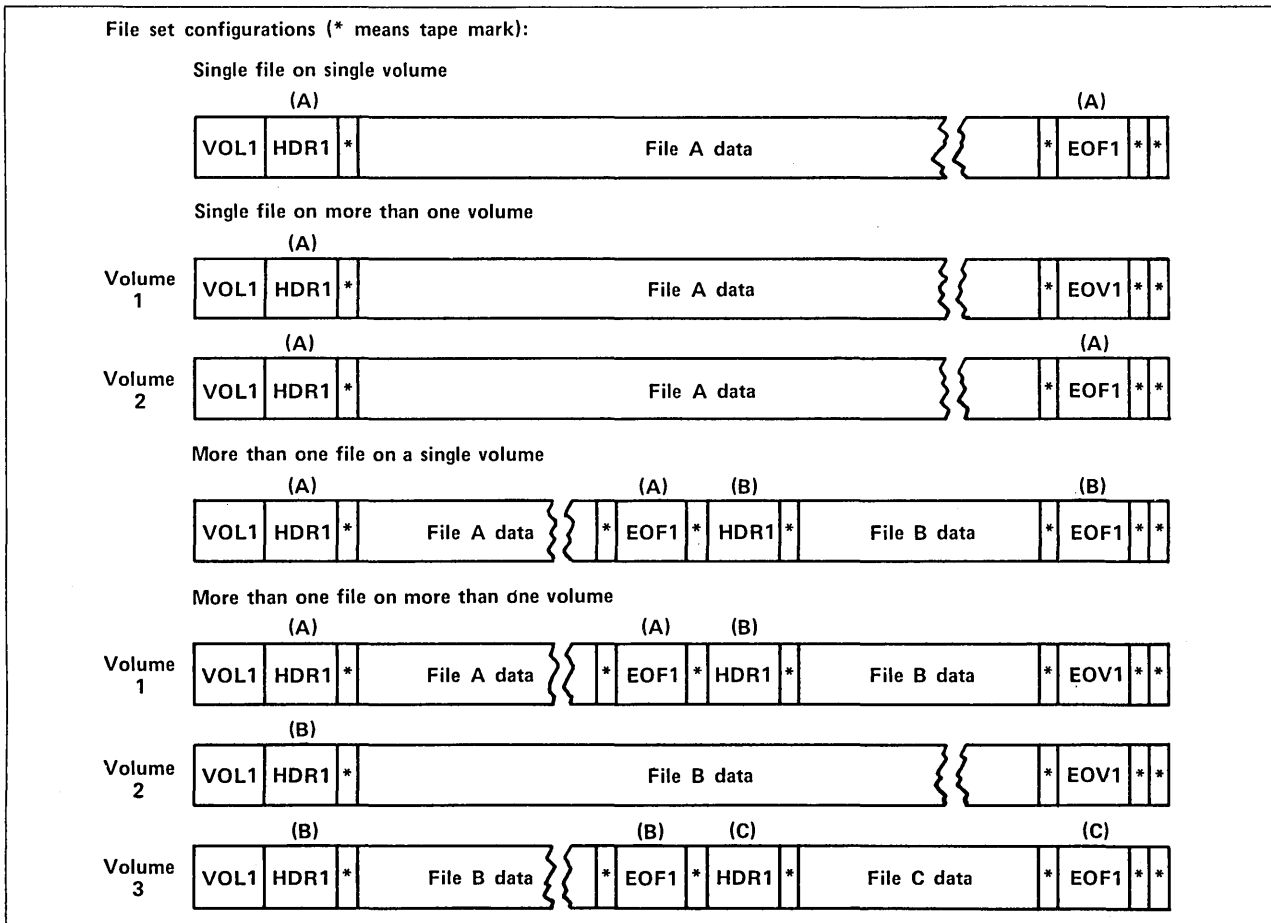


Figure 2-4. Use of ANSI Labels

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.

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.
HDR1	Beginning of information. If the file continues on to another volume, the HDR1 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 as follows:

<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
S	Labeled or unlabeled	Binary Coded 7-track Coded 9-track	Odd	User-selected; default is < 18 frames
			Even	
			Odd	
L	Labeled or unlabeled	Binary Coded 7-track Coded 9-track	Odd	User-selected; default is < 18 frames
			Even	
			Odd	
F	Unlabeled (labels read as data)	Binary Coded 7-track	Odd	User-selected; default is < 18 frames
			Even††††	

†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.

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

NOS terminates all blocks on I format tapes 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 12-bit 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.

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. 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, MFQUEUE, 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 automatically queues files named OUTPUT, PUNCH, PUNCHB, and P8; deferred routed files; and other system-reserved output files if they are assigned to the job. In interactive jobs, however, the files OUTPUT, PUNCH, PUNCHB, and PB are not automatically queued.

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 queueing files originating from the terminal. You can determine the default destination of batch output for your jobs by using the ENQUIRE command and specifying either the JSN=jsn or UJN=ujn parameter. ENQUIRE shows the default destination in the DS field of its display. RB indicates remote batch routing and BC indicates routing to the batch devices at the central site.

You can override the default queueing 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 the COPYSBF command and to appendix H). Files with uppercase and lowercase characters require special processing (refer to the FCOPY command). 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 when you 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 may, if you permit, 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, update, 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 alternate storage. Unless you specify otherwise, all permanent files are saved in your family. Refer to the NOS 2 Analysis 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 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.

ALTERNATE STORAGE

Disk is the usual residence of permanent mass storage files. However, your site may choose to move some files from disk to alternate storage (either to a cartridge alternate storage subsystem such as MSE or to tape alternate storage). Alternate storage is well suited for the storage of files that are accessed infrequently. Accessing a file residing on alternate storage takes extra time, because the file must be retrieved and copied (staged) to disk. You can specify the preferred residence of your permanent files with the PR parameter on the DEFINE, CHANGE, or SAVE commands. You can ascertain the residence of your files with the CATLIST command. In the CATLIST output, file names surrounded by parentheses indicate that these files reside on alternate storage rather than on disk.

Usually, when attaching a file from alternate storage, the system suspends the job until the file has been staged and assigned to the job. (You can determine the status of your staging request for an alternate storage file with the ENQUIRE or cE command.) However, if you specify the RT parameter on the ATTACH, GET, or OLD command, the job will be allowed to continue processing while the file is being staged to disk. You must then issue a second ATTACH, GET, or OLD command to assign the file to your job after staging. If you specified the RT parameter you can then check whether 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, CHANGE, or SAVE command, you can choose to have a tape backup copy of your file kept even if the file resides on alternate storage. You also can choose to have the alternate storage file copy serve as a backup, or you can require no backup copy of your 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 for indirect access files.

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 record. Section 15 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 two commands are processed by system routines that require no additional user input. The third 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. 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. However, you may group commands in a procedure and those commands are executed together when you call the procedure.

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 HELLO7, LDI, MFQUEUE, 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 and SUBMIT commands. Section 9 describes the MFQUEUE 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.

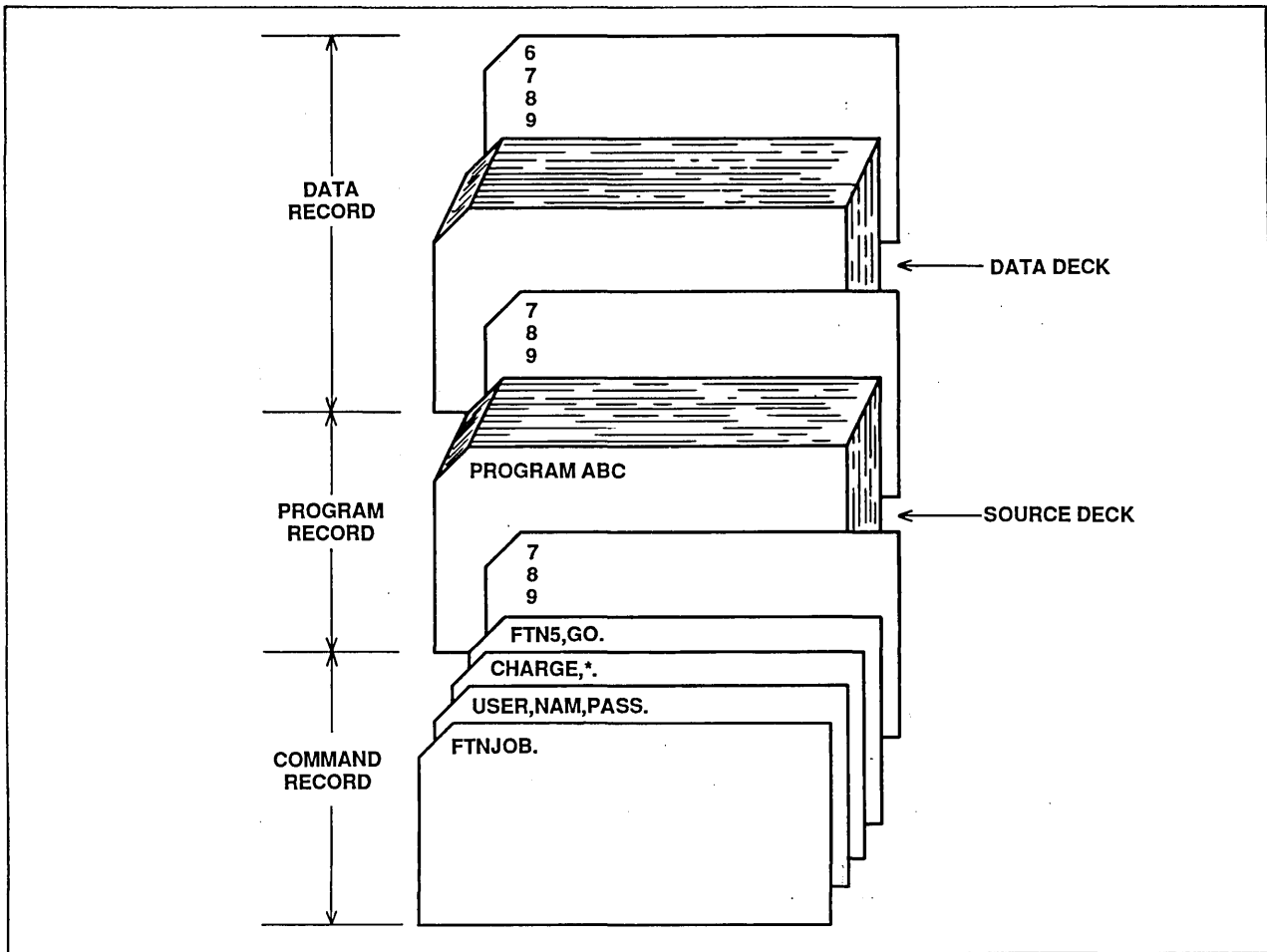


Figure 3-1. FORTRAN Compile and Execute Deck

INTERACTIVE JOBS

To enter an interactive job, you must use the Interactive Facility (IAF). Volumes 1 and 2 of this reference set give a more detailed description of login and logout procedures for interactive jobs. Section 16 describes terminal and input/output conventions.

Login

Before you can login to a host computer, you must establish a connection with the network software. If you need information on how to connect to Communications Control Program (CCP) software, refer to Volume 2, Guide to System Usage. If your network uses CDCNET, refer to the CDCNET Access Guide.

Once you have established a network connection, you can begin the login sequence.

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

```
WELCOME TO THE NOS SOFTWARE SYSTEM.  
COPYRIGHT CONTROL DATA 1978,198X.  
  
YY/MM/DD. HH,MM,SS TO2A57  
NETWORK OPERATING SYSTEM          NOS 2  
FAMILY: ,username,password,iaf  
YOUR PASSWORD WILL BE EXPIRED YY/MM/DD  
  
JSN: AEYT, NAMIAF  
  
READY.
```

The second line of the login sequence gives the original copyright date of NOS software and the copyright date of your version of NOS. The third line gives the current date and time, and the name of your terminal.

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

If you have recoverable jobs in the system, the system initiates a recovery dialogue with you immediately after it logs you in.

NOTE

Using a network to recover a job other than the one you were using before the job was disconnected may not always work. For this reason, we recommend that you always use the same network to recover a job.

† This manual assumes that your system does not require the entry of a personal identifier. For more information about the personal identifier refer to Volume 2 of the NOS 2 Reference Set.

The following example shows a login sequence where you have recoverable jobs.† Again, your entries appear in lowercase characters.

WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198X.

YY/MM/DD. HH.MM.SS T01A75
NETWORK OPERATING SYSTEM NOS 2
FAMILY: ,username,password,iaf
YOUR PASSWORD WILL BE EXPIRED YY/MM/DD

JSN: ADYE, NAMIAF

RECOVERABLE JOB(S)

JSN	UJN	STATUS	TIMEOUT
AARQ	AN2A	SUSPENDED	26 MIN.
AASF	AN2A	SUSPENDED	28 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 disconnects you from the host. 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  
CHARACTERS=xxxxxKCHS  
IAF CONNECT TIME hh.mm.ss.  
LOGGED OUT.
```

This display is followed by additional logout information provided by network software.

† This manual assumes that your system does not require the entry of a personal identifier. For more information about the personal identifier refer to Volume 2 of the NOS 2 Reference Set.

The following is a description of the variable items in the logout display:

<u>Item</u>	<u>Description</u>
username	User name you entered during login.
hh.mm.ss.	The first occurrence of hh.mm.ss. indicates the time of logout. The second occurrence indicates the length of time your terminal was connected to IAF.
jsn	Your job sequence name.
s.sss	A measure of the system resources used while connected to IAF.
xxxxxxxx	A count of the total number of input and output characters read from, or written to, your terminal.
ct	The network control character for your terminal. (The network control character for specific terminal models are listed in Table J-1.)
had	The host availability display (HAD). The HAD appears if you have enabled it with the HD terminal definition command (refer to appendix J) or if you are operating in a multihost environment.

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 dial-up terminal to the system, the system automatically disconnects the terminal after a site-determined period of time (2 minutes is the default). 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 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, TVF, PNI, 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.

SUBMITTING JOBS TO REMOTE HOSTS

Some sites use multiple host network configurations in which you can log in to a host mainframe and then submit jobs to other mainframes (remote hosts) in that computer system. There are various ways in which sites can configure these computer systems, and the means by which you submit jobs to a remote host depend on your system configuration, the type of remote host, and the nature of the jobs. In all cases, however, you must have special authorization to submit such jobs (refer to the LIMITS command in section 7).

In general, you can submit a batch job to a remote host by specifying the remote host with the ST=lid parameter of the Job command or the ROUTE command. If the remote host is a SCOPE 2 system and the batch job requires interactive input or output, use the HELLO7 command. If the remote host is connected through the Queued File Transfer Facility (QTF) and the batch job has special routing requirements (requirements not provided by ROUTE parameters), use the MFQUEUE command. If you want to retrieve or change the attributes of permanent files on such a remote host, you need not submit a complete batch job. You can use the MFLINK command from a job on your host mainframe.

If you want to connect your interactive terminal to a Virtual Storage Operating System (operable on a CYBER 200 computer system), select the Interactive Transfer Facility (ITF) during login or with the BYE, GOODBYE, HELLO, LOGIN, or LOGOUT command.

In a dual-state system, you can submit a batch job from the NOS system to the NOS/VE system using the ROUTE command. The ROUTE ST=lid parameter routes a NOS file to the NOS/VE batch input queue. The file must be written in 6/12-bit display code.

Figure 3-2 summarizes the methods of submitting jobs to remote hosts.

Command/Application	Type of Job Being Submitted	Special Host Type/ Connection Type
Job Command (ST=lid parameter) ROUTE Command (ST=lid parameter)	Batch	No special requirements.
HELLO7 Command	Batch (with interactive I/O)	SCOPE 2 system.
MFQUEUE Command	Batch (with special routing requirements)	RHF (LCN, CDCNET, or CCP).
ITF Application	Interactive	VSOS system (CYBER 200).
MFLINK Command	Part of a batch or interactive job (permanent file manipulations)	RHF (LCN, CDCNET, or CCP).

Figure 3-2. Submitting Jobs to Remote Hosts

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.

The four origin types and their system symbols are as follows (system symbols are described in section 6):

<u>Origin Type</u>	<u>Symbol</u>
System	SYO
Interactive	IAO
Local batch	BCO
Remote batch	RBO

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 so authorized (refer to the LIMITS command), you can also initiate jobs using the HELLO7, MFQUEUE, 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

Every job in the system is assigned a service class. This service class determines the job's scheduling priority as the job flows through the system. You can make this service class assignment or you can let the system assign it a default service class based on the job's origin type and your validation limits (the privileges your site has granted you).

Unless changed by your site, the system has the following 14 service classes:

<u>Service Class</u>	<u>Two-character Mnemonic</u>	<u>One-character Mnemonic</u>
Deadstart	DS	A
System	SY	S
Local batch	BC	B
Remote batch	RB	R
Interactive	TS	T
Detached interactive	DI	D
Network supervisor	NS	N
Subsystem	SS	X
Maintenance	MA	M

<u>Service Class</u>	<u>Two-character Mnemonic</u>	<u>One-character Mnemonic</u>
Communication task	CT	C
Installation class 0	I0	0
Installation class 1	I1	1
Installation class 2	I2	2
Installation class 3	I3	3

The scheduling priority associated with each service class is site-determined. You can ascertain how your site has prioritized the service classes you are authorized to use for each origin type by using the CLASS command. To determine the service class of a job, use the ENQUIRE command. The system denotes the service class with a one-character mnemonic in the SC field of the ENQUIRE output.

Your site allows you to use only a subset of the service classes. The LIMITS command shows you which service classes you can use. Further, your site can restrict the use of certain service classes to jobs of certain origin types. The CLASS command displays this information.

You can assign a service class to a job in the following ways:

- Use the CHVAL command to specify default assignments for your jobs according to origin type. The system uses one of these default assignments if you do not otherwise make a service class assignment in a job. The LIMITS command shows what your default service class assignments are for each origin type.
- Use the SC=class or P=priority parameter of the Job command for batch jobs.
- Use the CLASS command to change the service class of your current job.
- Use the SCL=scl parameter of the ROUTE command for jobs to be initiated by that command.

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 (a maximum of seven characters) 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.

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

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.

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 = 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.	60000	60000	0	3200	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	2500	GET command retrieves copies of an absolute program and a relocatable program.
RFL,40000.	60000	60000	0	2300	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	2300	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

Batch 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).

For interactive jobs, file INPUT* is present at the beginning of the session and cannot be returned with a RETURN command. You can create and manipulate a file called INPUT at any time, although various products, such as the Full Screen Editor, require special processing in order to handle files named INPUT or OUTPUT as data files.

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. You cannot resume jobs suspended in this fashion from the point of suspension; they enter exit processing.

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. The ENQUIRE,U command gives the number of commands you have entered in 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 higher 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 a NOEXIT command is in effect, 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, RECOVR, 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.

†Not applicable to model 176.

SECURITY FEATURES

A NOS computer system provides extensive protection of information contained in your jobs and files. Most notably, it protects the central memory associated with your active jobs and it enforces job and file access controls.

MEMORY PROTECTION

A job cannot dump or directly change the contents of the job field length immediately after a protected command or a user program. The following is a list of protected commands and a list of load/dump memory commands:

Protected Commands

APL	COPYCF	IF or (IFE)	PASSWOR	SECHDR
APPEND	COPYCR	LABEL	PERMIT	SET
ASSIGN	COPYEI	LDSET	PROFILE	SETFS
ATTACH	COPYSBF	LIBEDIT	PURGE	SKIP
BEGIN	COPYX	LIBLOAD	RECOVER	SORT
BLANK	DAYFILE	LIBTASK	REDUCE	SUBMIT
CATALOG	DEFINE	LIMITS	REPLACE	TCOPY
CATLIST	DISPLAY	LOAD	REQUEST	UNLOAD
CHANGE	EDIT	MAP	RESOURC	UPROC
CHARGE	ELSE	MFLINK	RESTART	USER
CHVAL	ENDIF	MFQUEUE	RETURN	VERIFY
CLASS	ENDW	NEW	REVERT	VFYLIB
CLEAR	ENQUIRE	OLD	REWIND	VSN
COPY	EXECUTE †	OUT	SATISFY	WHILE
COPYBF	FSE	OVWRITE	SAVE	
COPYBR	GET	PACKNAM	SCOPY	

Load/Dump Memory Commands

CKP	DMD	DMP	LBC	PBC	WBR
DMB	DMDECS	DMPECS	LOC	RBR	

In an interactive job, you cannot enter a load/dump memory 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.

This form of memory protection does 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.

SYSTEM AND FILE ACCESS CONTROLS

NOS operates in either secured or unsecured mode, depending on the option selected by your site. On an unsecured system, the primary control of system access is based on user names and user passwords. NOS may also require certain users to enter a personal identifier in addition to their user name and password to gain system access. Additionally certain users may also be restricted to a single terminal session at a time.

† This applies to EXECUTE when used as a loader command. It does not apply to EXECUTE when used to select the execute subsystem in an interactive job.

The system further restricts the use of certain system resources (for example, applications and special commands) to those explicitly granted that privilege by your site. NOS enforces file access controls based on user names, file passwords, and permissions granted by the owner of the file. The controls based on file passwords and permissions are discretionary; they are set at the discretion of the file owner.

In a secured system, NOS enforces an additional set of access controls based on security access levels and categories. These controls limit access to information based on the user's validated clearance level and need-to-know categories versus the security access level and category markings associated with each file. These mandatory controls take precedence over the discretionary controls, which also remain available.

You must be validated to use one or more security access levels and zero or more security access categories. Every file on a secured system has a single security access level and may have one or more access categories associated with it. To access a file, the user job must be validated for the file's security access level and for all of the file's access categories.

SECURITY ACCESS LEVELS

A security access level limits the disclosure of sensitive information to persons who are authorized to access information at the level of sensitivity indicated by that access level. A secured system supports up to eight security access levels. The number of access levels to be used, the names of those levels, and the degree of sensitivity associated with each level are selected by your site. The access levels are ordered so that the most sensitive information is associated with higher access levels. (The access levels have numerical values of 0 through 7; the corresponding released default names are LVL0 through LVL7.) You are validated for a set of security access levels (normally a contiguous range of levels, such as LVL0 through LVL4).

The system associates a set of access levels with each job. This set is the intersection of your validated set of access levels, the range of levels currently valid for the system, the range of access levels allowed for your job's origin type, and, if specified by the system, the range of access levels allowed over your communication line to the computer. For interactive jobs, your job's initial access level is the lowest level in the set of access levels for your job. For batch jobs, the initial access level is the access level of the local file that initiated the job.

Example 1: You are validated to use levels LVL0 through LVL5, the system is currently validated for levels LVL0 through LVL7, your job origin type has an access level range of LVL1 through LVL4, and your communication line has an access level range of LVL0 through LVL3. As shown in table 3-1, your job will have an access level set of LVL1 through LVL3 and the job's initial access level will be LVL1. If this job is initiated via a ROUTE or SUBMIT of a local file with access level LVL2, that will be the job's initial access level.

Table 3-1. User Job Access Levels

	LVL0	LVL1	LVL2	LVL3	LVL4	LVL5	LVL6	LVL7
User	X	X	X	X	X	X		
System	X	X	X	X	X	X	X	X
Job Origin		X	X	X	X			
Communication Line	X	X	X	X				
User Job		X	X	X				

Each file has a security access level. Unless you specify otherwise, each file you create will have the same access level as your job at the time you create the file. You can always raise the access level of your files (refer to the SETFAL and SETPFAL commands) to any level for which your job is validated, but you must have special permission to lower the access level of your files. However, the access level of direct access files or tape files cannot be changed while they are assigned to your job. If the access level of such a file is lower than that of your job, you can always read information from the file but can only write information to it if you have special authorization. The access level of a local file is automatically raised to the current job access level whenever an operation is performed on the file (this includes your primary file).

Your job's access level is automatically raised whenever you read information from a file that has a higher access level than your job's current access level (within your job's validated set of access levels). You can always raise your job's access level using the SETJAL command to any level for which your job is validated, but you must have special permission to lower the access level of your job.

Example 2: In the job referred to in example 1, you attach a file that has access level LVL3 and read information from that file. The access level of your job is immediately raised from LVL1 to LVL3.

Example 3: In the same job, you attempt to set your job's access level to LVL5 using the SETJAL command. This attempt fails because LVL5 is not in your job's validated set of access levels.

Unless you are specifically validated, a secured system does not allow you to write information from a file with a higher access level into a file with a lower access level.

Example 4: You attempt to copy a portion of a file whose access level is LVL2 to a file whose access level is LVL3. A secured system allows this operation.

Example 5: You attempt to copy a portion of a file whose access level is LVL3 to a file whose access level is LVL2. A secured system does not allow this operation (unless you possess special validation).

On an unsecured system, you may set access levels on your files (for informational purposes), but the system does not use those levels to restrict access to the files, nor does it propagate those levels to your job or to other files. No job has an access level on an unsecured system.

SECURITY ACCESS CATEGORIES

A secured system supports up to 32 security access categories. The number of categories to be used and the names of those categories are selected by your site. The access categories have numerical values of 0 through 31; the corresponding released default names are CAT00 through CAT31. Each user is validated for subset of the site-defined access categories. Your job's access category set is the intersection of this subset and the set of categories currently valid for the system.

When you create a permanent file (using the SAVE or DEFINE command), the file is automatically assigned all of the access categories in your job's access category set. You have the choice to explicitly assign all, any, or none of these categories to each of your permanent files (refer to the SETPFAC command). Access categories restrict the access of your permanent files to those users who are validated for all of the categories you have set for the files.

Example 6: You are validated for the categories CAT01, CAT06, CAT18, and CAT22. Using the SETPFAC command, you assign the categories CAT06 and CAT22 to permanent file ABC, which has an access level of LVL3. You then make file ABC public. Any user in your family who is validated for access level LVL3 and the categories CAT06 and CAT22 can then access file ABC.

On an unsecured system, you may assign access categories to your permanent files for informational purposes, but the system does not use those categories to restrict access to the files.

RESPONSIBILITIES FOR DATA SECURITY

In a secured system, security access levels and categories are checked whenever a user accesses a file, begins a job, or attempts to alter the security parameters of a file or job. This ensures that only those persons who are validated for the access levels and categories you have assigned to your files can access them. A secured system thus provides continuous protection of your data.

Although the system will automatically assign an access level and an access category set to each file you create, you are responsible for ensuring that the access level and access category set assigned to each of your files is appropriate. Your site should provide guidelines for the use of the different access levels and categories it has established.

All of the NOS security mechanisms depend on the protection of your user password. NOS provides the following mechanisms to help protect your password during interactive login:

- You should always use the secure login procedure (described in section 8) to ensure that you are communicating with the system-supplied login mechanism.

- If possible, you should set your terminal to full duplex and enable echoplex mode (refer to the EP terminal definition command). The network temporarily disables echoplex when you enter your password during login. If you use the abbreviated login sequence, all fields in the line containing your password will be masked. This NOS security feature prevents display of your user password at your terminal.
- A warning message stating when your user password is going to expire is displayed at your terminal and is issued to the job dayfile.

Although NOS encrypts your user password after login, 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 and charge numbers.

SECURITY CONFLICT PROCESSING

When your job attempts to perform an operation that could violate system security, the system identifies the operation as a security conflict and issues a SECURITY CONFLICT message. A security conflict causes the job step to be aborted and decrements your security count (you are given a specific security count by the site security administrator). If your security count is decremented to zero, your current job or interactive session is terminated without EXIT processing. You are not allowed any additional jobs or logins until the security administrator resets your security count.

On an unsecured system, a security conflict occurs in the following cases:

- An incorrect secondary USER command.
- A job file submitted with a SUBMIT or ROUTE command has an incorrect USER command.

On a secured system, a security conflict arises in the following cases in addition to those listed above:

- An attempt to set an invalid access level on a file or job.
- An attempt to access one of your permanent files from a job that is not validated for the access level or access category set of the file.
- An attempt to write data on a tape file or an attached direct access file when the access level of your job is higher than the access level of the file.
- An attempt to set an access level for a file or job which is lower than the current access level of the file or job.†

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.

†This action does not constitute a security conflict if you have special authorization to do so (refer to the LIMITS command).

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.
 - Application activity:

hh.mm.ss.UEAC, 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 Administration Handbook).

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 H).

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 automatically queue files OUTPUT, PUNCH, PUNCHB, and P8 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 NOS procedure is a sequence of NOS commands, residing in a separate record or file, that performs a specific task or sequence of tasks. A procedure header directive identifies the file as a procedure file, and a number of other procedure directives allow you to control additional procedure processing options.

Once you have created a procedure and stored it in a permanent file, the procedure is available to you at any future time. To execute the procedure, you must call the procedure using an appropriate form of the BEGIN command, which we describe later in this section. Procedures that you create yourself can be called from a local file, a permanent file (direct or indirect access), or from a local or global library set. Your site may also provide procedures of general use that you can call from system libraries or from user name LIBRARY. A procedure can be called from another procedure up to a maximum of 50 nested procedures.

A procedure called from a batch job functions much like a program subroutine. You can pass parameters to the procedure on the procedure call (BEGIN command). The procedure accepts any data passed to it, performs its processing functions, and returns control to the calling job or procedure.

When called from an interactive job, a NOS procedure functions like a user-defined system command. Once you have made the procedure file local to your job, you can execute the procedure simply by entering the procedure or file name, followed by any parameters you may have defined for the procedure.

Although NOS supports both interactive and noninteractive procedure formats (noninteractive procedures are also called passive procedures), any NOS procedure can be called from either a batch or interactive job. The interactive procedure formats described in this section differ from the noninteractive format (described in Appendix L) in that the interactive formats provide the end user with various types of interactive help and parameter prompts.

The major features of NOS interactive parameter formats are:

Two Interactive Formats

As the procedure writer, you can choose between the parameter-prompting (*I) or the menu (*M) interactive formats. The parameter-prompting format prompts the user to enter required parameter values. The menu format displays a list of possible parameter values and prompts the user to select one of the values by number.

Procedure Prompting

If the user fails to enter a required parameter value, or enters an invalid value, the system prompts the user to enter a correct value.

Interactive Help

Both interactive formats give the user easy access to system-generated, as well as procedure-defined help for a procedure and its parameters.

Parameter Validation

The interactive formats provide an easy way of defining the types of parameter values that can be entered for any particular parameter. The system will not execute an interactive procedure until the user has entered a valid value for each required parameter.

PROCEDURE FILE RESIDENCE AND SEARCH ORDER

A procedure is stored as a separate record on a file. Several procedures, or a collection of procedures and programs, can exist on the same file. Procedures can be stored on local files, indirect access or direct access permanent files, system libraries, or user name LIBRARY. Procedures can also reside on magnetic tape as well as mass storage.

The procedure search order is dependent on the form of the BEGIN command used to call the procedure. For more information on the search order, refer to the BEGIN command description in this section.

PROCEDURE STRUCTURE

A procedure file can be divided into five sections, as shown by the example in Figure 4-1. The first four sections comprise the procedure itself (the procedure begins with the procedure header (.PROC) directive and ends with a REVERT command at the end of the command section). The fifth (Data) section may contain programs, procedures, or other data to be referenced by the procedure.

The first line or lines of a procedure is the procedure header, which consists of a single procedure header (.PROC) directive. The procedure header names the procedure, indicates whether the system will prompt for required parameter values, defines what the prompts will consist of, and defines any restrictions on the types of values that can be entered for each parameter. The .PROC directive and all other procedure directives are listed in summary form in Table 4-1 and are described in detail under Procedure Directives in this section. Procedure directives are identified by the period (.) before the directive name.

The formatting section contains directives which alter the default text and format of user prompts. This section of the procedure body is optional, as the system does provide default text and format for all prompts. If you do specify directives in this section, the formatting directives must immediately follow the procedure header directive.

The help section contains directives and text which define the help to be provided for procedure parameters and for the procedure itself. This section of the procedure is optional, as the system can generate help from the information in the procedure header. If you do specify any directives in this section, you must include a .ENDHELP directive to end the section.

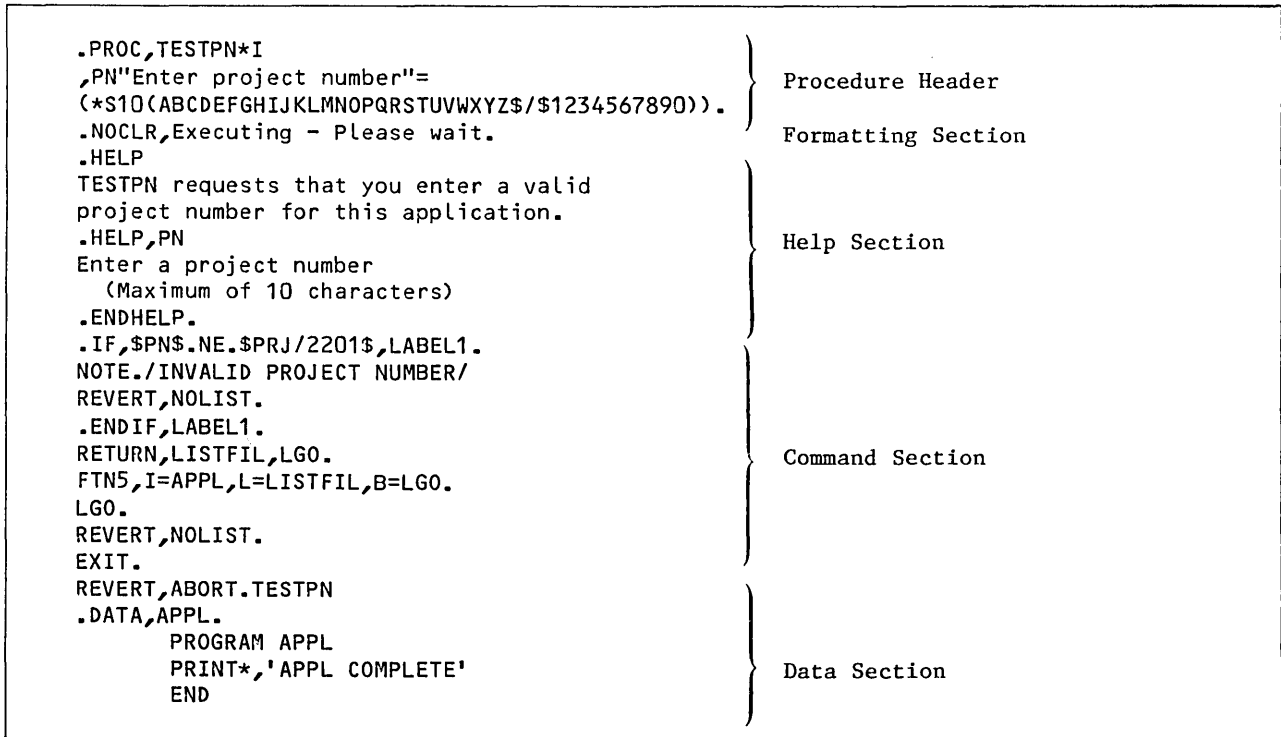


Figure 4-1. Procedure File Structure

The command section consists of procedure directives and system commands you want to be executed. Most system commands are allowed within a procedure. The exceptions are primarily the commands restricted to entry from interactive terminals (refer to section 8).

The data section consists of information you want written to a local file or files so that you can reference it while the procedure executes. You can also reference the data section in the procedure header. A separate .DATA directive identifies each local file to be created.

PROCEDURE PROCESSING

The system performs interactive procedure processing in three stages:

- Interactive parameter entry
- Procedure expansion
- Execution of the command record

INTERACTIVE PARAMETER ENTRY

During interactive parameter entry, the system processes the procedure header directive and all directives in the formatting and help sections.

The system prompts the user for all required parameter values as specified in the procedure header. Each value the user enters is verified against the validation requirements specified in the procedure header. If any value fails to meet the validation requirements, the system prompts the user to enter a corrected value.

The correction prompt and other interactive prompts can be modified by directives in the formatting section.

If the user requests help for the procedure or for a parameter (by entering a ? following the procedure call or following a parameter prompt), the system displays the applicable system default help text, if any, followed by procedure-defined help text.

The interactive parameter entry portion of procedure processing is completed when the user enters a carriage return after having correctly entered all required parameter values. If the user correctly enters all required values on the procedure call, no formatting or help directives (except .NOTE and .NOCLR) are processed, and procedure expansion begins immediately.

PROCEDURE EXPANSION

Procedure expansion is performed only on the procedure command and data sections. During procedure expansion, the system:

- Processes all directives in the command and data section.
- Substitutes parameter values specified by the user, or by the .PROC directive, for each occurrence of the corresponding parameter keyword in the command and data sections.
- Creates a scratch file, called the procedure command record, which contains the NOS commands from the procedure command section as modified by directive processing and parameter substitution.
- Creates a separate local file for each .DATA directive processed during expansion of the procedure.

At completion of expansion processing, the calling user's job has assigned to it a scratch file (the procedure command record), containing NOS commands ready to be executed, and a local file for each .DATA file referenced in the command record.

The procedure command record created during expansion processing is simply the sequence of NOS commands to be executed. Although procedure directives are used to define and create the command record, directives themselves do not appear in the command record since they are not executable system commands.

The procedure command record may not include all system commands from the procedure command section. This is determined by processing of the .IF, .ELSE, and .ENDIF directives. The .IF directive functions similarly to the IF command. In the same way that the IF command determines whether the following sequence of commands will be executed, the .IF directive determines whether the following sequence of commands is written to the procedure command record. In other words, if the expression in the .IF directive is true, the commands placed between .IF and .ELSE or .ENDIF are written to the command record; if the expression is

false, these same commands are not written to the command record. To see the results of the .IF directive, look at figures 4-2 and 4-3. These figures show the command records resulting from valid and invalid entries for the procedure in figure 4-1.

Note that at the end of processing in either figure 4-2 or 4-3, a listing of the user's local jobs would show a file called APPL. This file is created as a result of the .DATA directive in procedure TESTPN from figure 4-1.

```
Procedure Command record:

RETURN,LISTFIL,LGO.
FTN5,I=APPL,L=LISTFIL,B=LGO.
LGO.
REVERT,NOLIST.

Dayfile:

testpn,prj/2201
  APPL COMPLETE
/dayfile,fr=testpn,op=m
10.18.53.TESTPN,PRJ/2201.
10.18.54.RETURN,LISTFIL,LGO.
10.18.55.FTN5,I=APPL,L=LISTFIL,B=LGO.
10.18.55.      62700 CM STORAGE USED.
10.18.55.      0.009 CP SECONDS COMPILATION TIME.
10.18.55.LGO.
10.18.56.      END APPL
10.18.56.      14300 MAXIMUM EXECUTION FL.
10.18.56.      0.007 CP SECONDS EXECUTION TIME.
10.19.21.DAYFILE,FR=TESTPN,OP=M
  USER DAYFILE PROCESSED.
/
```

Figure 4-2. TESTPN with Valid Entry

```
Procedure Command record:

NOTE./INVALID PROJECT NUMBER/
REVERT,NOLIST.

Dayfile:

  testpn,888
  INVALID PROJECT NUMBER

/dayfile,fr=testpn,op=m
10.19.51.TESTPN,888.
10.19.51.NOTE./INVALID PROJECT NUMBER/
10.20.03.DAYFILE,FR=TESTPN,OP=M
  USER DAYFILE PROCESSED.
/
```

Figure 4-3. TESTPN with Invalid Entry

Following are some tools you can use to further control procedure expansion:

- .EXPAND Directive
- Inhibit Character
- Concatenation Character
- .EX Directive

.EXPAND Directive

The .EXPAND directive allows you to suspend expansion processing for one or more lines of a procedure. Inserting a .EXPAND,OFF directive suspends directive processing and parameter substitution for subsequent lines of the procedure until the end of the procedure or until a .EXPAND,ON directive appears.

An especially useful application of the .EXPAND directive is to inhibit expansion of a procedure in a .DATA file. Normally, calling a procedure file initiates expansion of the entire file, including the data section. However, if the data section contains another procedure file, you may not want the .DATA file to be expanded until it is called by a BEGIN command. Inserting a .EXPAND,OFF directive before the .DATA directive prevents the creation of the .DATA file. Inserting a .EXPAND,OFF directive immediately after the .DATA directive causes the .DATA file to be created but remain unchanged by expansion of the original procedure.

Inhibit Character

Like the EXPAND directive, the inhibit character also inhibits parameter substitution or directive processing during procedure expansion. In general, the scope of the inhibit character is more limited than that of the EXPAND directive. The default inhibit character is # (or ≡ if you are using a CDC graphic character set). You can define a different inhibit character using the .IC directive.

Usage rules for the inhibit character are:

- The inhibit character placed before a parameter keyword in the command or data section inhibits substitution of that keyword.
- The inhibit character placed before a procedure directive (including a .DATA directive) in the command or data section inhibits expansion processing of that directive.
- The inhibit character placed before a concatenation character inhibits processing of the concatenation operation; the concatenation character remains intact. In this case, the inhibit character does not prevent parameter substitution of a keyword following the concatenation character.
- The inhibit character placed before a second inhibit character causes the second inhibit character to remain intact and not be recognized as an inhibit character. In this case, parameter substitution of a keyword following the second inhibit character is not prevented.

- Expansion processing strips all inhibit characters from the procedure command record except for those appearing in a sequence of two or more inhibit characters. When they appear in sequence, only the odd numbered inhibit characters in the sequence are acted upon and omitted from the procedure command record.

The inhibit character is often used to allow use of the same character string for both a parameter keyword and value. For example, if an input file parameter is written as #I=I, procedure expansion removes the #, leaves the first I intact, and performs normal substitution on the second I.

If a keyword or directive is subject to expansion processing more than once, you can control when it is processed by inserting one or more inhibit characters before the keyword or directive. In the following example, expansion of procedure PROC1 creates one local file called DATA1 and strips the # from the second .DATA directive. Since the # character is no longer present when PROC2 is executed, PROC2 expansion creates local file DATA2.

```
.PROC,PROC1*I.
.
.
.
REVERT.
.DATA,DATA1.
.PROC,PROC2*I.
.
.
.
REVERT.
#.DATA,DATA2
.
.
.
```

Following is an example of inhibit character usage.

In this example, procedure INHIBIT 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.
```

If procedure INHIBIT is called by the command:

```
begin,inhibit
```

The resulting dayfile segment is:

```
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
```

Concatenation Character

The concatenation character allows parameter substitution for a substring of a character string. The default concatenation character is `_` (or `↵` if you are using a CDC graphic character set).

Usage rules for the concatenation character are:

- Expansion processing removes all concatenation characters from the procedure file except those included in a `$`-delimited literal string and those preceded by a `#` character.
- After removal of a concatenation character, the preceding and following character strings are concatenated by shifting the remainder of the line one character to the left. For example, if `PROG` is a parameter keyword for which the user enters a value of `NEW`, the string `PROG_B` becomes `NEWB` after expansion processing.

Following is an example of concatenation character usage.

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.
```

In the following example, 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=xbf,lfnl=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,lfnl=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
```

.EX Directive

The .EX directive immediately terminates expansion processing and executes the command included in the directive. No further expansion processing is performed, and the procedure command record is not executed.

The most common use of .EX is to initiate immediate processing of a menu procedure option as shown in the following example:

```
.PROC,APROC*M  
,PARM=  
(1"OPT1",  
 2"OPT2",  
 3"OPT3").  
.EX.BEGIN,PARM,LIBFILE.
```

EXECUTION OF THE COMMAND RECORD

At completion of expansion processing, the system increments the procedure nesting level count and begins execution of the procedure command record.

When all commands in the procedure command record have been processed, the system executes a system- or user-supplied REVERT command, decrements the procedure nesting count, and returns control to the calling job or procedure.

PROCEDURE DIRECTIVES

Procedure directives have these general syntax features:

- They all have a period as a prefix character.
- You can use a comma or an opening parenthesis as a separator within directives.
- You can use a period or a closing parenthesis as a terminator. You must use a period or closing parenthesis as a terminator if you append comments to the directive.
- A procedure directive can span more than one line. No line can contain more than 150 6-bit characters† and each line to be continued must either end with a nonblank separator or the succeeding line must begin with a nonblank separator.
- Unless otherwise stated, the system reads the characters you enter in directives as 6-bit display code (uppercase characters only). You can enter the text for help information and prompts in uppercase and lowercase characters (6/12-bit display code).

†This value can be changed during installation.

The detailed descriptions of the procedure directives later in this section include any exceptions to the general syntax features just cited.

Procedure directives can be grouped by function into six basic types. The directives are listed by type in table 4-1.

Table 4-1. Procedure Directives

Directive Type	Directive	Function
Procedure header	.PROC	Specifies the procedure name, defines parameters, and enables parameter prompting.
Formatting	.CORRECT	Specifies the prompt to follow an incorrect parameter entry.
	.ENTER	Specifies the prompt to appear before a parameter entry is made.
	.Fn	Specifies a label for programmable function key Fn.
	.NOCLR	Inhibits the automatic clearing of the screen while the procedure executes.
	.NOTE	Specifies a message to appear on the screen while the procedure executes.
	.PAGE	Specifies the string to precede the page number on the screen.
	.PROMPT	Specifies the text for the general request for input.
Help	.ENDHELP	Marks the end of the help section of the procedure.
	.HELP	Specifies help text for the procedure or a parameter.
Expansion Control	.CC	Specifies the concatenation character to be used in the procedure body.
	.ELSE	Terminates skipping initiated by a matching .IF or initiates skipping until a matching .ENDIF is found.
	.ENDIF	Terminates skipping initiated by a matching .IF or .ELSE.
	.EX	Submits a single command to the system for immediate execution.
	.EXPAND	Terminates or restores procedure expansion for subsequent lines of the procedure.
	.IF	Allows conditional expansion of the procedure body.
	.IC	Specifies the inhibit character to be used in the procedure body.
File	.DATA	Creates a local file.
	.EOF	Inserts an end-of-file mark in a (.DATA) file created by the procedure or in the procedure command record.
	.EOR	Inserts an end-of-record mark in a (.DATA) file created by the procedure or in the procedure command record.
Comment	.*	Identifies comments in the procedure.

PROCEDURE HEADER (.PROC) Directive

Every procedure begins with a procedure header (.PROC) directive. The .PROC directive names the procedure, controls parameter prompting, and defines any restrictions on values that can be entered for the parameters.

The .PROC directive must be the first line of the procedure and must begin in column 1. It can be continued on one or more succeeding lines by ending the continued line, or beginning the continuation line, with a non-blank separator character. The .PROC directive must end with a period (.).

This section describes the interactive and menu formats of the .PROC directive. The noninteractive format is described in appendix L.

Interactive Format

This directive enables interactive processing of a procedure call (refer to Requesting Help on Procedure Calls later in this section). The directive:

- Designates procedure parameters as optional or required.
- Designates permissible values and correct syntax for each parameter through a checklist.
- Describes each parameter to be used in prompting the procedure caller.

Format:

```
.PROC,pname*I"title",p1,p2,...,pn.ck.
```

<u>Parameter</u>	<u>Description</u>
pname	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.
title	Specifies the procedure title. The system assumes that your title is in 6/12-bit display code, which supports uppercase and lowercase characters. The title string can be a maximum of 40 lowercase (12-bit) characters, 80 uppercase (6-bit) characters, or any combination of lowercase and uppercase characters that does not exceed 480 bits. The title field can be null ("") or omitted. If omitted, the default title used in the screen mode parameter display is the procedure name pname.

The system displays the title whenever the procedure is called and required parameters are not supplied.

Parameter

Description

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

keywrđ"description"=(checklist)
or
keywrđ`description`=(checklist)

keywrđ Specifies the keyword of the parameter. keywrđ can be a string of 1 to 10 alphanumeric characters. The occurrences of keywrđ 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.†

If keywrđ is surrounded by dollar signs (\$keywrđ\$), any dollar symbols that surround or are contained within the parameter value are retained as a part of the value. Note that when the user is prompted for the value in screen mode, the value entered is treated as a literal value even if it is not surrounded by dollar symbols. If the specified value is not surrounded by dollar signs, the system converts it to a literal format (by appending prefixing and suffixing dollar signs and replicating any dollar signs that occur within the value) before substituting the value in the body of the procedure.

description Specifies an optional text string that the system displays when prompting for a parameter. The title string can be a maximum of 40 lowercase (12-bit) characters, 80 uppercase (6-bit) characters, or any combination of lowercase and uppercase characters that does not exceed 480 bits. In screen mode, the system displays a maximum of 40 characters. The description field may be null or omitted.

You must enclose the text string in quotation marks or apostrophes. If you use quotation marks, the default prompt for the parameter p_i has the following format:

Enter keywrđ description?

If you use apostrophes, the system uses only your description string in the prompt:

description?

Two examples of these prompting formats appear under the description of the .HELP directive.

†There is a seven-character restriction on the length of the keyword if the name call format of the BEGIN command is to be used, the keyword is to be specified on the call, or the procedure is a local file.

Parameter

Description

checklist . Specifies a list of the acceptable values and syntax for p_i . The checklist must be enclosed in 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 more than one match occurs, the first match determines the substitution.

If the checklist is omitted, the system assumes that any 0- to 40-character value is valid. The detailed description of acceptable checklist entries is in the following paragraphs.

ck Specifies a 1- to 10-character comment keyword. The keyword must be immediately preceded and followed by periods with no intervening spaces. This is an optional parameter. The system substitutes comments specified on the procedure call for the comment keyword in the procedure body. The system substitutes a null value for the comment keyword if no comments are on the procedure call. If you do not specify a comment keyword, the system ignores comments made on the procedure call.

Note that the system treats the comment keyword ck as a contiguous string upon substitution. If ck is concatenated to another long string in the procedure body, this substitution could produce a line that exceeds the line length limit (150 6-bit characters).

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 as described under Parameter Matching Modes later in this section.

CHECKLIST ENTRIES

Entries in the parameter checklist are of three types: simple values, equivalenced values, and checklist patterns. A single checklist can contain any combination of entry types and multiple entries of the same type.

A simple value is an entry of the form:

pstring

where pstring is any 0- to 40-character string. If pstring contains nonalphanumeric characters other than asterisks (*), it must be enclosed in literal delimiters (\$).

A simple value specifies that if the user enters a value of pstring on the BEGIN command, pstring is substituted for the parameter keyword in the procedure body.

An equivalenced value is an entry of the form:

pstring=pvalue

where pstring and pvalue are 0- to 40-character strings. If either pstring or pvalue contains nonalphanumeric characters other than asterisks (*), it must be enclosed in literal delimiters (\$).

An equivalenced value specifies that if the user enters a value of pstring on the BEGIN command, pvalue is substituted for the parameter keyword in the procedure body.

There are several patterns that can be used in the checklist. Each pattern defines a set of restrictions to be placed on values entered by the user. The checklist patterns are summarized in figure 4-4 and are described in detail on the following pages.

Most checklist pattern have three forms:

*p
*p=
*p=pvalue

where p is a character identifying the checklist pattern and pvalue is any 0- to 40-character string. If pvalue contains nonalphanumeric characters other than asterisks (*), it must be enclosed in literal delimiters (\$).

The *p form specifies that a parameter value entered by the user is substituted for the parameter keyword if the value meets the requirements defined by *p.

The *p= form specifies that a null substitution occurs if the value entered by the user meets *p requirements. In other words, all occurrences of the parameter keyword are omitted from the procedure command record.

The *p=pvalue form specifies that pvalue is substituted for the parameter keyword if the value entered by the user meets the requirements of the *p pattern.

Special Values

There are three special values that can be specified in an equivalenced value entry in a *p=pvalue checklist pattern. These special values are:

<u>Special Value</u>	<u>Significance</u>
#DATA	Specifies the name of the local file created by an unlabeled .DATA directive (that is, a .DATA directive on which the lfn parameter is omitted). Note that there can be no more than one unlabeled .DATA directive in a procedure. The actual file name used is ZZCCLAA, ZZCCLAB, ZZCCLAC, or so on, depending on the nesting level of the procedure.
#FILE	Specifies the file containing this procedure.
#PRIMARY	Specifies the primary file.

<u>Checklist Pattern</u>	<u>Description</u>
*Am..n†	Allows entry of any character string up to 40 characters. The string replaces keyword.
*Am..n=†	Entry of character string causes a null substitution.
*Am..n=value†	Entry of character string replaces keyword with value.
*Fm..n†	Allows entry of valid NOS file name. File name replaces keyword.
*Fm..n=†	Entry of a file name causes a null substitution.
*Fm..n=value†	Entry of a file name replaces keyword with value.
*K	Allows keyword to be entered as a value. No substitution results.
*K=	Entry of the keyword causes a null substitution.
*K=value	Entry of the keyword replaces keyword with value.
*N or *D	Specifies that parameter is optional. No substitution results.
*N= or *D=	Specifies that omission of parameter causes null substitution.
*N=value or *D=value	Specifies a default value for the parameter. Omission of parameter replaces keyword with value.
*Pm..n†	Allows entry of a valid NOS file name that does not begin with a digit. The file name replaces keyword.
*Pm..n=†	Entry of a file name causes a null substitution.
*Pm..n=value†	Entry of a file name replaces keyword with value.
*R	Specifies a restricted parameter whose values are not displayed in the dayfile. Substitution is not affected.
*Sm..n(chars)† or *Sm..n/k†	Allows entry of any characters selected from chars or from the set represented by k. These characters replace keyword. chars defines all allowable alphanumeric or special characters. k can be any of the following mnemonic abbreviations: A Any alphabetic characters B Any octal characters D Any decimal characters AB Any alphabetic and octal characters AD Any alphabetic and decimal characters
	To include an asterisk (*) as a member of the set, append an asterisk to the mnemonic. For example, D* specifies any decimal character and * as allowable characters.
*Sm..n(chars)=† or *Sm..n/k=†	Entry of allowable characters causes a null substitution.
*Sm..n(chars)=value† or *Sm..n/k=value†	Entry of allowable characters replaces keyword with value.
pstring	Entry of string pstring replaces keyword with pstring.
pstring=	Entry of string pstring causes a null substitution.
pstring=pvalue	Entry of string pstring causes pvalue to replace keyword.
†The specification m..n is optional and defines the minimum and maximum length of the character string or file name that can be entered for the parameter value. Valid ranges are 0 to 40 for *A, 1 to 7 for *F and *P, and 1 to 40 for *S. If only a single length value is specified, it is assumed to be the maximum. For example, *A10 allows entry of any string from 0 to 10 characters long.	

Figure 4-4. Summary of Checklist Entries

Checklist Patterns

<u>Checklist Pattern</u>	<u>Description</u>
*Am..n=value *Am..n= *Am..n	<p>Specifies the substitution for keywrd regardless of the specifications for p_i on the procedure call. $m..n$ specifies the minimum and maximum values for the length of the entry on the procedure call. The default value for m is 0 and for n is 40. The maximum value for n is 40. If $*Am..n=value$ is in the checklist, value replaces keywrd in the procedure body regardless of what is specified on the procedure call.</p> <p>You can specify any string of 0 to 40 uppercase characters for value. You must delimit special characters, other than asterisks (*), with dollar signs.</p> <p>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 the procedure resides on a library and is called by a name call form of the BEGIN command, keywrd is replaced by a null value. If value is #PRIMARY, all occurrences of keywrd in the procedure body are replaced by the current primary file.</p> <p>If $*Am..n=$ is in the checklist, a null value replaces keywrd. If $*Am..n$ is specified, whatever is specified on the procedure call replaces keywrd.</p> <p>If you specify some form of the *A entry more than once in a single checklist, the system uses the leftmost entry.</p> <p>Example:</p> <pre>.PROC,EBOC*I,P1>(*A5),P2>(*A10..15),P3>(*A20..).</pre> <p>In this procedure header directive, P1 allows entry of a character string 0 to 5 characters long, P2 allows a string 10 to 15 characters long, and P3 allows a string 20 to 40 characters long.</p>
*Fm..n=value *Fm..n= *Fm..n	<p>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. $m..n$ specifies the minimum and maximum length of the file name. The default value for m is 1 and for n is 7. If $*Fm..n=value$ is in the checklist, value replaces keywrd in the procedure body when a file name is specified on the procedure call. You can specify any 0- to 40-character string for value. You must delimit special characters, other than asterisks (*), with dollar signs.</p>

Checklist Pattern

Description

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 the procedure resides on a library and is called by a name call form of the BEGIN command, keywrđ is replaced by a null value. If value is #PRIMARY, all occurrences of keywrđ in the procedure body are replaced by the current primary file.

If *Fm..n= is in the checklist, a null value replaces keywrđ. If *Fm..n is specified, the file name specified on the procedure call replaces keywrđ.

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

Example:

Procedure EXEC is on local file EXEC:

```
.PROC,EXEC*I,I=(*F4..7),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.
```

*K=value
*K=
*K

Specifies the substitution for keywrđ when the parameter entry for p_i on the procedure call is only the keyword keywrđ or if the keyword is entered in response to an interactive prompt. If *K=value is in the checklist, value replaces keywrđ in the procedure body. You can specify any string of 0 to 40 uppercase characters for value. You must delimit special characters, other than asterisks (*), with dollar signs.

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 the procedure resides on a library and is called by a name call form of the BEGIN command, keyword is replaced by a null value. If value is #PRIMARY, all occurrences of keywrđ in the procedure body are replaced by the current primary file.

Checklist Pattern

Description

If *K= is in the checklist, a null value replaces keywr. 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. However, you can specify another pattern that is equivalent. Refer to procedures SUBN1 and SUBN2 in the following examples.

Examples:

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,P2,P3.
```

The procedure body becomes:

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

The following calls to procedures SUBN1 and SUBN2 illustrate the left-to-right checking of the checklist and the use of equivalent patterns. Procedures SUBN1 and SUBN2 are on local files by those same names and have the following lines, respectively:

```
.PROC,SUBN1*I,P1=(*K=KEYWORD,P1=STRING,*N=DEFAULT).  
NOTE.+#P1=P1
```

```
.PROC,SUBN2*I,P1=(*N=DEFAULT,P1=STRING,*K=KEYWORD).  
NOTE.+#P1=P1
```

Procedure SUBN1 is called:

```
subn1,p1
```

The system writes:

```
P1=KEYWORD
```

Procedure SUBN2 is called:

```
subn2,p1
```

The system writes:

```
P1=STRING
```

Checklist PatternDescription

A *K checklist pattern applies only when the keyword is entered by itself, or in response to an interactive prompt. It does not apply when an equivalenced value matching the keyword is entered as shown in the following call to procedure SUBN1:

```
subn1,P1=p1
```

The system writes:

```
P1=STRING
```

*N=value
*N=
*N
*D=value
*D=
*D

Specifies the substitution for keywrđ when there is no parameter entry for p_i on the procedure call. If *N=value or *D=value is in the checklist, value replaces keywrđ in the procedure body. You can specify any string of 0 to 40 uppercase characters for value. You must delimit special characters, other than asterisks (*), with dollar signs.

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 the procedure resides on a library and is called by a name call form of the BEGIN command, keywrđ is replaced by a null value. If value is #PRIMARY, all occurrences of keywrđ in the procedure body are replaced by the current primary file.

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

If no format of the *N or *D entry is in the checklist, p_i is a required parameter and interactive prompting occurs when it is omitted from the procedure call. However, if the only entry in the checklist is a form of *N or *D, any value supplied for it on the procedure call is ignored and interactive prompting does not occur. Specifying some form of the *N or *D entry more than once in a single checklist is an error.

Example:

Procedure SUB is on local file PROCFIL.

```
.PROC,SUB*I,P1=(*) ,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
```

Checklist PatternDescription

*N and *D values are displayed under different conditions. *D values are always displayed when screen mode interactive prompting is required and a non-null value has not been entered for the parameter. *D values are always accepted if entered. *N values are displayed in screen mode only when they are valid according to another checklist pattern which has no replacement value. If an *N value matches more than one checklist pattern, the first checklist pattern determines whether the value is displayed.

*Pm..n=value
*Pm..n=
*Pm..n

Specifies that the parameter entry for p_i on the procedure call may be a file name that does not begin with a digit and that conforms to the operating system format for a local file name. $m..n$ specifies the minimum and maximum length of the local file name. The default value for m is 1 and for n is 7. If *Pm..n=value is in the checklist, value replaces keyword in the procedure body when a file name is specified on the procedure call. You can specify any 0- to 40-character string for value. You must delimit special characters, other than asterisks (*), with dollar signs.

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 the procedure resides on a library and is called by a name call form of the BEGIN command, keyword is replaced by a null value. If value is #PRIMARY, all occurrences of keyword in the procedure body are replaced by the current primary file.

If *Pm..n= is in the checklist, a null value replaces keyword. If *Pm..n is specified, the file name specified on the procedure call replaces keyword.

If you specify some form of the *P entry more than once in a single checklist, the system uses the leftmost entry.

Example:

Procedure EXEC is on local file EXEC:

```
.PROC,EXEC*I,I=(*P4..7),B>(*N=LGO,*P),L>(*P=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.

*R

Specifies that the parameter entry for p_i on the procedure call is restricted and is not to be displayed in the job or system dayfiles. When *R is part of the checklist, any value supplied for p_i is removed from the command before it is placed in the dayfile.

*Sm..n(chars)=value Specifies that the parameter entry for p_i on the procedure call may consist of a string of characters selected (in any order) from the set of characters represented by chars or from the set of characters represented by k. m..n specifies the minimum and maximum values for the length of the string entered on the procedure call. The default for m is 1 and for n is 40. The maximum value for n is 40. chars represents a set of up to 40 uppercase alphanumeric and/or special characters. Special characters other than asterisks must be \$-delimited. k selects one of the following sets:

*Sm..n(chars)=
*Sm..n(chars)=
*Sm..n(chars)
or
*Sm..n/k=value
*Sm..n/k=
*Sm..n/k

<u>k</u>	<u>Characters</u>
A	All alphabetic characters
B	All octal characters
D	All decimal characters
AB	All alphabetic and octal characters
AD	All alphabetic and decimal characters

To include one or more asterisks as members of the set, append an asterisk to the mnemonic. For example, AB* specifies all alphabetic and octal characters and asterisks are allowed.

If *Sm..n(chars)=value or *Sm..n/k=value is in the checklist, value replaces keywrd in the procedure body when 1 to n characters from chars are specified on the procedure call. You can specify any string of 0 to 40 uppercase characters for value. You must delimit special characters, other than asterisks (*), with dollar signs.

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 the procedure resides on a library and is called by a name call form of the BEGIN command, keywrd is replaced by a null value. If value is #PRIMARY, all occurrences of keywrd in the procedure body are replaced by the current primary file.

Checklist Pattern

Description

If *Sm..n(chars)= or *Sm..n/k= is in the checklist, a null value is substituted for keyword. If *Sm..n(chars) or *Sm..n/k is in the checklist, the parameter entry for p_i on the procedure call replaces keyword.

Example:

Procedure COPIL is on local file COPI:

```
.PROC,COPIL*I,0"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,0,R,N,L,F.  
REVERT.
```

COPIL is called:

```
BEGIN,COPIL,COPI,0=OLD,R=MODIFID,  
N=NEW,F=AE.
```

The procedure body becomes:

```
COPYL,OLD,MODIFID,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/D).
```

Parameter entries could include P=BB, P=BCA, or P=901 but not P=XZY, P=AZ, P=P or P=A2.

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

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

Checklist Pattern

Description

pstring=pvalue
pstring=
pstring

Specifies the substitution for keywrd when the parameter entry for p_i on the procedure call matches pstring. pstring must be 0 to 40 uppercase characters. Special characters must be \$-delimited. If pstring=pvalue is in the checklist, pvalue replaces keywrd in the procedure body when pstring is specified on the procedure call. You can specify any string of 0 to 40 uppercase characters for pvalue. You must delimit special characters, other than asterisks (*) with dollar signs.

If pvalue 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 pvalue 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 contains the procedure. If the procedure resides on a library and is called by a name call form of the BEGIN command, keywrd is replaced by a null value. If pvalue is #PRIMARY, all occurrences of keywrd in the procedure body are replaced by the current primary file.

If pstring= is in the checklist, a null value replaces keywrd. If pstring is in the checklist, pstring replaces keywrd.

Example:

Procedure LABL on a global library file:

```
.PROC,LABL*I, FN"FILE NAME"=(*F),  
VSN "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.
```

Menu Format

This format of the .PROC directive generates a menu when you call the procedure and make no menu selection, when you make an incorrect selection, or when you request help information (refer to Requesting Help on Procedure Calls). The presence of the NOLIST parameter on the .HELP directive suppresses the menu display when you request .HELP information.

Format:

`.PROC,pname*M"title",keywr=(selections).ck.`

<u>Parameter</u>	<u>Description</u>
<code>pname</code>	Specifies the procedure name. <code>pname</code> must be a 1- to 7-character string of alphanumeric characters. Make the first character of <code>pname</code> an alphabetic character to avoid inadvertent editing of the primary file on name calls to procedures. You must append <code>*M</code> to <code>pname</code> to enable menu generation.
<code>title</code>	Specifies the menu title. The title string can be a maximum of 40 lowercase (12-bit) characters, 80 uppercase (6-bit) characters, or any combination of lowercase and uppercase characters that does not exceed 480 bits. The title field can be null (""), or completely omitted. If you omit <code>title</code> , the system uses the procedure name <code>pname</code> as the title when the procedure is called in screen mode. The title appears whenever the system displays the menu.
<code>keywr</code>	Specifies the keyword in the procedure body for which the system substitutes one of the selections. <code>keywr</code> can be a string of 1 to 10 alphanumeric characters.
<code>selections</code>	Defines the menu selections. A maximum of 50 selections can be defined for a menu. The definitions must have the following format: <code>num₁"descript₁",...,num_n"descript_n"</code> <code>num_i</code> Defines an integer number that identifies the menu selection in the screen display. <code>num_i</code> can be any integer up to 10 digits in length. Menu selections appear in the screen display in the order they are defined in the <code>.PROC</code> directive, not necessarily in the order defined by the selection numbers. <code>descript_i</code> Describes the menu selection. The description can be a maximum of 40 lowercase (12-bit) characters, 80 uppercase (6-bit) characters, or any combination of lowercase and uppercase characters that does not exceed 480 bits. In screen mode, the system will display a maximum of 40 characters. You can also specify a null description (""), or omit this field. This description appears whenever the system displays the menu.
<code>ck</code>	Specifies a 1- to 10-character comment keyword. This is an optional parameter. A period must follow the comment keyword. The system substitutes comments specified on the procedure call for the comment keyword in the procedure body. The system substitutes a null value for the comment keyword if no comments are on the procedure call. If you do not specify a comment keyword, the system ignores comments made on the procedure call.

Note that the system treats the comment keyword `ck` as a contiguous string upon substitution. If `ck` is concatenated to another long string in the procedure body, this substitution may produce a line that exceeds the line length limit (150 6-bit characters).

A menu has the following format:

```
title  
  
num1. descript1  
num2. descript2  
.  
.  
numn. descriptn
```

SELECT BY NUMBER OR TYPE Q TO QUIT?

where title, num_i, and descript_i conform to the descriptions given them previously. You can specify the prompt for a selection using the .PROMPT directive. If you do not, the system uses a default prompt as just shown. Additionally, the system centers all lines when in screen mode.

The Q option aborts the procedure call and initiates a search for an EXIT command in the procedure. If the procedure was called from another procedure, the EXIT commands in the calling procedure or procedures are also processed.

FORMATTING AND HELP DIRECTIVES

Along with the procedure header, the formatting and help directives further define the parameter prompts and help information that appears at the terminal. The formatting directives, if present, must follow the procedure header. Comment directives (beginning with `.*`) may be interspersed among the formatting directives. Help directives and help text follow the formatting directives. Comment directives placed in the help section are considered a part of the help text.

.CORRECT Directive

The `.CORRECT` directive specifies the first field of the prompt that follows a set of incorrect parameter entries. When the system prompts the procedure caller for corrections to an incorrect entry, the prompt has the following three fields:

```
text    parameter    description†
```

This directive specifies the text field.

Format:

```
.CORRECT,text.  
.CORRECT=text.
```

<u>Parameter</u>	<u>Description</u>
text	<p>Specifies a 1- to 80-character text string that appears on the screen. The system interprets your string in 6/12-bit display code, which supports uppercase and lowercase characters.</p> <p>In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (<code>.</code> or <code>)</code>). Special characters need not be enclosed in <code>\$</code> signs.</p> <p>In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in <code>\$</code> signs.</p> <p>In screen mode, the default text string is:</p> <pre> Please correct</pre> <p>In line mode, the default string is:</p> <pre> Correct</pre>

In screen mode, if the prompt is too long, the system truncates the prompt and indicates the truncation with an ellipsis (no truncation occurs in line mode). You can suppress display of the parameter field of the prompt by using apostrophes rather than quotation marks around the description in the procedure header. In screen mode, the prompt appears left-justified on the top line of the screen.

This directive does not apply to menu procedures.

† The description field does not appear in screen mode.

.ENDHELP Directive

The .ENDHELP directive specifies the end of the help section of the procedure body. You must include an .ENDHELP directive if there are any help directives in the procedure. Only one .ENDHELP directive is allowed in a procedure.

Format:

```
.ENDHELP.
```

.ENTER Directive

The .ENTER directive specifies the text for the initial prompt for a parameter. When the system initially prompts the user for a parameter, the prompt has the following three fields:

```
text      parameter  description
```

This directive specifies the text field.

Format:

```
.ENTER,text.  
.ENTER=text.
```

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

text	Specifies a 1- to 80-character text string that appears on the screen. In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (. or)). Special characters need not be enclosed in \$ signs.
------	---

In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in \$ signs.

The system interprets your string in 6/12-bit display code, which supports uppercase and lowercase characters. The default for the screen mode is:

```
      Please enter
```

The default for line mode is:

```
      Enter
```

In screen mode, if the prompt is too long, the system truncates the prompt and indicates the truncation with an ellipsis (no truncation occurs in line mode). You can suppress display of the text and parameter fields of the prompt by using apostrophes around the description in the procedure header. In screen mode, the prompt appears left-justified on the top line of the screen.

This directive does not apply to menu procedures.

.Fn Directive

The .Fn directive specifies a label for one of the six programmable function keys (F1 through F6) you can use with screen mode parameter displays.

Format:

.Fn,text.
.Fn=text.

where n is one of the integers 1 through 6.

<u>Parameter</u>	<u>Description</u>
text	Specifies a label string of not more than eight characters. You can specify lowercase and uppercase characters. In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (. or). Special characters need not be enclosed in dollar signs (\$). In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in dollar signs.

The function and default label of each of the programmable function keys are given in table 4-2.

.F7 Directive

The .F7 directive enables you to use function key 7 (F7) instead of the NEXT or RETURN key to execute screen mode parameter displays and menu selections. When you specify this directive, the system displays the F7 key along with the other programmable keys. If you do not label the key, the system labels it EXECUTE.

Format:

.F7.
.F7,text.
.F7=text.

<u>Parameter</u>	<u>Description</u>
text	Specifies a label string of not more than eight characters. You can specify lowercase and uppercase characters. If you enter the first format and omit text, EXECUTE is displayed (refer to table 4-2). In the second format, any alphanumeric or special characters may be used except a period or right parenthesis (. or). Special characters need not be enclosed in dollar signs (\$). In the third format, any characters may be used including a period or right parenthesis, but special characters must be enclosed in dollar signs.

Table 4-2. Programmable Function Keys for Procedures

Key	Default Label	Function
F1	FWD	Pages forward through screens of parameters or menu selections.
F2	BKW	Pages backward through screens of parameters or menu selections.
F3	HELP FWD	Pages forward through screens of .HELP text.
F4	HELP BKW	Pages backward through screens of .HELP text.
F5	HELP	Displays .HELP text at the bottom of the screen.
F6	QUIT	Aborts the procedure call and initiates search for an EXIT command in the current procedure and (if nested) in the calling procedures.
F7	EXECUTE	Initiates execution of the procedure if all required values are supplied.

.HELP Directive

The .HELP directive indicates that the text that follows 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 a question mark 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 is not read. Refer to Requesting Help on Procedure Calls later in this section. If specified, .HELP directives must immediately precede the .ENDHELP directive.

Formats:

```
.HELP.
.HELP,,NOLIST.
.HELP,param.
.HELP,param,NOLIST.
```

<u>Parameter</u>	<u>Description</u>
NOLIST	Suppresses display of some .HELP text during help processing.
parm	Specifies one of the parameter keywords in an interactive procedure (*I) or specifies one of the menu selections (choice _i) in a menu procedure (*M).

The first and second formats indicate the text that follows the directive describes the procedure. For interactive procedures in line mode, the NOLIST parameter suppresses the display of a parameter list when the caller requests information about the procedure. For menu procedures, the NOLIST parameter suppresses the display of the menu in line mode.

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

For menu procedures, the third and fourth formats indicate that the text following the directive describes the specified menu selection. The NOLIST parameter suppresses the display of the menu after the display of the .HELP text.

The text information starts on the line following the .HELP directive. The text can span multiple lines and must end with either another .HELP directive or an .ENDHELP directive. Parameter substitution does not occur within the .HELP text.

The system assumes that your .HELP text is in 6/12-bit display code, which supports uppercase and lowercase characters. In batch jobs, the system displays .HELP text in uppercase only.

Interactive (*I) Help

When you request help information about an interactive procedure, 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 (.HELP,,NOLIST).
- 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 particular parameter of an interactive procedure, 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.

Menu (*M) Help

When you request help information about a menu procedure, the system displays the following:

- The text following the procedure's .HELP directive.
- The menu unless suppressed by the NOLIST parameter of the .HELP directive (.HELP,,NOLIST).
- A prompt for a menu selection.

When you request information about a particular menu selection, the system displays the following:

- The text following the .HELP directive for that menu selection.
- The menu unless suppressed by the NOLIST parameter of the .HELP directive.
- A prompt for a menu selection.

Line Mode Example:

Procedure PRINT (on file PRINT) verifies that the selected file is local before it routes the file to a printer.

NOTE

The .HELP text uses lowercase and uppercase characters.

```
.PROC,PRINT*I,F"FILE NAME"=(*F),
DC"DISPOSITION CODE"=(*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.
.IF,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,?
  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? ?
ALLOWABLE VALUE(S)
MUST BE A FILE NAME
The name of the permanent file to be routed.

ENTER F FILE NAME? datafil
ENTER DC DISPOSITION CODE? ?
ALLOWABLE VALUE(S)
LP
PR
LR
LS
LT
PARAMETER MAY BE OMITTED
The disposition code. DC accepts only the line printer options.

ENTER DC DISPOSITION CODE? pr
```

To illustrate the effect of enclosing the description string for a parameter in apostrophes instead of quotation marks, we use the same procedure as in the previous example and replace the quotation marks with apostrophes:

```
.PROC,PRINT*I,F'FILE NAME'>(*F),
DC'DISPOSITION CODE'>(*N=LP,LP,PR,LR,LS,LT).
```

Note the difference in the prompts.

```
/begin,print,print?
This procedure routes a permanent file to the selected line printer.
PARAMETERS FOR PRINT ARE F, DC
FILE NAME? ?
ALLOWABLE VALUE(S)
MUST BE A FILE NAME
The name of the permanent file to be routed.

FILE NAME? datafil
DISPOSITION CODE? ?
ALLOWABLE VALUE(S)
LP
PR
LR
LS
LT
PARAMETER MAY BE OMITTED
The disposition code. DC accepts only the line printer options.

DISPOSITION CODE? pr
```

Screen Mode Example:

If you are at a terminal supported by the full-screen display products and your terminal is in screen mode, the system can use full-screen displays to prompt you for parameters and to provide you with help information.

The following screen results if you call procedure PRINT (as defined in the previous example):

PRINT

F FILE NAME: _____
DC DISPOSITION CODE: LP

Specify values and press NEXT when ready

F5 HELP F6 QUIT

If you press the HELP key while the cursor is positioned in the input field for first parameter, you get the following screen:

PRINT

F FILE NAME: _____
DC DISPOSITION CODE: LP

Specify values and press NEXT when ready

— F FILE NAME —

ALLOWABLE VALUE(S)
MUST BE A FILE NAME
The name of the permanent file to be routed.

F5 HELP F6 QUIT

If you press the HELP key while the cursor is positioned in the input field for the second parameter, you get the following screen:

PRINT

F FILE NAME: _____
DC DISPOSITION CODE: LP

Specify values and press NEXT when ready
DC DISPOSITION CODE

ALLOWABLE VALUE(S)
PR
LR
LS
LT

PARAMETER MAY BE OMITTED
The disposition code. DC accepts only the line printer options.

F5 F6

To get help information about the procedure itself, you can append a question mark to the name of procedure on the call or press the HELP key again after you get help for any of the parameters.

PRINT

F FILE NAME: _____
DC DISPOSITION CODE: LP

Specify values and press NEXT when ready
PRINT

This procedure routes a permanent file to the selected line printer.

F5 F6

Refer to Volume 2 for a detailed description of calling procedures in screen mode.

.NOCLR Directive

The .NOCLR directive inhibits the system from automatically clearing the screen at the end of the procedure call (once all required parameters are supplied). You can also specify a message to appear at the top of the screen. Unless specified to do otherwise, the system clears the screen at the end of the call, and the screen is blank until the procedure completes execution or sends output to the terminal.

The .NOCLR directive must be placed before any .HELP directives in the procedure.

Format:

.NOCLR,message.
.NOCLR=message.

<u>Parameter</u>	<u>Description</u>
message	<p>Specifies a 1- to 40-character text string that appears at the top of the screen. You can use uppercase and lowercase characters in the string.</p> <p>In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (. or)). Special characters need not be enclosed in dollar signs (\$).</p> <p>In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in dollar signs.</p>

This directive applies only if the procedure is executed in screen mode.

.NOTE Directive

The .NOTE directive specifies the message that appears on the screen and in your dayfile at the end of the procedure call (once all required parameters are supplied). Unless specified to do otherwise, the system clears the screen at the end of the call (refer to the .NOCLR Directive), and the screen is blank until the procedure completes execution or sends output to the terminal.

Format:

.NOTE,message.
.NOTE=message.

<u>Parameter</u>	<u>Description</u>
message	<p>Specifies a 1- to 40-character text string that appears on the screen and in your dayfile. You can use uppercase and lowercase characters in the string.</p> <p>In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (. or)). Special characters need not be enclosed in dollar signs (\$).</p> <p>In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in dollar signs.</p>

This directive applies only if the procedure is executed in screen mode.

.PAGE Directive

The .PAGE directive specifies the string that precedes the page number for procedure displays that require more than one screen.

Format:

```
.PAGE,text.  
.PAGE=text.
```

<u>Parameter</u>	<u>Description</u>
text	<p>Specifies a 0- to 40-character text string. You can use uppercase and lowercase characters in the string. The default is:</p> <p style="text-align: center;">Page</p> <p>In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (. or)). Special characters need not be enclosed in dollar signs (\$).</p> <p>In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in dollar signs.</p>

.PROMPT Directive

When a procedure is called in screen mode, the .PROMPT directive defines the last line of the screen, assuming no .HELP text is being shown. By default, this line tells the procedure caller how to proceed. For menu procedures, the input field immediately follows this prompt. In line mode, this directive applies only to menu procedures, where the system produces a screen-like display.

Format:

```
.PROMPT,text.  
.PROMPT=text.
```

<u>Parameter</u>	<u>Description</u>
text	<p>Specifies a 0- to 40-character text string that appears as the prompt on the screen. The text string can be a maximum of 40 lowercase (12-bit) characters, 80 uppercase (6-bit) characters, or any combination of lowercase and uppercase characters that does not exceed 480 bits.</p> <p>In the first format, any alphanumeric or special characters may be used except a period or right parenthesis (. or)). Special characters need not be enclosed in dollar signs (\$).</p> <p>In the second format, any characters may be used (including a period or right parenthesis), but special characters must be enclosed in dollar signs.</p>

Parameter

Description

If the text parameter is omitted, the system issues the following prompts:

Mode and
Procedure Format

Prompt

Screen mode

Interactive
procedure

Specify values and press NEXT when ready.

Menu procedure

Select from the list above and press NEXT.

Line mode

Interactive
procedure

(Not applicable)

Menu procedure

SELECT BY NUMBER OR TYPE Q TO QUIT.

EXPANSION CONTROL DIRECTIVES

The expansion control directives allow you to control certain aspects of procedure expansion. These directives allow you to inhibit expansion of portions of a procedure, to change the inhibit and concatenation characters, and to skip portions of the procedure when writing lines to the procedure command record.

.CC Directive

The .CC directive specifies the character used to concatenate character strings during procedure expansion. The new concatenation character remains in effect until the procedure terminates or until you enter another .CC directive. The default concatenation character is the ASCII character _ or the CDC graphics character ▸. The .CC directive can be placed anywhere in the procedure.

Format:

.CC(n)

Parameter

Description

n

Specifies the new concatenation character.

.EXPAND Directive

The .EXPAND directive terminates or restores procedure expansion for subsequent lines of the procedure. The .EXPAND directive can be used anywhere in the procedure following the formatting and help directives.

Format:

EXPAND,option

<u>Parameter</u>	<u>Description</u>
ON	Specifies that subsequent lines of the procedure are expanded normally. This is the default value.
OFF	Specifies that no parameter substitutions or directive processing is performed on subsequent lines. In other words, any subsequent commands or procedure directives are written to the command record just as they appear in the procedure file.

.IC Directive

The .IC directive specifies the character used to inhibit parameter substitution or directive processing during procedure expansion. The new inhibit character remains in effect until the procedure terminates or until you enter another .IC directive. The default inhibit character is the ASCII character # or the CDC graphics character ≡. The .IC directive can be placed anywhere in the procedure.

Format:

.IC(n)

<u>Parameter</u>	<u>Description</u>
n	Specifies the new inhibit character.

.SET Directive

The .SET directive allows you to build new parameters using strings and substrings created by the STR, STRB, and STRD functions and concatenation operations, which are described in section 6.

Format:

.SET,keyword₁=strex₁,...,keyword_n=strex_n.

<u>Parameter</u>	<u>Description</u>
keyword _i	<p>Defines a new keyword or references an existing keyword, either from the .PROC header or from an earlier .SET directive. keyword_i can be substituted in subsequent statements as if it were a formal parameter in the .PROC header.</p> <p>If keyword_i is an existing keyword, it is prefixed with an inhibit character to avoid unwanted substitution of the existing value.</p> <p>If keyword_i is enclosed in dollar signs, values substituted for keyword_i in subsequent statements remain in or are converted to the literal format (by appending prefixing and suffixing dollar signs and replicating any dollar signs that occur within the values).</p>
strex _i	<p>May be any valid CCL expression, usually but not necessarily an expression that produces a string result. If the result of strex_i is not a string but a value, the value is left-justified and considered a string. The resulting string in either case is substituted for any occurrences of keyword_i in subsequent statements of the procedure.</p>

Example 1:

```
.PROC,CATTER*I,P1=(*A),P2=(*S3/D).
.SET,K9=STR($P1$,1,4)//STRD(P2,-1).
.IF,$K9$.EQ.$GOOD1$,GO.
NOTE./THE 1ST 4 CHAR OF P1 AND THE LAST OF P2 = K9.
.ELSE,GO.
REVERT,EX.DISPLAY,STR($SORRY, K9 IS NOT GOOD1$).
.ENDIF,GO.
```

In this example, two substrings were concatenated to produce one string in the .SET expression. Every occurrence of keyword K9 in the lines following the .SET directive is replaced by the string resulting from the concatenation of substrings P1 and P2. If GOODY and 101 were supplied as values for P1 and P2, the value of K9 would be set to GOOD1. (For a detailed explanation of how the STR and STRD functions operate, refer to section 6.) The subsequent .IF statement would be true, causing the NOTE command to be included and executed. If BADNEWS were substituted for P1, the .IF statement would be false, resulting in the execution of the REVERT command and then the DISPLAY command.

Example 2:

```
.PROC,VALUES*I,P1=(*S/D),P2=(*S/D).
.SET,V1=STRD(P1+P2),V2=STRD(P1*P2),V3=STRD(P1**P2).
.IF,V1=V2=V3.NOTE./SUM, PRODUCT AND POWER ARE EQUAL.
NOTE./THE SUM OF P1 AND P2 IS V1.
NOTE./THE PRODUCT OF P1 AND P2 IS V2.
NOTE./P1 TO THE POWER OF P2 IS V3.
```

As this example demonstrates, the .SET directive along with the STRD function can create new keywords that make it possible to include arithmetic values as strings in subsequent commands and directives.

Example 3:

```
.PROC,LITKW*I,P1=(*A),$P2$=(*A).
NOTE./ #P1(P1) #P2(P2) #P3(P3) #P4(P4)
.SET,P3=$GHI$, $P4$=$JKL$.
NOTE./ #P1(P1) #P2(P2) #P3(P3) #P4(P4)
.SET,$#P1$=$P1$,#P2=$P2$,#P3$=$P3$,#P4=$P4.
NOTE./ #P1(P1) #P2(P2) #P3(P3) #P4(P4)
REVERT. LITKW COMPLETED.
```

Execution of procedure LITKW using a BEGIN command produces:

```
07.16.51.BEGIN,,LITKW,ABC,DEF.
07.16.52.NOTE./ P1(ABC) P2($DEF$) P3(P3) P4(P4)
07.16.52.NOTE./ P1(ABC) P2($DEF$) P3(GHI) P4($JKL$)
07.16.52.NOTE./ P1($ABC$) P2(DEF) P3($GHI$) P4(JKL)
07.16.52.REVERT. LITKW COMPLETED.
```

This example illustrates the creation of keywords not previously defined, redefinition of existing keywords, and changing the literal keyword characteristics of existing keywords.

FILE DIRECTIVES

The file directives either create local files during procedure expansion time or place file marks in the procedure command record.

.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. The data file can contain program source code, other NOS procedures, or any other type of data.

Format:

```
.DATA, lfn.
```

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the 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. You cannot specify INPUT for lfn. If lfn is not specified, the .DATA file can be referenced in the procedure header using the special symbol #DATA. The default for lfn depends on the nesting level of the procedure. At the first procedure level, the system calls this file ZZCCLAA; at the second procedure level, it is ZZCCLAB; and so on.

You can use the .IF, .ELSE, and .ENDIF directives within the data lines following the .DATA directive to conditionally include lines in the data file.

File marks generated by .EOR and .EOF directives 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 unless you insert a .EXPAND directive to suspend expansion processing. After the data file is written, it is rewound.

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),  
LSTFILE>(*F,*N=FTNOUT).  
FTN5,I=P1,L=LSTFILE.  
LGO.  
REPLACE,LSTFILE.  
.DATA  
PROGRAM APROG  
.  
.  
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.  
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 is performed on the FORTRAN program prior to its being written to 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>(*F,*N=#FILE),  
LSTFILE>(*F,*N=FTNOUT).  
FTN5,I=P1,L=LSTFILE.  
LGO.  
REPLACE,LSTFILE.  
-EOR-  
PROGRAM BPROG  
.  
.  
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  
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.  
FTN5,I=P1,L=X.  
LGO.  
REPLACE,X=LISTFIL.
```

The following call accesses procedure file GAMMA.

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

Parameter substitution occurs 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.FTN5,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
```


.EOF (or .EOP) Directive

The .EOF (or .EOP) directive generates an end-of-file on the data file created with a .DATA directive. Whenever 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. The .EOP directive may be used interchangeably with the .EOF directive.

Format:

.EOF.
.EOP.

.EOR (or .EOS) Directive

The .EOR (or .EOS) directive is used to separate records in a data file created with a .DATA directive. Whenever 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. The .EOS directive may be used interchangeably with the .EOR directive.

Format:

.EOR.
.EOS.

BRANCHING DIRECTIVES

The branching directives provide conditional inclusion of commands, other directives, or data lines in the procedure. The function of the .ELSE, .ENDIF, and .IF directives is much like that of their command equivalents. The branching can appear in any of the four sections of the procedure body.

.ELSE Directive

The .ELSE directive in conjunction with the .IF directive either terminates or initiates the skipping of lines in the procedure body.

Format:

.ELSE,label.

<u>Parameter</u>	<u>Description</u>
label	Specifies a 0- to 10-character alphanumeric string. The string must begin with an alphabetic character unless it is a null string.

If the condition in the matching .IF directive is true, the .ELSE directive initiates the skipping of the lines between itself and the next matching .ENDIF directive (two directives match only if their label parameters are identical). If the condition in the matching .IF directive is false, the .ELSE directive terminates the skipping that the .IF directive initiated. An .ELSE directive cannot terminate the skipping initiated by another .ELSE directive.

.ENDIF Directive

The .ENDIF directive terminates skipping initiated by a matching .IF or .ELSE directive (directives match only if their label parameters are identical).

Format:

```
.ENDIF,label.
```

<u>Parameter</u>	<u>Description</u>
label	Specifies a 0- to 10-character alphanumeric string. The string must begin with an alphabetic character unless it is a null string.

.EX Directive

The .EX directive submits a single command to the system for immediate execution. The system performs keyword substitutions in the command before executing the command. No further procedure directives are normally processed. Although .DATA files are created, the command record is never executed. Since the command record is not executed, the procedure nesting level of your job is not incremented.

Format:

```
.EX.command.
```

<u>Parameter</u>	<u>Description</u>
command	Any valid command. The length of the expanded command cannot exceed 80 characters.

If the command entered by .EX aborts, the system responds as though the command had been entered from the calling job or procedure. For example, assume that procedure A calls procedure B and that procedure B contains a .EX directive. If the command entered by the .EX directive is erroneous, the system immediately suspends command processing and begins searching for an EXIT command in procedure A.

The .EX directive is most often used in menu procedures in which the command executed by .EX is a call (BEGIN command) to another procedure. Since the purpose of the menu procedure is merely to prompt for a single item of user input, the most efficient way of calling the resulting procedure is in a .EX directive.

Example:

The following menu procedure initiates one of three nested procedures, (EXEC1, EXEC2, or EXEC3) depending on the menu item selected.

```
.PROC, FIX*M, P=  
(1"OPTION 1",  
 2"OPTION 2",  
 3"OPTION 3").  
.EX.BEGIN, EXEC_P, PFILE.  
REVERT, NOLIST.
```

.IF (or .IFE) Directive

The .IF directive conditionally writes lines from the procedure body to the procedure command record or the file associated with the .DATA directive. The .IF directive is to the procedure command record as the IF command is to the job command record (refer to the IF command). Before evaluating the conditional expression, the system performs parameter substitution for the portion of the directive that follows the .IF. Since the system processes the .IF directive at procedure expansion time, the .IF directive cannot depend upon the effects of commands in the procedure body.

Formats:

.IF,expression.command.
.IF,expression,label.

<u>Parameter</u>	<u>Description</u>
expression	Specifies an expression that is either true or false (refer to Command Syntax in section 6). The separator following expression in the first format must be a terminator.
command	Specifies any valid system command.
label	Specifies a 0- to 10-character alphanumeric string. The string must begin with an alphabetic character unless it is a null string. An identical string must appear in a subsequent .ELSE or .ENDIF directive.

If expression is true, the first format writes command in the procedure command record while the second format processes all lines between the .IF directive and the first matching .ELSE or .ENDIF directive (two directives match only if their label parameters are identical).

If expression is false, the system does not write command in the procedure command record (for the first format) nor does the system write and process the lines between the .IF directive and the first matching .ELSE or .ENDIF directive (for the second format).

NOTE

When using the first format (.IF,expression.command.), values for .command. that match a relational, logical, or string operator (such as .EQ. or .OR.) are interpreted as operators, causing the procedure to abort. To write a valid .IF directive when .command. matches an operator, terminate the expression or command with a right parenthesis instead of a period.

COMMENT (.) 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.

Format:

```
.*comment
```

<u>Parameter</u>	<u>Description</u>
comment	Represents your comments.

The comment directive can appear anywhere in the procedure body. However, a comment directive inserted in the help section is considered as part of the help text.

CALLING A PROCEDURE (BEGIN COMMAND)

The BEGIN command initiates execution of a procedure. When a BEGIN command is entered, the system begins executing the specified procedure and continues until a REVERT command is entered. The REVERT command causes the system to return to the command record of the calling job or procedure. The system then continues execution with the first command following the BEGIN command. Procedures can be nested up to a level of 50 nested procedures.

Figure 4-5 shows a diagram of a procedure call from a batch job and from an interactive job. The interactive example also diagrams a first level nested procedure call.

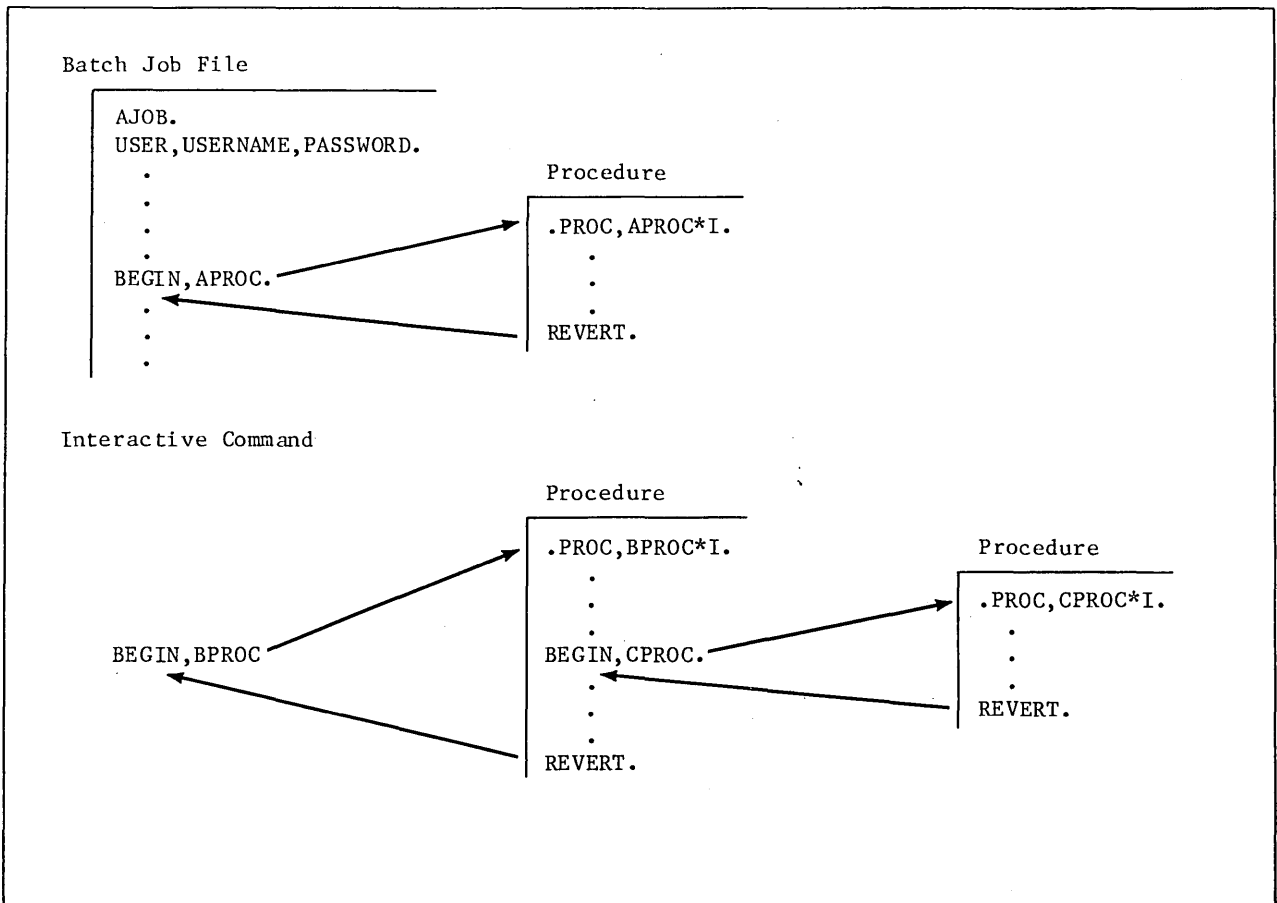


Figure 4-5. Calling a Procedure

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).

The system allows you to input the parameters from a full-screen format if the following conditions are true:

- Your terminal supports full-screen input for procedures.
- Your terminal is in screen mode (refer to the SCREEN command).
- The procedure is interactive; that is, the .PROC directive contains a *M or *I.
- You do not supply all the required parameters on the procedure call.

Formats:

BEGIN,pname,pfile,p1,p2,...,pn.comment

or

-pname,pfile,p1,p2,...,pn.comment

or

pname,p1,p2,...,pn.comment

or

pfile,p1,p2,...,pn.comment

The first format is the full BEGIN command format that can be used in either batch or interactive jobs. When you use this format, the system uses the following search order when attempting to locate file pfile:

- local files
- indirect access files in your permanent file catalog
- direct access files in your permanent file catalog
- indirect access files under user name LIBRARY
- direct access files under user name LIBRARY

The second format can be used only from an interactive job. It uses the same search order as the first format.

The third and fourth formats are the name call formats in which the command name BEGIN can be omitted.

The third format can be used from a batch or interactive job if one of the following is true:

- pname is a procedure within a record in a file called pname. The system locates this procedure record and executes it.
- pname is a procedure in a global library set.
- pname is a procedure in the system library.

The fourth format can be used from a batch or interactive job. pfile is the name of a local file. The system executes the first procedure on pfile.

<u>Parameter</u>	<u>Description</u>
pname	Specifies the procedure name from the procedure header directive. 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.
pfile	Specifies the name of the file containing the procedure. The file name pfile must be inserted as 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 Requesting Help on Procedure Calls later in this section). File pfile is not automatically returned after the procedure executes. If pfile is omitted from the first format, the installation-defined default file name is used (PROCFIL is the default).
pi	Specifies an optional parameter that may affect the substitution to be made for a keyword used in the procedure. The value you specify is compared to the appropriate checklist specified for this parameter in the procedure header. The checklist entries are checked in a left-to-right fashion and the first match determines the substitution. Refer to the .PROC Directive in this section for more details on parameter substitution. The specific formats for pi are described later in this subsection. If you are using the third or fourth format of the BEGIN command as previously shown, there is a 7-character restriction on both the keyword portion and the value portion of the parameter pi.
comment	Specifies the value associated with the comment keyword (ck) on the procedure header. The value consists of any characters you specify following the terminator. If the ck keyword appears on the procedure header, the system substitutes your comment for the comment keyword ck in the procedure body. Otherwise, the system ignores this comment.

For interactive procedures (*I), optional parameters p_i can have the following formats:

<u>P_i</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 Requesting Help on Procedure Calls later in this section).
keywrđ=	Removes keywrđ in the procedure body (null substitution) unless overridden by a checklist specification.
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. val can be a 1- to 40-character symbolic name or value. Section 6 describes symbolic names. val can contain special characters (nonalphanumeric) if they are delimited by dollar signs. Asterisks are the exception; they can appear without dollar sign delimiters.

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 keywrđ=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.

<u>P_i</u>	<u>Description</u>
val	Unless overridden by a checklist specification, assigns this 0- to 40-character (uppercase only) 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). Special characters (nonalphanumeric), other than asterisks (*), must be delimited by dollar signs. 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.

NOTE

If you specify a parameter p_i on a name call to a local file, neither the keyword keywrđ or the value val can exceed seven characters.

For menu procedures (*M), the procedure call can contain only one substitution parameter p_i. On such calls, p_i can have one of the following formats:

<u>P_i</u>	<u>Description</u>
choice	Specifies a menu selection that appears on the procedure header. The menu selection must be an unsigned integer.
choice? or ?	Specifies interactive help processing (refer to Requesting Help on Procedure Calls later in this section).

If you omit p_i, the system displays the menu and prompts for a menu selection.

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,FK.
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
10.15.35.BEGIN,TEST1,FKTEST,*.
10.15.35.COMMENT.    *
10.15.35.REVERT.CCL
```

REQUESTING HELP ON PROCEDURE CALLS

After you call a procedure interactively but before the system executes the specified procedure, you can have a dialogue with the system about the procedure. This is possible only if the procedure writer appended the *I or *M 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 information about the procedure itself.
- Request help information about a procedure parameter.
- Supply omitted parameters.
- Reenter parameter values that are in error.

If the parameter specifications in the procedure call are incorrect or incomplete, 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 for *I procedures or to a menu selection for *M procedures.

- Enter a question mark as a parameter on the procedure call.
- Enter a question mark in response to an interactive prompt.
- Press the HELP key any time after you call the procedure if your terminal is in screen mode.

If you omit required parameters or if 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:

- You satisfy all parameter requirements.
- You enter a parameter followed by a terminator (a period or a right parenthesis), or enter just a terminator. If all required parameters have been entered, the system executes the procedure. Otherwise, the system continues prompting until all required parameters are satisfied.
- It is a menu procedure executing in line mode and you enter Q. The system does not execute the menu. For nested menu procedures, entering Q causes the system to discontinue expansion of the current procedure and begin searching for an EXIT command in the calling procedure.
- Your terminal is in line mode and you enter the termination or interruption character for your terminal (refer to appendix J). The BEGIN command processing ends and the system does not execute the procedure. For nested procedures, terminating the dialogue in this way causes the system to search for an EXIT command in the calling procedure.
- Your terminal is in screen mode and you press the QUIT key. The system does not execute the procedure. For nested procedures, terminating the dialogue in this way causes the system to search for an EXIT command in the calling procedure.

As with all commands, you can always append a comment after the terminator on the call. Additionally, you can append a comment to a parameter entry made during a parameter-prompting dialogue. Still, the comment must follow a terminator (refer to the last example in this subsection).

The procedure calls in table 4-3 illustrate the various methods of requesting help on procedure calls. For convenience, we show the name call format of the BEGIN command.

Table 4-3. Help on Procedure Calls

Procedure Call	Result	
	Interactive Procedure (*I)	Menu Procedure (*M)
pname	The system prompts for required parameters or parameter values in error.	The system displays the menu and prompts for a selection.
pname?	The system provides any .HELP text about the procedure itself and then prompts for parameters.	The system provides any .HELP text about the procedure itself, displays the menu, and prompts for a selection.
pname,?	Same as pname? .	Same as pname,? .
pname,keyword?	The system provides any .HELP text for the parameter associated with keyword and then prompts for parameters.	Not applicable.
pname,choice?	Not applicable.	The system provides any .HELP text for that menu selection, displays the menu, and prompts for a selection.

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 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"OUT PUT"=(*F,*N=OUT PUT),
LO"LIST OPT IONS"=(*N=0,0,0,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,?

The following dialogue ensues:

```
PARAMETERS FOR F5 ARE I, B, L, LO
ENTER I INPUT? test
ENTER B BINARIES ? bfile
ENTER L OUT PUT ? listing
ENTER LO LIST OPT IONS? s
REVERT.
```

The following example is the same as the previous example, except that apostrophes replace the quotation marks in the procedure header:

```
.PROC,F5*I,I'INPUT'=(*F,*N=INPUT),
B'BINARIES'=(*F,*N=LGO),
L'OUT PUT'=(*F,*N=OUT PUT),
LO'LIST OPT IONS'=(*N=0,0,0,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,?

The following dialogue ensues:

```
PARAMETERS FOR F5 ARE I, B, L, LO
INPUT? test
BINARIES ? bfile
OUT PUT ? listing
LIST OPT IONS? s
REVERT.
```

The following example shows help processing with a menu procedure:

NOTE

The .HELP text uses lowercase and uppercase characters.

```
/list,f=menu
.proc,menu*m"sample menu",option=
(1"option 1",2"option 2",3"option 3").comment.
.help
This procedure shows the structure of a menu procedure.
.help,1
This is HELP text for option 1.
.help,2
This is HELP text for option 2.
.help,3
This is HELP text for option 3.
.endhelp.
.if,option.eq.1.note.+you chose #option 1 comment.
.if,option.eq.2.note.+you chose #option 2 comment.
.if,option.eq.3.note.+you chose #option 3 comment.
revert,nolist.
```

/menu?

This procedure shows the structure of a menu procedure.

SAMPLE MENU

1. OPTION 1
2. OPTION 2
3. OPTION 3

SELECT BY NUMBER OR TYPE Q TO QUIT ? 1?

This is HELP text for option 1.

SAMPLE MENU

1. OPTION 1
2. OPTION 2
3. OPTION 3

SELECT BY NUMBER OR TYPE Q TO QUIT ? 2?

This is HELP text for option 2.

SAMPLE MENU

1. OPTION 1
2. OPTION 2
3. OPTION 3

SELECT BY NUMBER OR TYPE Q TO QUIT ? 3?

This is HELP text for option 3.

SAMPLE MENU

- 1. OPTION 1
- 2. OPTION 2
- 3. OPTION 3

SELECT BY NUMBER OR TYPE Q TO QUIT ? 1. finally
 YOU CHOSE OPTION 1 FINALLY.

/menu,2
 YOU CHOSE OPTION 2.

ENDING A PROCEDURE (REVERT COMMAND)

The REVERT command terminates procedure processing.

Format:

REVERT,option.command

Parameter

Description

option

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

option

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 causes the REVERT command to appear at the terminal and in the job dayfile.

EX

Returns control to the command record of the calling procedure and executes the command parameter as if the command had actually appeared in the calling procedure. 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 option, 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.

command

Specifies a comment when used with the EX option. Otherwise, it specifies a comment.

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-supplied 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.
  .IF,FILE(LFN1,RD),LABEL1.
    TDUMP,I=LFN1.
  .ELSE,LABEL1.
    REVERT,NO READ PERMISSION
  .ENDIF,LABEL1.
  .IF,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 or execute file.

```
REVJOB1.                                REVJOB2.
USER,USERNAM,PASSWRD,FAMNAME.           USER,USERNAM,PASSWRD,FAMNAME.
ATTACH,FILE1/UN=ALTUSER,PW=PW1,M=E.     ATTACH,FILE2/UN=ALTUSER,PW=PW2,M=R.
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 by error processing.

```
17.03.53.REVJOB1.                        17.05.05.REVJOB2.
17.03.53.USER,USERNAM,,FAMNAME.          17.05.05.USER,USERNAM,,FAMNAME
17.03.53.CHARGE,*                         17.05.05.CHARGE,*
17.03.53.* CHARGE(1514,5612PAY)           17.05.05.* CHARGE(1514,5612PAY)
17.03.56.ATTACH,FILE1/UN=ALTUSER,M=E.     17.05.13.ATTACH,FILE2/UN=ALTUSER,M=R.
17.03.56.BEGIN,REVTST,PROCFL,FILE1,XFIL. 17.05.13.BEGIN,REVTST,PROCFL,FILE2,XFIL.
17.03.56.REVERT.NO READ PERMISSION        17.05.13.TDUMP,I=FILE2.
17.03.56.COMMENT. RETURNS HERE           17.05.13. TDUMP COMPLETE.
17.03.56.EXIT.                            17.05.13.REVERT,ABORT. NO READ/MODIFY
                                           PERMISSION
17.05.13.EXIT.                            17.05.13.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.
 - Flow 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 consist of from one to four fields. The first field is optional. It 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 one of your global library files, 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 first 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 first 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 an 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:

If file EXTRACT is a local file and you enter

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

the system treats the file as a local file command. The parameters are processed in operating system format because of the leading slant.

The second field contains the name of the program to be executed. The command name can be any valid file name.

The third 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 third field (or the second field if no parameters are present). In interactive jobs, the terminator is optional.

The fourth field consists of a comments field. The comments field follows the terminator. In general, the system ignores this field. However, some commands (MODIFY, FSE, GTR, LIBEDIT, and the like) read this comments field for directives.

In general, commands may not be continued beyond a single line. Exceptions are the execution control commands (described in section 6) and tape management commands (described in section 12).

Global library file commands containing an ARG= entry point (ARG= is described in Volume 4, Program Interface) are not required to follow either the operating system or product set format. The only syntax requirements enforced by the system for these commands is that the command name must be a valid file name followed by a separator. The command itself may have additional syntax requirements.

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).

Operating System Format

1. Valid separators are
+ - " / = , (

and any other character with a display code value greater than 44₈ except *) \$. and blank.
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 job'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

1. Same as for the operating system 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

Operating System Format

Product Set Format

<u>Character</u>	<u>Code (Octal)</u>
;	10
) or .	17
Other valid separators	16

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

7. File names are one to seven alphanumeric characters.

7. File names are one to seven alphanumeric characters. In some products, file names beginning with a numeric character are invalid.

8. Not NOS/BE compatible.

8. NOS/BE compatible.

In general, no parameter can contain more than seven characters. The exceptions include procedure calls, flow control commands, some tape management commands and system or global library file commands with an ARG= entry point.

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 1008 words of your job's field length, from RA through RA+778. Section 1 and appendix E in Volume 4, Program Interface, describe 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.
```

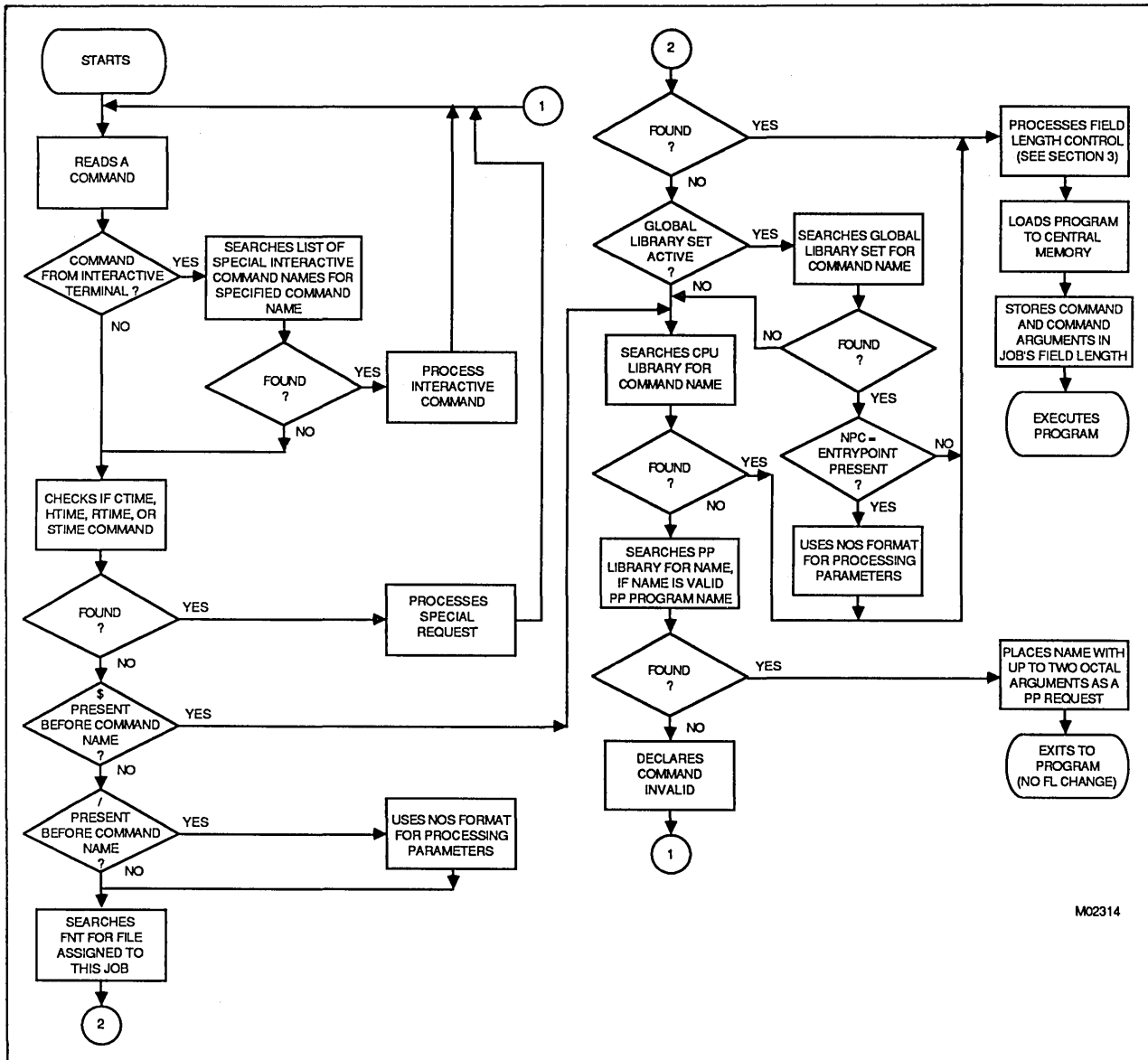
COMMAND PROCESSING FLOW

The system translates a command by the following steps (figure 5-1 shows the processing flow).

1. Reads the command from the job's command record or interactive terminal.
2. Verifies the format of the command as described in Command Format.
3. If the command is from an interactive terminal, the system compares the command with the list of special interactive command names. (See section 8 for a list of these commands.) If the command name is in this list, the command is processed by IAF.
4. Compares the specified command names with the list of command names. If the command name is CTIME, HTIME, RTIME, *, or STIME, the system processes the command.
5. Searches the job's file name table (FNT) 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) unless the command is a system or global library file command with an ARG= entry point. The CPU is requested to begin execution unless special loader commands follow.
6. Searches 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 5; otherwise, the system searches further.
7. Searches the central library directory (CPU library) 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.
8. If the command name is a three-character name with the first character alphabetic, and if there are no more than two parameters (each of which is an octal number), the system searches the peripheral processor library directory (PP library) for a program name that matches the command name. If found, the name is placed as a peripheral processor request.
9. If the system does not find the command name during any of the searches, it declares the command invalid and aborts the job.

NOTE

A command is normally echoed to the dayfile except for system commands or global library commands containing an SDM= entry point in the entry point list.



M02314

Figure 5-1. NOS Command Processing Flow

EXIT PROCESSING

When an error condition occurs during a job, 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 job is not interactive or returns control to the terminal if the job is interactive. 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.  
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 ONEXIT command, processing continues until reaching the EXIT command and then the job terminates (ENQUIRE command is not processed).

† After a security conflict, the system does not always allow EXIT processing. Refer to Security Features in section 3.

FLOW CONTROL COMMANDS

6

Flow 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 flow 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 IF command or initiates skipping to a matching ENDIF command.
ENDIF	Terminates skipping initiated by a matching IF, SKIP, or ELSE command.
ENDW	Establishes the end of the loop.
EXIT	Controls the command flow in the event of errors.
IF(or IFE)	Conditionally skips one or more commands.
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.

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 four types of operators: arithmetic, relational, logical, and string. Operators are used in the expressions within the IF, WHILE, DISPLAY, and SET commands and the FILE, STR, STRB, and STRD functions.

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 $2^{48}-1$. The accuracy of computations depends on whether the operands are binary or decimal. Binary operands (specified using a postradix of B) are restricted to 9 digits not including the postradix; results involving binary operands are restricted to 16 digits. Decimal operands and displayed decimal results are restricted to 10 digits (the system interprets each digit as a 6-bit quantity). For both binary and decimal operands, overflow is ignored.†

†If an operand is one of the symbolic names R1, R2, R3, or R1G, there are further restrictions. Refer to the SET command in this section.

The following are the arithmetic operators.

<u>Operator</u>	<u>Operation</u>
+	Addition.
-	Subtraction.
*	Multiplication.
/	Division.
**	Exponentiation.
Leading -	Negation. This operator may be used to obtain a one's complement.
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 all bits of each operand. Each bit of the first operand is compared to the corresponding bit of the second operand. If the comparison is true, the corresponding bit in the result is set to 1. If the comparison is false, the corresponding bit in the result is set to 0. The operator `.NOT.` is a special case, operating on a single operand. If the operand is nonzero, the corresponding bit is set to 0. If the operand is 0, the corresponding bit is set to 1.

If neither of the operands in a logical operation is a string operand, the result of a logical operation is a 60-bit quantity. If one or both is a string operand, the result of the logical operation is a string whose length depends upon the result of the operation. The shorter of the two strings is logically extended with binary zero-fill to the length of the longer string for the logical operation, and trailing binary zero characters are truncated from the result string.

<u>Operator</u>	<u>Operation</u>
<code>.EQV.</code>	Equivalence (complement of the logical sum).
<code>.OR.</code>	Inclusive OR (logical sum).
<code>.AND.</code>	AND (logical product).
<code>.XOR.</code>	Exclusive OR (logical difference).
<code>.NOT.</code>	NOT.

STRING OPERATORS

String operators operate on string operands. A string operand is the result of one of the functions `STR`, `STRB`, `STRD`, or of a logical operation involving one or more string operands.

<u>Operator</u>	<u>Operation</u>
<code>// .CAT.</code>	Concatenation of two strings into one string. (Either <code>//</code> or <code>.CAT.</code> may be used.)

Example:

```
.PROC,CATTER*I,P1=(*A),P2>(*S3/D).
.IF,STR($P1$,1,4)//STRD(P2,-1).EQ.STR($GOOD1$),GO.
NOTE./THE 1ST 4 CHAR OF P1 ARE GOOD AND P2 ENDS IN 1
.ELSE,GO.
REVERT,EX.DISPLAY,STR($SORRY,P1 IS NOT GOOD$).
.ENDIF,GO.
```

In this example, two substrings were concatenated to produce one string in the `.IF` expression. If `GOODY` and `101` were supplied as values for `P1` and `P2`, the `.IF` statement would be true, and the `NOTE` command would be included and executed. If `BADNEWS` were substituted for `P1`, the `.IF` statement would be false, resulting in the execution of the `REVERT` command and then of the `DISPLAY` command.

ORDER OF EVALUATION

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

1. Exponentiation.
2. Multiplication, division.
3. Addition, subtraction, negation, concatenation.
4. Relations.
5. NOT.
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 IF, 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, unless it appears within the STR function, in which case it may be any length up to the maximum length of the line.

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\$	*\$LITERAL\$
\$\$\$\$	\$\$*LITERAL\$
	\$\$LITERAL\$
	\$\$\$

When a literal appears as an operand in a command, a function, or an expression, the system processes it as an integer whose value is equal to the display code representation of the literal value rather than as a character string.

A non-numeric string that appears in an expression to be evaluated must be entered as a literal; otherwise, the system treats it as an erroneous symbolic name.

Example.

```
/IF,AF ILE.EQ.AF ILE,LABEL.
  CCL157- UNKNOWN NAME - AF ILE
```

The above command will execute successfully if the file names are entered as literals (that is, as integer values):

```
/IF,$AF ILE$.EQ.$AF ILE$,LABEL.
```

When you instruct the system to display or compare the value of a literal string, the system right-justifies the display code value of the string and treats it as a numeric value. If the string is the result of the STR, STRB, or STRD function, it is left-justified and treated as a character string. As a result of this different treatment, a literal string within the STR function does not compare equally with the same literal string outside the STR function. In other words, STR(\$ABC\$) is not equal to \$ABC\$.

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
/display,STR($REALLY A STRING$).
REALLY A STRING
/
```

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 zero. The exceptions are shown in the following list:

<u>Name</u>	<u>Description</u>
CSET	Terminal character set mode (NORMAL=0 or ASCII=1).
CS	Connection status.
DATE	The date in the form yymmdd.
DAY	The day of the month.
DAYS	Number of days since January 1, 1977.
HID	Two-character machine identifier (6-bit display code).
MONTH	The month of the year.
NWK	Terminal network connection.
OT	Your job's origin type.
PD	Page density (default=6 lines per inch; allowable values are 6 or 8).
PL	Page length (default=60 lines, minimum=16, maximum=255).
PS	Page size (same as page length).
PW	Page width (default=136 characters, minimum=40, maximum=255).
SC	Your job's service class.
SL	Terminal display mode.
SS	Your current subsystem for interactive jobs.
TIME	Current time of day in the form hhmm.
VER	Operating system version number (6-bit display code).
VERCCL	CYBER control language release level displayed as a numeric value.
WEEKDAY	The day of the week returned as a numerical value.

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 with 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.
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 100 ₈ (refer to RFL Command in section 7).
CS	Connection status.
CSET	Terminal character set mode (NORMAL=0 or ASCII=1).
DATE	The date in the form yymmdd.
DAY	The day of the month.
DAYS	Number of days since January 1, 1977.
DSC	Flag indicating that skipped commands are to be entered in the dayfile.
ECN	Extended memory RFL setting divided by 1000 ₈ (refer to RFL Command in section 7).
EF	Previous error flag.
FAMILY	Current family name.
FL	Current CM field length.
HID	Two-character machine identifier (6-bit display code).
JSN	Four-character job sequence name that uniquely identifies this job (6-bit display code).
MFL	Maximum CM field length.
MFLL	Maximum extended memory field length.
MONTH	The month of the year. (January = 1, February = 2, etc.)
NWK	Network type. Possible values for NWK are: 0 = Terminal not connected to NAM. 1 = Terminal connected to NAM/CCP. 2 = Terminal connected to NAM/CDCNET.
OT	Job origin type.
PACK	Current pack name.
PFDT	Permanent file device type (DJ1, DL2, and so on) possibly set by the PACKNAM command.
SC	Service class of the job.
SL	Terminal display mode (LINE or SCREEN).
SPS	System page size (in lines).
SPW	System page width (in characters).
SPD	System page density (in lines per inch).

<u>Name</u>	<u>Description</u>
SSM	Operating system security mode (0 means unsecured; 1 means secured).
SYS	Host operating system.
TIME	Current time of day (hhmm).
UI	User index associated with the current user name.
UJN	User job name assigned to this job. It defaults to the user index hash for interactive jobs, or it may be set by SETJOB or by the UJN parameter of the ROUTE command for batch jobs initiated by ROUTE.
VER	Operating system version number (6-bit display code).
USER	Current user name.
WEEKDAY	The day of the week (Monday = 1, Tuesday = 2, etc.).

- 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 with fixed values that can be compared with the error flag value (EF or EFG) within an expression. These values correspond 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>
ARE	3	Arithmetic error.
CPE	7	CPU abort.
DRE	37	Deadstart rerun.
ECE	30	Extended memory parity error.
FLE	16	File limit error.
FSE	22	Forced error; set by operator entry of ERR. under DIS utility.

<u>Name</u>	<u>Value (Octal)</u>	<u>Description</u>
IDE	25	Idledown.
ITE	4	SCP invalid transfer address.
JSE	41	Job step abort.
MLE	14	Message limit.
MXE	47	Maximum number of error flags.
ODE	24	Operator drop.
OKE	35	Operator kill.
ORE	46	Override error condition.
PCE	10	PPU call error.
	11	Reserved for installation.
PEE	44	CPU parity error.
PPE	6	PPU abort.
PSE	5	Program stop error.
RAE	40	Recovery abort.
RCE	23	Job hung in auto recall.
RRE	34	Operator rerun.
RSE	31	Recovered system (level 3).
SPE	26	Reserved for your site.
SRE	20	SRU limit error.
SSE	32	Subsystem aborted.
STE	27	Suspension timeout.
SVE	42	Security conflict.
SWE	45	Software error.
SYE	43	System abort.
TAE	2	User break 2.
TIE	1	User break 1.
TKE	17	Track limit error.
TLE	15	Time limit error.

- Symbolic names with fixed values that can be compared with the job's connection status (CS) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
NICS	0	Not interactive.
DTCS	1	Detached.
OLCS	2	Online.

- Symbolic names with fixed values that can be compared with the terminal character set-mode (CSET) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
NORMAL	0	Uppercase characters only.
ASCII	1	Uppercase and lowercase characters.

- Symbolic names with fixed values that can be compared with the month of the year (MONTH) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
JAN	1	January.
FEB	2	February.
MAR	3	March.
APR	4	April.
MAY	5	May.
JUN	6	June.
JUL	7	July.
AUG	8	August.
SEP	9	September.
OCT	10	October.
NOV	11	November.
DEC	12	December.

- Symbolic names with fixed values that can be compared to the terminal's network connection (NWK) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
NONE	0	Terminal has no NAM connection.
CCP	1	Terminal is connected using NAM/CCP.
CDCNET	2	Terminal is connected using NAM/CDCNET.

- 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.
IAO	3	Interactive origin.
RBO	2	Remote batch origin.
SYO	0	System origin.

- Symbolic names with fixed values that can be compared with the service class (SC) value within an expression:

<u>Name</u>	<u>Value</u>	<u>Description</u>
SYSC	1	System service class.
BCSC	2	Local batch service class.
RBSC	3	Remote batch service class.
TSSC	4	Interactive service class.
DISC	5	Detached interactive service class.
NSSC	6	Network supervisor service class.
SSSC	7	Subsystem service class.
MASC	10	Maintenance service class.
CTSC	11	Communication task service class.
IOSC	12	Installation service class 0.
I1SC	13	Installation service class 1.
I2SC	14	Installation service class 2.
I3SC	15	Installation service class 3.
DSSC	77	Deadstart service class.

- Symbolic names with fixed values that can be compared with the terminal display mode (SL) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
LINE	0	Line mode.
SCREEN	1	Screen mode.

- 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.
SC2	2	SCOPE 2 Operating System.

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

<u>Name</u>	<u>Value</u>	<u>Description</u>
ACCESS	6	The access subsystem.
BASIC	1	The BASIC subsystem.
BATCH	5	The batch subsystem.
EXECUTE	4	The execute subsystem.
FORTRAN	2	The FORTRAN 5 subsystem.
FTNTS	3	The FORTRAN Extended 4 subsystem.
NULL	0	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).

- Symbolic names with fixed values that can be compared with the day of the week (WEEKDAY) value within an expression.

<u>Name</u>	<u>Value</u>	<u>Description</u>
MON	1	Monday.
TUE	2	Tuesday.
WED	3	Wednesday.
THU	4	Thursday.
FRI	5	Friday.
SAT	6	Saturday.
SUN	7	Sunday.

FUNCTIONS

Functions are used as expressions or operands within expressions in commands. Functions are not commands. The functions are FILE, DT, LEN, LVL, NUM, STR, STRB, and STRD.

FILE Function

The FILE function determines whether a local 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 attributes. Only the equipment number (EQ) and file size (FS) attributes 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,expression)

<u>Parameter</u>	<u>Description</u>
lfn	Name of the local file for which attributes are being determined.
expression	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 IF command or .IF directive, 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.

<u>Parameter</u>	<u>Description</u>
AP	File has append permission.
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 ordinal of the equipment on which the file resides. If the file is not assigned to a device, it has an EST ordinal of zero.
EX	File has execute permission.
FS	File size in PRUs.
GL	File is a global library.
IN	File type is input.
LB	File is on a labeled tape.
LI	File is a system library file.

<u>Parameter</u>	<u>Description</u>
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.
RU	File has read update permission.
TP	File is on magnetic tape.
TT	File is assigned to a terminal.
UP	File has update permission.
WR	File has write permission.
ZL	File has zero length.

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.

Example:

The following job segment shows the FILE function being used inside an IF command. The FILE function determines if file ACCT is not at the beginning-of-information (BOI). If ACCT is not at BOI, the IF command is true and the system rewinds ACCT before copying it onto ITEM. If ACCT is at BOI, the IF 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.

```
IF,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:

<u>Type</u>	<u>Equipment</u>
DB	885-42 Disk Storage Subsystem (full-track).
DC	895-1/2 Disk Storage Subsystem (full-track).
DD	834 Disk Storage Subsystem (full-track).
DE	Extended memory.
DF	887 Disk Storage Subsystem (4K sector; full-track).
DG	836 Disk Storage Subsystem (full-track).
DH	887 Disk Storage Subsystem (16K sector; full-track).
DI	844-21 Disk Storage Subsystem (half-track).
DJ	844-41/44 Disk Storage Subsystem (half-track).
DK	844-21 Disk Storage Subsystem (full-track).
DL	844-41/44 Disk Storage Subsystem (full-track).
DM	885-11/12 Disk Storage Subsystem (half-track).
DN	9853 Disk Storage Subsystem (2K sector; full-track).
DP	Distributive data path to extended memory.
DQ	885-11/12 Disk Storage Subsystem (full-track).
DV	819 Disk Storage Subsystem (single-density).
DW	819 Disk Storage Subsystem (double-density).
MT	Magnetic tape drive (7-track).
NE	Null equipment.
NT	Magnetic tape drive (9-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.IF,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.
```

LEN Function

The LEN function returns the length of a specified string.

Format:

LEN(string)

<u>Parameter</u>	<u>Description</u>
string	A string of any length up to the maximum length of a line. If the string is a literal, the length returned is the length of the string after the removal of the surrounding dollar signs (\$) and of any extra dollar signs within the string.

LVL Function

The LVL function returns the PSR level (or equivalent) for the NOS operating system or the CYBER Control Language.

Format:

LVL(name)

<u>Parameter</u>	<u>Description</u>
name	Specifies one of the following symbolic values:
<u>name</u>	<u>Description</u>
SYS	Network Operating System
NOS	Network Operating System
CCL	CYBER Control Language, the set of commands that determines the processing sequence within the command record

Default is SYS. Symbols for other operating systems that use the CYBER Control Language are recognized, but return a value of zero.

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(string)

<u>Parameter</u>	<u>Description</u>
string	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 non-numeric.

Example:

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

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

STR Function

The STR function produces a left-justified string of any length up to the maximum line length. The STR function makes it possible to manipulate character strings greater than 10 characters in length. An STR function result may be compared or combined with the results of STRB, STRD or other STR functions in a variety of ways.

Format:

STR(strexp,lc,rc)

<u>Parameter</u>	<u>Description</u>
strexp	May be any legal expression, usually but not necessarily a literal. Literals are left-justified, and may be any length up to the maximum line length. Nonliteral values are treated normally during evaluation of the expression. However, a numeric result will be left-justified and treated as a string.
lc	Must be an expression producing a numeric result, either positive or negative. lc indicates which character of strexp is to be the leftmost character of the string produced by STR, counting from the left if lc is positive or from the right if negative. A value of zero or a negative value which exceeds the length of strexp is treated as equivalent to 1. A positive value greater than the length of strexp produces a null string. lc may be omitted if it would reference the leftmost character of strexp and if rc is also omitted. (The preceding comma must also be omitted when lc is omitted.)

<u>Parameter</u>	<u>Description</u>
rc	<p>Must be an expression producing a numeric result, either positive or negative. rc indicates which character of strexp is to be the rightmost character of the string produced by STR, counting from the left if rc is positive or from the right if negative. A positive value greater than the length of strexp is considered equal to the length. A value of zero or a negative value which exceeds the length of strexp produces a null string. rc may be omitted if it would reference the rightmost character of strexp. (The preceding comma must also be omitted when rc is omitted.)</p> <p>In summary, lc and rc both reference the string strexp in the same way: a positive value of n indicates the nth character of strexp counting from the leftmost character, 1 meaning the first character. A negative value of -n indicates the nth character of strexp counting from the rightmost character, -1 meaning the last character.</p>

Example:

The following expressions all produce the same result, the string \$ABET\$.

```
STR($ALPHABETAG$,5,-3)
STR($ALPHABETAG$,-6,8)
STR($ALPHABET$,5)
```

STRB Function

The STRB function produces a left-justified string of 1 to 17 characters representing the octal value of a specified expression.

Format:

```
STRB(numexp,lc,rc)
```

<u>Parameter</u>	<u>Description</u>
numexp	<p>May be any legal expression, usually but not necessarily numeric. If numexp is or contains a literal, the literal is limited to no more than 10 characters and is right-justified and binary-zero filled for evaluation.</p> <p>The result produced by numexp is treated as a one-word signed value and is converted to a character string in octal. For example, the decimal value 25 for numexp would result in the string \$31\$.</p>
lc	<p>Specifies which character of numexp is to be the leftmost in the resulting string. For further explanation of this parameter, refer to the description of the STR function.</p>
rc	<p>Specifies which character of numexp is to be the rightmost in the resulting string. For further explanation of this parameter, refer to the description of the STR function.</p>

Example:

Each of the following expressions produces the string \$102\$.

```
STRB($ABC$,-5,-3).
STRB(10203B,-5,-3).
```

STRD Function

The STRD function produces a left-justified string of 1 to 16 characters representing the decimal value of a specified expression.

Format:

STRD(numexp,lc,rc)

<u>Parameter</u>	<u>Description</u>
numexp	May be any legal expression, usually but not necessarily numeric. If numexp is or contains a literal, the literal is limited to no more than 10 characters and is right-justified and binary-zero filled for evaluation. The result produced by numexp is treated as a one-word signed value, and is converted to a character string in decimal. For example, the decimal value 25 for numexp would result in the string \$25\$.
lc	Specifies which character of numexp is to be the leftmost in the resulting string. For further explanation of this parameter, refer to the description of the STR function.
rc	Specifies which character of numexp is to be the rightmost in the resulting string. For further explanation of this parameter, refer to the description of the STR function.

Example:

The following expression produces a string of \$0012\$.

STRD(10000+36/3,-4).

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 or in character string form, depending on the expression. (If the expression is an STR, STRB, or STRD function or a concatenation thereof, the resulting string is displayed as a character string; otherwise, the decimal and octal integer format is used.) 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 16 digits can be displayed. For an expression larger than $2^{48}-1$, zeros are displayed.

Format:

DISPLAY,expression₁,expression₂,...,expression_n.

<u>Parameter</u>	<u>Description</u>
expression _i	Specifies any valid symbolic name or expression described earlier in this section.

Example:

The following sample dayfile shows several display operations.

```
12.53.18.DISPLAY,TIME.
12.53.18.    1253    2345B.
12.53.27.SET,R1,=99.
12.53.40.SET,R2,=901.
12.54.05.DISPLAY,R1.
12.54.05.    99    143B.
12.54.16.DISPLAY,R1+R2.
12.54.16.   1000   1750B.
12.55.09.DISPLAY,STR($NOW IS THE TIME$).
12.55.09.NOW IS THE TIME.
12.56.10.DISPLAY,STRB(R1+R2).
12.56.10.1750.
12.56.26.DISPLAY,STRD(R1+R2).
12.56.26.1000.
12.58.26.DISPLAY,STRB(R1+R2)//STR($B IS EQUAL TO $)//STRD(R1+R2).
12.58.26.1750B IS EQUAL TO 1000.
13.01.32.DISPLAY,STR($FIRSTRATE$,4,6)//STR($RINGER$,2,4).
13.01.32.STRING.
13.01.53.DISPLAY,3/2.
13.01.53.    1    1B
13.02.07.DISPLAY,2**47.
13.02.07. GT 999999999    4000000000000000B.
13.02.24.DISPLAY,-2**47.
13.02.24. LT -999999999    -4000000000000000B.
13.02.38.DISPLAY,2**48.
13.02.38.    0    0B.
13.03.28.DISPLAY,9999999999.
13.03.28. 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 next 10 lines illustrate the use of the STR, STRB, and STRD functions for generating display character strings. 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 IF 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 IF command.

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

Format:

ELSE,label.

<u>Parameter</u>	<u>Description</u>
label	Specifies a 0- to 10-character alphanumeric string. The string must begin 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.

```
IF,FILE(TEST1,L0),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.IF,FILE(TEST1,L0),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.IF,FILE(TEST1,L0),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, IF, 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,label.
```

<u>Parameter</u>	<u>Description</u>
label	Specifies a 0- to 10-character alphanumeric string. The string must begin 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,label.
```

<u>Parameter</u>	<u>Description</u>
label	Specifies a 0- to 10-character alphanumeric string. The string must begin with an alphabetic character.

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 prior to the EXIT command. If no error is encountered, the EXIT command indicates where to terminate normal command processing.

When used in procedures, we recommend that the EXIT command be preceded by a SKIP or REVERT command to prevent execution of the EXIT command in the case where no error occurs. An EXIT command executed in this way terminates the executing procedure as well as all calling procedures if nesting has occurred, but does not restore control registers to their previous values.

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.

IF (OR IFE) COMMAND

The IF command conditionally skips one or more commands.

Formats:

```
IF,expression.command.  
IF,expression,label.
```

<u>Parameter</u>	<u>Description</u>
expression	Specifies an expression that is either true or false. The separator following expression in the first format must be a terminator.
command	Specifies a valid command.
label	Specifies a 0- to 10-character alphanumeric string. The string must begin with an alphabetic character. An identical string must appear in a subsequent ELSE or ENDIF command.

The first format conditionally skips only the command specified within the IF command. If expression is true, the system executes the specified command. If expression is false, the system skips the command.

NOTE

When using the first format (IF,expression.command.), values for .command. that match a relational, logical, or string operator (such as .EQ. or .OR.) are interpreted as operators, causing the procedure to abort. To write a valid IF command when .command. matches an operator, terminate the expression or command with a right parenthesis instead of a period.

The second format conditionally skips all commands between the IF command and the first matching ELSE or ENDIF command (two commands match if their label parameters are identical). If expression is true, the system executes all commands between the IF and the matching ELSE or ENDIF. If expression is false, the system skips all intervening commands.

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.  
IF,EF.NE.0.DISPLAY,EF.
```

If the job step executes without error, the error flag (EF) is 0. In this case, the DISPLAY command is not executed. If an error occurs, the error flag is not 0, expression is true, and control passes to the DISPLAY 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 example, the FORTRAN program FTNPROG attempts to call a subroutine BETA which does not exist (outside the field length of the job). The commands that are used appear first and the dayfile segment follows.

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

```
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.IF,EF.NE.0.DISPLAY,EF.  
12.57.04.DISPLAY,EF.  
12.57.04. 3 3B
```

Example 2:

The following procedure file is an indirect access file called COLORPR. It uses the IF 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.  
IF,$COLOR$.EQ.$RED$,L1.  
COMMENT. PROCESSING DONE FOR #COLOR OF COLOR  
REVERT.  
ENDIF,L1.  
IF,$COLOR$.EQ.$BLUES$,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.IF,$BLUE$.EQ.$RED$,L1.  
08.34.32.ENDIF,L1.  
08.34.32.IF,$BLUE$.EQ.$BLUE$,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.IF,$RED$.EQ.$RED$,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.IF,$PINK$.EQ.$RED$,L1.  
08.34.35.ENDIF,L1.  
08.34.35.IF,$PINK$.EQ.$BLUE$,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 ($0 \leq m \leq 17$). Modes 10 through 17 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 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 overflow, indefinite operand, or operand underflow.††
1	Address out of range.	Indefinite operand, operand overflow, or underflow.
2	Operand overflow.†	Address out of range, indefinite operand, or operand underflow.
3	Address out of range or operand overflow.	Indefinite operand or operand underflow.††
4	Indefinite operand.†	Address out of range or operand overflow or underflow.††
5	Address out of range or indefinite operand.	Operand overflow or underflow.††
6	Indefinite operand or operand overflow.	Address out of range or operand underflow.
7	Indefinite operand, address out of range, or operand overflow (default value).	Operand underflow.
10,11†††	Operand underflow or address out of range.	Operand overflow or indefinite operand.
12,13†††	Operand underflow, address out of range, or operand overflow.	Indefinite operand.
14,15	Operand underflow, address out of range, or indefinite operand.	Operand overflow.
16,17	Operand underflow, address out of range, operand overflow, or indefinite operand.	None.

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

††Operand 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.

Descriptions of the error conditions are as follows:

<u>Error Condition</u>	<u>Description</u>
Address out of range error.	An error caused by an attempt to reference central memory or extended memory outside of established limits.
Operand overflow error.	An error resulting when a floating-point arithmetic unit receives an operand with an exponent value of +1777 ₈ (3777 or 4000 in packed form). This is the largest exponent value that can be represented in floating-point format.
Indefinite operand error.	An error resulting when a floating-point arithmetic unit encounters a calculation it cannot resolve. Examples of indefinite operand errors are an attempt to divide 0 by 0 and an attempt to multiply an overflow operand by an underflow operand.
Operand underflow error.	An error resulting when a floating-point arithmetic unit receives an operand with an exponent of -1777 ₈ (0000 or 7777 in packed form). This is the smallest exponent value that can be represented in floating-point format.

For further information about the processing of error mode errors, refer to Error Control in section 3. For further information on the error conditions described above, refer to the appropriate CYBER 180, CYBER 170, CYBER 70, or 6000 Computer System reference manual.

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, symbol₁=expression₁, symbol₂=expression₂, ..., symbol_n=expression_n

symbol_i One of the following symbolic names.

<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. The initial value for each is 0.
RIG	Global control register. When a procedure is called or reverts, RIG keeps its current value. The initial value for RIG is 0.
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 the procedure reverts, the error flag is restored to the value it had when the procedure was called. The initial value for EF is 0.
EFG	Global error flag. When a procedure is called or reverts, EFG keeps its current value if it is not zero. If EFG has a value of zero when the procedure reverts, the system sets EFG equal to EF. The initial value of EFG is 0.
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 two leading periods.
PL or PS	Page length (or page size). Default is 60 lines.
PW	Page width. Default is 136 characters.
PD	Page density. Default is six lines per inch.
SS	Interactive subsystem indicator. The default of SS is NULL.

expression_i 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>symbol</u>	<u>Suggested Value</u>
R1, R2, R3, or R1G	Any integer between -131071 and 131071. If the value is outside this range, it is truncated. The values in these registers are 18-bit quantities, 17 digits and a sign. When truncation occurs, the upper bits are truncated but the sign bit is always retained. The system does not issue a message as a result of the truncation.
EF or EFG	Any integer between 0 and 63. The values in these registers are 6-bit unsigned quantities. When truncation occurs, the upper bits are discarded. If the value is greater than 63, it is truncated. To assign the value defined by the system for an error condition, 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.
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.
PL or PS	Any integer between 16 and 255.
PW	Any integer between 40 and 255.
PD	6 or 8. You can select a print density of either six or eight lines per inch.
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-  
-EOF-
```

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 24B
DISPLAY,EFG.	16.43.35.BEGIN,P2,SETFILE.
	16.43.36.GET,ABC.
	16.43.36. ABC NOT FOUND.
	16.43.36.DISPLAY,EF.
	16.43.36. 6 6B
	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 6. 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.
SET,EFG=0.	09.42.52.SET,EFG=0.
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. 7 7B
	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. 7 7B

The procedure attempts to compile a BASIC program that is not an INPUT record. This generates error code 7 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 7 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,label.

<u>Parameter</u>	<u>Description</u>
label	Specifies a 1- to 10-character alphanumeric string. The string must begin 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.

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,expression,label.
```

<u>Parameter</u>	<u>Description</u>
expression	Specifies any valid symbolic name or expression described earlier in this section. The separator following expression must be a comma.
label	Specifies a 0- to 10-character alphanumeric string. The string must begin 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:

BLOCK	LISTLID	SETASL
CFO	MACHINE	SETCORE
CHARGE	MFL	SETJAL
CHVAL	NORERUN	SETJOB
CLASS	NOTE	SETJSL
COMMENT	OFFSW	SETPR
CTIME	ONSW	SETTL
DAYFILE	PASSWOR	SHELL
DROP	PAUSE	STIME
ENQUIRE	PROTECT	SUBMIT
ENTER	QGET	SWITCH
ERRMSG	QUEUE7	UPROC
GO	REDO	USECPU
HTIME	RERUN	USER
Job	RESOURC	
LDI	RFL	
LENGTH	ROLLOUT	
LIMITS	RTIME	

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.

BLOCK COMMAND

The BLOCK command defines one or more lines of 10 by 10 (10 lines by 10 columns) block letters to be added to a file. The large, 10-line characters provide easier identification of output listings.

Format:

BLOCK, lfn, rewind, cc./line₁/line₂/.../line_n

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the local file to receive the block character lines. Default is OUTPUT.
rewind	Specifies whether or not lfn is rewound before the block character lines are printed. Values that can be specified for rewind are R (rewind) and NR (no rewind). The default is NR.

NOTE

If file lfn contains data and you specify that lfn is to be rewound, the block characters written to the file will overwrite data at the beginning of the file.

cc	Specifies the carriage control character to be inserted before the first line of block characters. The default value is 1 (top of form). The carriage control characters are described in appendix H (a space, PM, and special characters are not accepted).
/	Specifies the delimiting character that separates the block character lines. The delimiter can be any character and must be the first character following the command terminator. Consecutive delimiters generate blank lines.
line _n	Specifies either a string of up to ten characters which constitute one line of block characters, or one of the following special values:

<u>Special Value</u>	<u>Description</u>
DATE	Current date.
TIME	Current time.
USER	Current user name.
UJN	User job name.
JSN	Job sequence name.

Example:

Following is a BLOCK command and its resulting output:

```

/block,blockf.*myjob*date
BLOCK,BLOCKF.*MYJOB*DATE
/list,f=blockf
1
1
      MM      MM YY      YY      JJJJJJJJ 00000000 BBBB BBBB
      MMMM MMMM YY      YY      JJJJJJJJ 0000000000 BBBB BBBB
      MM MMMM MM  YY  YY      JJ  00  000 BB  BB
      MM MM MM  YYYY      JJ  00  0 00 BB  BB
      MM MM MM  YY      JJ  00  0 00 BBBB BBBB
      MM      MM  YY      JJ  00  0 00 BBBB BBBB
      MM      MM  YY      JJ  JJ  00 0 00 BB  BB
      MM      MM  YY      JJ  JJ  000 00 BB  BB
      MM      MM  YY      JJJJJJJ 0000000000 BBBB BBBB
      MM      MM  YY      JJJJJ 00000000 BBBB BBBB
0
88888888 77777777 // 11 0000 // 22222222 88888888
8888888888 7777777777 // 1111 000000 // 2222222222 8888888888
88 88 7 77 // 11 11 00 00 // 2 22 88 88
88 88 77 // 11 00 00 // 22 88 88
88888888 77 // 11 00 00 // 222 88888888
88888888 77 // 11 00 00 // 222 88888888
88 88 77 // 11 00 00 // 222 88 88
88 88 77 // 11 00 00 // 222 88 88
8888888888 77 // 1111111111 000000 // 2222222222 8888888888
88888888 77 / 1111111111 0000 / 2222222222 88888888

```

Column one of the listing contains printer carriage control characters 1 and 0.. The 1 character produces a form feed. The 0 characters produce double spacing.

CFO COMMAND

The CFO (comment from operator) 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,CN=chargenumber,PN=projectnumber.

or

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.

The CHARGE command has four formats:

- If you use the first format, the parameters are order-independent.
- If you use the second format, the parameters are order-dependent.
- If you use the third format, the system uses your default charge and project numbers (refer to the LIMITS command in this section).
- If you use the fourth format, 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 prompts you to enter your charge and project numbers.

The CHARGE command is used in the control of user accounting. The project master user can impose limits on the SRUs your jobs may accumulate under that project number, restrict your access to the system to a certain time of the day, and, in general, monitor your use of the system.

If the specified charge and project numbers have a project prologue associated with them, the project prologue is executed each time the charge number is entered on a CHARGE command. A CHARGE command entered as the first command in the job also causes the system and user prologues, if any, to be executed. The system prologue is executed first, followed by the project and user prologues. For a CHARGE command other than the first CHARGE command in the job, only the project prologue is executed.

If you use a second or subsequent CHARGE command in a job and if the current active charge and project numbers have a project epilogue associated with them, that project epilogue is executed, followed by the specified CHARGE command and then by the project prologue (if any) for the new charge and project numbers.

If your site requires that you use a CHARGE command, the system automatically uses your default charge and project numbers when necessary. For interactive jobs, the system uses your default charge and project numbers each time you log in. If your default charge and project numbers are not defined or are not valid, the system prompts you for valid charge and project numbers. For batch jobs, the system uses your default charge and project numbers if you do not include a CHARGE command immediately following the USER command. If you do include a CHARGE command following the USER command, the system uses the charge number you supply.

The CHARGE command sets the SRU limits for the account block and the job step to the user name validation limits or the account validation limits, whichever is smaller. Refer to the SETJSL and SETASL commands later in this section.

In addition to supplying charge and project numbers at job initiation, you may also be required to supply charge and project numbers under certain other conditions. Specifically, you may be required to enter additional charge and project numbers as a result of an SRU limits exceeded condition or as a result of a project prologue.

If you are assigned more than one charge 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 and 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.

CHVAL COMMAND

The CHVAL command redefines your validation limits or privileges. Currently, you are only allowed to redefine your default service class assignments.

Format:

CHVAL,UC=otsc.

<u>Parameter</u>	<u>Description</u>
UC=otsc	Specifies the default service class assignment according to origin type. The value otsc is the desired service class mnemonic appended to the mnemonic for the origin type. Do not place a separator between the mnemonics. The value ot must be one of the following:

<u>ot</u>	<u>Description</u>
BC	Local batch
IA	Interactive origin
RB	Remote batch

The value sc must be one of the following:

<u>sc</u>	<u>Description</u>
BC	Local batch
CT	Communication task
DI	Detached interactive

<u>Parameter</u>	<u>Description</u>
<u>sc</u>	<u>Description</u>
MA	Maintenance
NS	Network supervisor
RB	Remote batch
SY	System
TS	Interactive
In	Installation class n ($0 \leq n \leq 3$)

You must be authorized to use the service class you select. The LIMITS command shows the service classes you are allowed to use and your default service class assignments.

Example:

In the following example, the default service class assignment for interactive jobs is changed from interactive (TS) to detached interactive (DI).

/limits

.
.
.

THE FOLLOWING ARE ALLOWED SERVICE CLASSES -

BATCH (BC)
REMOTE BATCH (RB)
INTERACTIVE (TS)
DETACHED INTERACTIVE (DI)

THE USER DEFAULT SERVICE CLASS

FOR BATCH ORIGIN IS.....BC
FOR REMOTE BATCH ORIGIN IS.....RB
FOR INTERACTIVE ORIGIN IS.....TS
INQUIRY COMPLETE.

/chval,uc=iadi

UPDATE COMPLETE.

/limits

.
.
.

THE FOLLOWING ARE ALLOWED SERVICE CLASSES -

BATCH (BC)
REMOTE BATCH (RB)
INTERACTIVE (TS)
DETACHED INTERACTIVE (DI)

THE USER DEFAULT SERVICE CLASS

FOR BATCH ORIGIN IS.....BC
FOR REMOTE BATCH ORIGIN IS.....RB
FOR INTERACTIVE ORIGIN IS.....DI
INQUIRY COMPLETE.

CLASS COMMAND

The CLASS command changes the service class of your current interactive job or of a batch job you have submitted for execution. For interactive jobs, the CLASS command can only change the service class of the interactive job from which it was entered; it cannot change an interactive job connected to another terminal. For batch jobs, you can change the service class of any executing job or any job in the input queue. This includes the job from which the CLASS command was entered.

The CLASS command also shows the relative priority of the available service classes for a specified origin type. Implicitly, the relative priority display also shows which service classes are allowed within the chosen origin type.

Format:

CLASS,SC=sc,OP=A.

or

CLASS,sc,,,A.

or

CLASS,OT=ot,L=lfm.

or

CLASS,,ot,lfm.

or

CLASS,SC=sc,JSN=jsn.

or

CLASS,sc,,,jsn.

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

<u>Parameter</u>	<u>Description</u>
SC=sc	Specifies the desired service class for the job. The value sc must be one of the following:
<u>sc</u>	<u>Description</u>
SY	System
BC	Local batch
RB	Remote batch
TS	Interactive

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<u>sc</u>	<u>Description</u>
DI	Detached interactive
NS	Network supervisor
MA	Maintenance
CT	Communication task
In	Installation class n ($0 \leq n \leq 3$)

There is no default for this parameter. The service class you specify must be within your validation limits (refer to the LIMITS command).

OP=A Selects the abort option. The command aborts if too many jobs in the system already have the service class you specified. Otherwise, if this situation occurs, the system rolls out your job and waits until it can make the desired assignment.

OT=ot Specifies the origin type for which the system shows the relative priority of the available service classes. The value ot can be any of the following:

<u>ot</u>	<u>Description</u>
SY	System origin
IA	Interactive origin
BC	Local batch origin
RB	Remote batch origin

The default is the origin type of the current job. If you specify OT=ot along with SC=sc, OT=ot is ignored.

L=lfm Specifies the file that receives the display generated by this command. OUTPUT is the default. If you specify L=lfm along with SC=sc, L=lfm is ignored.

JSN=jsn Specifies the JSN of an executing job or a job in the input queue.

The first and second formats change the service class of your current job. If you use the first or second format and do not specify a valid service class (for instance, you omit that value), the system displays the relative priority of the service classes available for your job. Further, if your job is interactive, you receive a prompt for a service class.

The third and fourth formats generate a display of the relative priority of the available service classes for a designated origin type.

The fifth and sixth formats change the service class of an executing batch job or of a job in the input queue. They cannot be used to change the service class of an interactive job.

Example:

The following terminal session shows how you can ascertain the relative priority of the service classes for your current job.

```
/class
          AVAILABLE SERVICE CLASSES
          ---RELATIVE PRIORITY---
CLASS INPUT FILES  EXECUTING JOBS  OUTPUT FILES
DI      *          *****      *****
IO      *          *****      *****
TS      *****    *****      *****  CURRENT

ENTER CLASS: di
CLASS COMPLETE.
/
```

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.

For printed DAYFILE output, you can specify output page size and print density using the PL and PD parameters. The DAYFILE command also honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

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	Specifies the 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 an 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.

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

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.

PD=pd Specifies the print density; default is PD=6.

<u>pd</u>	<u>Print Density</u>
3	Double space; 6 lines per inch
4	Double space; 8 lines per inch
6	Single space; 6 lines per inch
8	Single space; 8 lines per inch

PL=pl Specifies page size; if specified, the actual page size used is determined by page density (pd) and the value specified for pl, as follows:

<u>pd</u>	<u>Page Size</u>
3	pl / 2
4	pl / 2
6	pl
8	pl

If this parameter is omitted, page size is determined from system page size and print density as follows:

<u>pd</u>	<u>Default Page Size</u>
3	30 lines
4	30 lines
6	60 lines
8	60 lines

I=infile Specifies a file which contains a dayfile 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).

Format:

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

or

DROP,jsn,q,ujn,R.

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 ALL.																
	<table><thead><tr><th><u>q</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>WT</td><td>Jobs queued with a wait disposition.</td></tr><tr><td>PR</td><td>Jobs queued for printing.</td></tr><tr><td>PU</td><td>Jobs queued for punching.</td></tr><tr><td>PL</td><td>Jobs queued for plotting.</td></tr><tr><td>IN</td><td>Jobs queued for input.</td></tr><tr><td>EX</td><td>Executing jobs.</td></tr><tr><td>ALL</td><td>All jobs belonging to you.</td></tr></tbody></table>	<u>q</u>	<u>Description</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.
<u>q</u>	<u>Description</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.																
OP=R	Directs the system to drop the specified executing jobs without EXIT processing but with single-reeve processing for jobs containing such.																

Examples:

DROP,ABDA. Drops executing job ABDA.

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

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

DROP,JSN=ABFF,OP=R. Drops executing job ABFF with a single reeive and without EXIT processing.

ENQUIRE COMMAND

The ENQUIRE command gives you information about your jobs in the system. For output listing files not connected to your terminal, the ENQUIRE command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

Format:

ENQUIRE,OP=p₁p₂...p_n,FN=lf_{n1},O=lf_{n2}.

or

ENQUIRE,p₁p₂...p_n.

or

ENQUIRE,JSN=js_n,O=lf_{n2}.

or

ENQUIRE,UJN=uj_n,O=lf_{n2}.

or

ENQUIRE.

<u>Parameter</u>	<u>Description</u>
OP=p ₁ p ₂ ...p ₇	Any combination of the following options. You can request a maximum of seven options on a command (for example, ENQUIRE,OP=BDFJLRS.).

<u>P_i</u>	<u>Description</u>
A	Gives listings of the B, D, R, U, J, L, and F options. If you specify no parameters or only the O=lf _{n2} parameter in a batch job, the OP=p ₁ p ₂ ...p _n parameter defaults to OP=A.
B	Returns identification and priority information. On a secured system, the system also lists the current job access level, the upper limit for the job access level, the lower limit for the job access level, and the security access categories you can assign to permanent files.

Parameter

Description

Pi

Description

Example:

/enquire,b

SYSTEM ACTIVITY.

USER NAME	STARSKY
USER INDEX HASH	AN2A
JOB SEQ. NAME	ABLE
FAMILY	NOSHOP
CHARGE NUMBER	2222
PROJECT NUMBER	666N666
PACK NAME	*NONE*.
PACK TYPE	DI1
PRIMARY FILE	*NONE*.
SUBSYSTEM	BATCH.
CPU PRIORITY	30
MAX FL (CM)	203700
MAX FL (EM)	0
LAST FL (CM)	0
LAST FL (EM)	0

D Returns a listing of the resources you have demanded and those which have been assigned (refer to RESOURC Command in this section 6).

Example:

RESOURCE DEMAND INFORMATION.

RESOURCE	DEMAND	ASSIGNED
PE	1	0
HD	1	0

Parameter

Description

Pi

Description

F Gives the status of files assigned to your job.

Example:

/enquire,f

LOCAL FILE INFORMATION.

FILENAME	LENGTH/PRUS	TYPE	STATUS	FS	LEVEL
INPUT*	1	IN.*	EOR		
PROLOG	1	LO.	EOR	NAD	
PROCFIL	3	LO.	EOI		
INPUT		TT.			
OUTPUT		TT.			
PRCFIL	6	LO.	EOR	NAD	
ZZZZLD	5	LO.	EOR	NAD	
PUBTOOL	835	PM.*	EOR	NAD	
LIBRT	6429	PM.**	INACCESS*	NAD	

TOTAL = 8

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. For magnetic tape files (TYPE=TP), the current position block number is displayed in the LENGTH/PRUS column instead of the file length. 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 entry *INACCESS* indicates that the file is inaccessible; it resides on a mass storage device that is currently off or down. The FS column indicates whether a file has the no-auto-drop status (refer to the SETFS command). The LEVEL column gives the security access level of the file.

On an unsecured system, the security access level is listed only if it is not null.

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 = 0
R2 = 0
R3 = 0
EF = 0
EFG = 0
R1G = 0

JOB PAGE PARAMETERS.

PW = 136
PL = 60
PD = 6

PENDING COMMANDS

WHILE,R1<20,COPY.
COPYEI,AGENDA.
REWIND,AGENDA.
SET,R1=R1+1.
ENDW,COPY.
REWIND,ENQWAJ.
COPYSBF,ENQWAJ.
REPLACE,ENQWAJ.

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

Parameter

Description

Pi

Description

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 and job step SRU limits (refer to the CHARGE command).

Example:

RESOURCE USAGE ALLOWED.

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

Parameter

Description

UJN=ujn

Returns a one-line report of each of your jobs with the specified user job name (UJN). The report for each job has the following fields:

jsn sc cs ds lid ujn status executing message

The fields have the following descriptions:

Field

Description

jsn The job sequence name (JSN). It is a four-character, system-defined name.

sc The service class of the job. One of the following mnemonics appears in this field:

<u>sc</u>	<u>Service Class</u>
S	System
B	Batch
R	Remote batch
T	Interactive
D	Detached interactive
N	Network supervisor
C	Communication task
M	Maintenance
0	Installation class 0
1	Installation class 1
2	Installation class 2
3	Installation class 3

cs The connection status (CS) of your job; that is, whether your job is online and interactive, detached and interactive, not interactive or queued. The following values can appear:

<u>cs</u>	<u>Description</u>
ON	Online and interactive
DT	Detached and interactive
NI	Not interactive
blank	Queued

Parameter

Description

Field

Description

ds The default destination of batch output. It indicates whether batch output will be routed to the central site (BC) or to a remote batch terminal (RB).

lid The logical identifier (LID) of the mainframe that is to process the job. This is a three-character, site-defined mnemonic.

ujn The user job name (UJN) of the job. It is either the name you assigned the job or it is your user index hash if you did not assign the job a name.

status The overall job status. It indicates whether the job is executing, whether it has been rolled out, or the queue in which it now resides.

executing message The command that the system is now executing.

Example:

/enquire,ujn=ldijob

JSN	SC	CS	DS	LID	UJN	STATUS	EXECUTING MESSAGE
ABCF.B.	.BC.	.LDIJOB				WAIT	QUEUE

If you specify only the keyword UJN, the system returns a one-line report for each job associated with your user name.

Example:

/enquire,ujn

JSN	SC	CS	DS	LID	UJN	STATUS	EXECUTING MESSAGE
ABCH.B.	.BC.	M42.	LEVEL			INPUT	QUEUE
ABCB.T.	ON.	BC.	.AJLA			EXECUTING	
ABCI.B.	.BC.	M42.	LDIJOB			PRINT	QUEUE
ABCF.B.	.BC.	.LDIJOB				WAIT	QUEUE

Do not specify the UJN=ujn parameter with any parameter other than O=lf_n2.

JSN=jsn

Returns a detailed report of the job with the specified job sequence name (JSN). The format of the report varies with the status of the job. If the job is executing, the report includes the JSN, the job's service class, its connection status, the default destination of its batch output, the current status of the job, the user job name (UJN), the job access level (on a secured system only), job resource information, the executing command, a dayfile extract (maximum of 10 messages), and a list of the next commands to be executed (maximum of 10 commands).

Parameter

Description

Example:

/enquire,jsn=abcb

ABCB.T.ON.BC. .EXECUTING UJN=AJLA LEVEL=LVLO
SRUS= 2.573. SRU LIMIT=NO LIMIT. CM FL= 5000. ECS FL= 0.

DAYFILE=

17.26.55. SUBMIT COMPLETE. JSN IS ABCE.
17.27.07.GET,OPLJOB.
17.27.21.REPLACE,CATEGOR.
17.27.32.SUBMIT,OPLJOB,TO.
17.27.32. SUBMIT COMPLETE. JSN IS ABCF.
17.28.04.ROUTE,CATEGOR,DC=IN.
17.28.04. ROUTE COMPLETE. JSN IS ABCH.
17.28.13.ENQUIRE,UJN.
17.28.26.ENQUIRE,JSN.
17.29.04.ENQUIRE,JSN=ABCB.

If the job is a queued file, the report includes the job's JSN, its service class, the default destination of its batch output, the logical identifier (LID) of the mainframe that is to process the job, its queue residence, its UJN, length in PRUs, and output specifications.

Example:

/enquire,jsn=abcf

ABCF.B. .BC. .WAIT QUEUE UJN=LDIJOB LEVEL=LVLO
PRU LENGTH= 32. DC=TT. FC= . ID= 0. EC= .

If you specify the keyword JSN, the system returns a one-line report on each job associated with your user name. The fields in the display are identical to those you get when you specify only the keyword UJN, except that the UJN field and the EXECUTING MESSAGE fields are omitted.

Example:

/enquire,jsn

JSN SC CS DS LID STATUS JSN SC CS DS LID STATUS
ABCH.B. .BC.M42.INPUT QUEUE ABCB.T.ON.BC. .EXECUTING
ABCI.B. .BC.M42.PRINT QUEUE ABCF.B. .BC. .WAIT QUEUE

Do 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: BATCH 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 ctE or ctS 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#1go#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.

ERRMSG COMMAND

The ERRMSG command controls the display of error messages generated by commands executed within a NOS procedure. An ERRMSG,OFF command inhibits the display of error messages at the terminal, although all messages are still listed in the dayfile. This command has no effect when issued from a batch job.

The ERRMSG command must be executed from within a procedure. The error message off status remains in effect until an ERRMSG,ON command is issued, or until your job returns to interactive command status. The error message off status is not affected by the entry of subsequent BEGIN or REVERT commands (that is, nested procedures).

Format:

ERRMSG,status

<u>Parameter</u>	<u>Description</u>
status	Specifies whether error messages generated by commands within a procedure are displayed at the terminal. Values that can be specified for status are OFF (messages are not displayed) and ON (messages are displayed). The default value is ON.

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.
-----	---

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).

If you want to determine whether a PAUSE or GO command has been entered, write a program to set bit 13 of word RA and then status it. The system unconditionally clears bit 13 of word RA whenever the PAUSE/GO command sets/clears the pause flag (bit 12 of word RA). Hence, when the system clears it, you can check the pause flag. CPU programs that check the pause flag on NOS 1 continue to work properly on NOS 2, but these programs cannot determine when a PAUSE or GO command has been entered.

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.

Format:

ujn,P=priority,T=timelimit,CM=memory,EC=xmemory,ST=lid,AL=level.punchmode

or

ujn,Ppriority,Ttimelimit,CMmemory,ECxmemory,STlid,ALlevel.punchmode

or

ujn,SC=class,T=timelimit,CM=memory,EC=xmemory,ST=lid,AL=level.punchmode

or

ujn,SCclass,Ttimelimit,CMmemory,ECxmemory,STlid,ALlevel.punchmode

or

ujn,priority,timelimit,memory,xmemory,lid,level.punchmode

or

ujn,class,timelimit,memory,xmemory,lid,level.punchmode

All parameters except the parameter ujn are optional. If you include keywords when you specify optional parameters, they are order-independent. Otherwise, they are order-dependent.

Parameter

Description

ujn

Alphanumeric user job name (UJN) of one to seven characters. This name identifies the individual jobs being run under the same user name.

P=priority

Specifies the priority level at which the job enters the system. Using this priority level, the system assigns the job a service class according to a site-defined algorithm. The value you specify must be an integer in the range of 0 through 7. The default for P=priority depends on your default service class assignments and your site-defined algorithm for mapping priority levels and service classes (refer to Job Service Classes in section 3).

To assign a service class to your job directly, use the SC=class parameter.

SC=class

Specifies the service class to be assigned to the job. The value class must be one of the following values:

<u>class</u>	<u>Description</u>
SY	System

Parameter

Description

<u>class</u>	<u>Description</u>
BC	Local batch
RB	Remote batch
TS	Interactive
DI	Detached interactive
NS	Network supervisor
SS	Subsystem
MA	Maintenance
CT	Communication task
In	Installation-defined class n ($0 \leq n \leq 3$)

The default for SC=class depends on your default service class assignments (refer to Job Service Classes in section 3).

AL=level

Specifies the maximum security access level your job can attain. Unless changed by your site, the value level can be any one of the following names:

LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7

You may not be authorized to use all of these access levels (refer to the LIMITS command). The default is the lowest access level you are authorized to use. On an unsecured system, this parameter has no effect, but the system does ensure that you specify a valid level name.

T=timelimit

Specifies the central processor job step time limit in seconds. Values can range from 1 to 99999. Decimal is the default base. Octal values from 1 to 77777₈ (1 to 32767) can be entered if followed by a B suffix. The default limit is 64 (100₈).[†] Decimal values from 32767 to 99999, or values that are greater than your validated time limit, 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).

CM=memory

Specifies the 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 100₈. The default value 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 376500₈. This is the absolute upper limit; however, your site may choose a lower limit.

[†]Your site can change this default.

<u>Parameter</u>	<u>Description</u>
	A value containing an 8 or 9 or the suffix D is interpreted as decimal.
EC=xmemory	Specifies the maximum octal number of 1000 ₈ , 2000 ₈ , or 4000 ₈ word extended memory blocks required by the job. The extended memory block size is site dependent; the default block size is 1000 ₈ words.† This extended memory field length cannot exceed 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.
ST=lid	Specifies the logical identifier (LID) of the host to which you are routing the specified job. The LID must be a three-character alphanumeric string defined by site personnel. The local host is the default. You must have special authorization to use this parameter (refer to the LIMITS command).
punchmode	Specifies the conversion mode for card decks. This parameter must be entered in columns 79 and 80. A 26 indicates that coded cards are to be converted in 026 mode; 29 indicates cards are converted in 029 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 with mode 4 card readers; it is effective for remote batch jobs entered with HASP, 2780, and 3780 card readers.

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 central memory 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,CM=50000,EC=20,T=1.

† Refer to the UEBS parameter in common deck PPCOM under Installation Parameters in the NOS 2 Installation Handbook.

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, lfn, 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. <table><thead><tr><th><u>dc</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>IN</td><td>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).</td></tr><tr><td>NO</td><td>Discards all output.</td></tr><tr><td>TO</td><td>Queues the output with a wait disposition.</td></tr></tbody></table>	<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>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.								
UN=un	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). You may not specify this parameter if you specify the ID=id parameter.								

<u>Parameter</u>	<u>Description</u>
FM=fn	Specifies the family name of the remote batch user to whom the system routes each job's output. You must have special authorization to specify an alternate family name (refer to the LIMITS command). 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).

You may not specify this parameter if you specify the ID=id parameter.

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.

<u>Parameter</u>	<u>Description</u>
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.

<u>Parameter</u>	<u>Description</u>
L=lfm	Specifies 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 the following categories:

- Resource Limits.
- Other Characteristics.
- User Permissions.
- Network Applications.
- Privileged Network Applications.
- Local Applications.
- Shell Permissions.
- Security Permissions.
- Security Access Categories.
- Security Access Levels.
- Service Classes.
- Default Service Classes.

If there are no entries for a category, the system suppresses the header for that category.

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.

†The value SYSTEM denotes that you do not have a static limit for the number of permanent files allowed in your catalog. Whenever you initiate a permanent file operation, the system assigns a limit, which can vary with the service class of your job.

OTHER CHARACTERISTICS

This category lists some general characteristics associated with your user name and your jobs. The list may include any of the following:

- User index.
- Default charge number.
- Default project number.
- User prologue.
- System prologue.
- Default subsystem for interactive jobs.
- Default character set mode for interactive jobs.
- Expiration dates for your user passwords.

USER PERMISSIONS

In addition to specifying resource limits for certain resources, the system lists special privileges that you possess. The following is a list of these special privileges:

- Change your passwords.
- 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 the BIO subsystem privilege to use the V carriage control character in a user-specified PFC array in a file.
- Use the PROTECT command.
- Transfer permanent files between hosts.
- Transfer queued files between hosts.
- Access the NVE subsystem.
- Specify charge numbers other than the default charge number.
- Define a password without random characters.
- Have remote maintenance privileges.
- Control network processing units (NPU's).
- Specify an alternate family during a job.
- Specify mainframe logical identifiers (LIDs) on the Job and ROUTE commands.
- Specify permanent file residence as Locked to Disk (PR=L).
- Access the system without specifying a personal identifier.
- Use more than one interactive terminal at the same time.
- Perform privileged analyst operations if PRIVILEGED ANALYST MODE is enabled.

NETWORK APPLICATIONS

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

- Interactive Facility (IAF).
- Interactive Transfer Facility (ITF).
- Message Control System (MCS).
- Network Operator Utility (NETOU).
- Network Job Entry Facility (NJF).
- PLATO - NAM Interface (PLATO).
- Printer Support Utility (PSU).
- Remote Batch Facility (RBF).
- Terminal Verification Facility (TVF).
- Tieline Facility (TLF).
- Transaction Facility (TAF).
- NOS/VE Interactive Facility (VEIAF).

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.

PRIVILEGED NETWORK APPLICATIONS

The system lists the privileged network applications you are authorized to use. The Communications Supervisor (CS) is one such application.

SHELL PERMISSIONS

The system lists controls applicable to a shell program you write to monitor commands entered under your user name. The list includes the following:

Clear controls upon loading errors with shell program.

Clear shell processing if screening program aborts.

Local file load of shell program.

Global file load of shell program.

System library load of shell program.

Monitor commands from terminal.

Monitor commands outside of procedures.

Monitor commands inside of procedures.

SECURITY PERMISSIONS

There are special security-related permissions you may possess. The list may include any of the following permissions:

- Write unlabeled magnetic tapes.
- Write to or extend a file with an access level lower than your job access level.
- Lower file access levels.
- Lower your job access level.
- Assign an expiration date for a file password or file permission.
- Assign an expiration date for your user password.
- Execute on-line diagnostics.
- Perform security administrator duties.

SECURITY ACCESS CATEGORIES

The system lists the access categories you can associate with your permanent files. When you create a permanent file, the system associates all categories for which you are authorized with this file. You can change this association with the SETPFAC command.

SECURITY ACCESS LEVELS

The system lists a set of access levels that determine the access levels at which your jobs may run and the access levels at which you may create or access files.

SERVICE CLASSES

The system lists all of the service classes you can assign to a job. Your site can further restrict the use of a service class to jobs of certain origin types. You can determine these restrictions with the CLASS command.

DEFAULT SERVICE CLASSES

The system lists the service class it initially assigns to your job if you do not specify a service class on the Job command or ROUTE command. This default assignment can vary with the origin type of your job. Unlike most other validation limits and privileges, you can change this default assignment with the CHVAL command.

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

/limits
LIMITS FOR STARSKY FM/NOSCLSH 83/08/09. 17/11/13/

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.....	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.....	UNLIMITED

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 FOR 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.....	5410B
DEFAULT CHARGE NUMBER ASSIGNED.....	1234
DEFAULT PROJECT NUMBER ASSIGNED.....	123A456
SYSTEM PROLOGUE.....	2B
USER PROLOGUE.....	PROLOG
INTERACTIVE SUBSYSTEM.....	BATCH
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
TRANSFER PERMANENT FILES BETWEEN HOSTS
TRANSFER QUEUED FILES BETWEEN HOSTS
SPECIFY LID-S ON JOB AND *ROUTE* COMMANDS
CHARGE NOT RESTRICTED TO DEFAULT
DEFINE NON-RANDOMIZED LOGIN PASSWORD

THE FOLLOWING ARE VALID NETWORK APPLICATIONS -
INTERACTIVE FACILITY (IAF)
REMOTE BATCH FACILITY (RBF)
TRANSACTION FACILITY (TAF)
TERMINAL VERIFICATION FACILITY (TVF)

THE FOLLOWING ARE VALID SHELL PERMISSIONS -
PROCESS COMMANDS OUTSIDE CCL PROCEDURE
SYSTEM LIBRARY LOAD OF SHELL PROGRAM

THE FOLLOWING ARE VALID SECURITY PERMISSIONS -
WRITE UNLABELED MAGNETIC TAPES

THE FOLLOWING ARE VALID SECURITY ACCESS LEVELS -
ACCESS LEVEL LVLO

THE FOLLOWING ARE ALLOWED SERVICE CLASSES -
BATCH (BC)
REMOTE BATCH (RB)
INTERACTIVE (TS)
DETACHED INTERACTIVE (DI)

THE USER DEFAULT SERVICE CLASS
FOR BATCH ORIGIN IS.....BC
FOR REMOTE BATCH ORIGIN IS.....RB
FOR INTERACTIVE ORIGIN IS.....TS
INQUIRY COMPLETE.

LISTLID COMMAND

The LISTLID command lists information on the configuration of your network and on the availability of hosts in the network. Information available through LISTLID includes:

- Logical identifiers (LIDs) and physical identifiers (PIDs) for hosts in the network. LIDs are used in certain NOS commands (for example, Job, ROUTE, MFLINK, and MFQUEUE) to specify the destination host for a job or file routing operation.
- The subsystem(s) available to provide network communications. This could be any or all of the following: Remote Host Facility (RHF), Network Access Method (NAM), and SCOPE 2 Station Facility (SSF).
- Host characteristics such as local host or linked status, store and forward capabilities, and host availability.†

Note that the site administration has the option of choosing whether information for any given LID or PID can be listed using the LISTLID command.

Format:

LISTLID,LID=lid,PID=pid,l=lfn

or

LISTLID,ST=lid,PID=pid,l=lfn

<u>Parameter</u>	<u>Description</u>
LID=lid or ST=lid	Specifies the LID for which information is to be displayed. If LID=lid or ST=lid is specified, information is listed for the specified LID. (The LID and ST keywords are interchangeable; either may be used, but not both.) If this parameter is omitted and the PID parameter is also omitted, information is listed for all LIDs in the network.
PID=pid	Specifies that output is to be listed only for LIDs associated with this PID. If the LID=lid parameter is also specified, information is listed only for that LID. If no LID=lid parameter is specified, information is listed for all LIDs associated with this PID. If only the keyword PID is specified, information is listed on all LIDs, grouped according to their associated PIDs.
L=lfn	Specifies that LISTLID output is to be written to file lfn. If this parameter is omitted, output is written to file OUTPUT.

Examples:

LISTLID.

This command lists information on all LIDs in the network for which the site allows information to be listed. The information is listed in output format 1.

LISTLID,LID=lid,L=lfn.

This command lists information on all LIDs in the network with the name lid except those that the site does not allow to be listed. The information is listed in output format 1 on file lfn.

†Hosts connected through an X.25 network are always considered available as long as the network processing unit (NPU) connecting your host to the X.25 network is available.

LISTLID,LID=lid,PID=pid.

This command provides information on the named LID associated with the specified PID. If the site does not allow listing of the named LID, or of both the named LID and PID, the system returns a dayfile message indicating that the specified LID was not found. If the site allows the LID to be listed but not the PID, LISTLID outputs information using format 2; however, the system will not display the PID name or the names of available subsystems. Instead, the system replaces those names with the characters ---.

LISTLID,LID=lid,PID.

This command lists all LIDs with the name lid except those that the site does not allow to be listed. The specified LIDs are listed under their associated PIDs using output format 2.

LISTLID,PID.

This command uses output format 2 to list all LIDs, grouped according to their associated PIDs. Any LIDs that the site does not allow to be listed are omitted from the output. Any PIDs that cannot be listed are replaced by the characters ---.

LISTLID,PID=pid.

This command uses output format 2 to list all LIDs associated with the specified PID. Any LIDs that the site does not allow to be listed are omitted from the output. If the PID cannot be listed, pid is replaced with the characters ---.

LISTLID uses two output formats. Output format 1 is used when the LISTLID command does not contain a PID parameter.

- Example of output format 1:

<u>LID</u>	<u>MAINFRAME</u>	<u>ATTRIBUTE</u>	<u>NETWORK</u>
M64	NOS 964	LOOPBACK	RHF,SSF
M42	NOS 420	LINKED	NONE AVAILABLE
B2C	VSOS	LINKED	RHF
MFZ	SCOPE2	LINKED	SSF
MS1	NOS MS1	LINKED	NAM

Output format 2 is used when a PID parameter is specified on the LISTLID command.

- Example of output format 2:

<u>PID</u>	<u>LID</u>	<u>MAINFRAME</u>	<u>TYPE</u>	<u>ATTRIBUTE</u>	<u>RHF</u>	<u>NAM</u>	<u>SSF</u>
M64		SN 964	HOST	HV	IN	NA	DISABLED
	M64						
	LBK			B			

In output format 2, the attributes listed for each PID are also applicable for each LID listed under that PID. In the above example, therefore, LID M64 has the attributes H and V, while LID LBK has the attributes H, V, and B.

Attributes listed in the ATTRIBUTE column can be listed by name or by mnemonic.

<u>Attribute Name</u>	<u>Mnemonic</u>	<u>Description</u>
HOST	H	Indicates that this is the local host (that is, the host you are logged into).
LINKED	L	Indicates that this host is linked to your local host.
STORE FORWARD	S	Indicates that you can transfer queued files to this host through an intermediate linked host.
VALIDATION	V	Indicates that user validation is performed by the local host as well as by the destination host. If this attribute is not listed for your local host, user validation is performed only by the destination host.
LOOPBACK	B	Indicates that the LID is a loopback LID (used for test purposes).

Entries that can be listed in the RHF, NAM, and SSF columns are as follows:

<u>Entry</u>	<u>Significance</u>
IN	Indicates that the link to the PID is in service. Note that for RHF networks and NAM X.25 public data networks, an in service (IN) status is returned whenever the network is in service, without regard to whether the remote host itself is available. To determine if the remote host is available, you can attempt a file transfer to that host.
NA	Indicates that the subsystem is not available or that no link exists to this PID through this subsystem.
OUT	Indicates that the link to the PID is out of service due to a network fault condition.
DISABLED	Indicates that the link to the PID has been disabled by the console operator.

MACHINE COMMAND

Some programs that ran on mainframes other than models 815, 825, 835, 845, and 855 will not run on the models 815, 825, 835, 845, and 855 until changes are made in your job. Programs most likely to run incorrectly on models 815, 825, 835, 845, and 855 are COMPASS programs with tight instruction-modification loops. The MACHINE command enables some of these programs to run correctly on these models.

Format:

MACHINE,EP=state.

<u>Parameter</u>	<u>Description</u>
EP=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 815, 825, 835, 845, 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 815, 825, 835, 845, and 855.

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-independent form and the second format is the order-dependent 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 at deadstart or 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	Specifies the 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.
/	Specifies the 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	Specifies the 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*I,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,switch1,switch2,...,switchn,jsn.
```

<u>Parameter</u>	<u>Description</u>
switch _i	Specifies the switches to be cleared. You must denote each 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.
jsn	Specifies the job sequence name (JSN) of the job whose sense switches will be changed. jsn is an optional parameter; its default is the JSN of the current job. The system recognizes the jsn parameter by its alphabetic characters; therefore, jsn can appear anywhere in the parameter list and can be omitted without leaving a blank parameter field.

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,switch₁,switch₂,...,switch_n,jsn.

<u>Parameter</u>	<u>Description</u>
switch _i	Specifies the switches to be set. You must denote each 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.
jsn	Specifies the job sequence name (JSN) of the job whose sense switches will be changed. jsn is an optional parameter, its default is the JSN of the current job. The system recognizes the jsn parameter by its alphabetic characters; therefore, jsn can appear anywhere in the parameter list and can be omitted without leaving a blank parameter field.

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

PASSWOR COMMAND

The PASSWOR command changes the passwords associated with your user name. To use PASSWOR, you must be authorized to change your password. (Refer to the LIMITS command to determine if you are authorized to change your password.) Your user name has two passwords:

- The interactive password, which must be entered when logging in.
- The batch password, which must be included in batch and remote batch jobs.

You may specify the password you want the system to change. If you omit this parameter selection, the system changes the password that matches the origin type of your job. That is, a PASSWOR command in an interactive job changes your interactive password; a PASSWOR command in a batch or remote batch job changes your batch password.

If your site uses the password randomization feature, you can enter the PASSWOR command only from an interactive job. When you select a new password, the system appends a site-determined number (from two to five) of randomly generated characters to it.

Format:

PASSWOR,oldpassword,newpassword,XD=expiredate,PT=passwordtype.

or

PASSWOR,oldpassword,newpassword,XT=expireterm,PT=passwordtype.

or

PASSWOR.

<u>Parameter</u>	<u>Description</u>
oldpassword	Specifies the old password. This is a required parameter if password randomization is used.
newpassword	Specifies the new password. The new password must not be more than seven characters long. Your site determines the minimum length of passwords (4 is the default). This is a required parameter if password randomization is used.
XD=expiredate	Specifies the expiration date for the password. The value expiredate must be a six-digit string of the form yymmdd, where yy, mm, dd are two-digit year, month, and day designators, respectively. The latest date you can specify and the default expiration date are site-determined. You must have special authorization to specify an expiration date for a password (refer to the LIMITS command).
XT=expireterm	Specifies the life of the password in days. If you specify 0, the password expires immediately. If you specify 4095 or *, the password does not expire. The default expiration term and the maximum expiration term are site-determined (no site can choose a value greater than 4095). You must have special authorization to specify an expiration term for a password (refer to the LIMITS command).
PT=passwordtype	Specifies the type of password to be changed.

<u>passwordtype</u>	<u>Description</u>
B	Batch.
I	Interactive.

If you omit the PT parameter, the password that matches the origin type of your job is changed.

You cannot specify a new expiration date (using either XD or XT) without also specifying the oldpassword and newpassword parameters. Also, if you omit both the XD and XT parameters when you enter the PASSWOR command, the default expiration date is the current date plus the site-determined expiration term.

The first and second formats of the PASSWOR command can be entered either from batch or interactive jobs. These formats cannot be used if the password randomization feature is used.

The use of the third format of the PASSWOR command is dependent on whether or not your site uses the password randomization feature.

For sites that do not use password randomization, the third format of the PASSWOR command can be used for added security and can be entered from either an interactive or batch job. When entered from an interactive job, the system prompts you to enter the password type, the old password, and the new password. If you are authorized to set an expiration date, the system also prompts you for an expiration date or term. When entered from a batch job, the third format of the PASSWOR command can be used only to change the batch password. The system reads the parameters from a record in file INPUT. This record must be a single line of the format:

oldpassword,newpassword

or

oldpassword,newpassword,expiredate

or

oldpassword,newpassword,expireterm

For sites using password randomization, the third format is the only valid format, and it must be entered from an interactive job.

PAUSE COMMAND

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

Format:

PAUSE,jsn.

<u>Parameter</u>	<u>Description</u>
jsn	Specifies the job sequence name (JSN) of the job whose pause bit you want to set.

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).

If you want to determine whether a PAUSE or GO command has been entered, write a program to set bit 13 of word RA and then status it. The system unconditionally clears bit 13 of word RA whenever the PAUSE/GO command sets/clears the pause flag (bit 12 of word RA). Hence, when the system clears it, you can check the pause flag. CPU programs that check the pause flag on NOS 1 continue to work properly on NOS 2, but these programs cannot determine when a PAUSE or GO command has been entered.

PROTECT COMMAND

The PROTECT command is used to activate or deactivate preservation of the extended memory assigned to your job field between job steps. 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).

NOTE

The PROTECT command inhibits the release of extended memory by the system between job steps. However, if the LOADER is called to load an absolute program or program overlay, PRESET must be defined as NONE with an LDSET command to the LOADER or with the LOADER build directive PRESET=NONE at the time that the system is built. If PRESET=NONE is not defined, then extended memory is not preserved across job steps regardless of the status of the PROTECT command.

Format:

PROTECT,EC=state.

or

PROTECT,state.

<u>Parameter</u>	<u>Description</u>
EC=state	Activates or deactivates extended memory preservation. You can specify two values for state:
<u>state</u>	<u>Description</u>
ON	Activates the preservation of memory.
OFF	Deactivates the preservation of memory. OFF is the default.

QGET COMMAND

The QGET command assigns a queued file to your job. You must specify the job sequence name (JSN) or the user job name (UJN) associated with the queued file. Unless explicitly routed elsewhere, the file remains with your current job. If a local file already exists with the same name as the queued file, the QGET operation is aborted.

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.
<u>q</u>	<u>Description</u>
PR	Print
PU	Punch
PL	Plot
WT	Wait
IN	Input
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 system error occurs.

QUEUE7 COMMAND

The QUEUE7 command allows you to examine the system batch queues in linked SCOPE 2 systems.

The QUEUE7 command has two formats:

QUEUE7, ID=pid, L=lfm, P1, P2, P3, P4.

or

QUEUE7, help.

<u>Parameter</u>	<u>Description</u>
ID=pid	Specifies the physical identifier (PID) of the SCOPE 2 mainframe to be searched. If no PID is specified, all mainframes are searched. This parameter may be specified with the parameter name (ID) and equal sign omitted.
L=lfm	Specifies the name of the file to receive the QUEUE7 output. The default is OUTPUT.
Pi	Specifies the SCOPE 2 queues (input, output, execute, or punch) to be displayed. Valid options for pi are listed below. From one to four options can be specified. The default is E.

<u>Pi</u>	<u>Description</u>
A	Lists jobs found in all queues (E, I, O, and P).
E	Execute queue; lists the SCOPE 2 R display. Figure 7-1 provides an example of an execute queue listing.
I	Input queue; lists the SCOPE 2 R display for input. Figure 7-2 provides an example of an input queue listing.
O	Output queue; lists the SCOPE 2 H display for output. Figure 7-3 provides an example of an output queue listing.
P	Punch queue; lists the SCOPE 2 H display for punch output. Punch queue listings are the same as output queue listings, as shown in figure 7-3.

<u>Parameter</u>	<u>Description</u>
help	Displays the command format or the physical identifiers for all available SCOPE 2 mainframes. The following are valid entries for the help parameter.

<u>help</u>	<u>Description</u>
ID	Displays physical identifiers for all SCOPE 2 mainframes in the network.
SYNTAX	Displays the QUEUE7 command format.
or	
S	

```

/queue7,e
JOBS      6/  52/  70 R(Z),1.

ORD   ACTION  JOBNAME PRTY  JOB    TIME  TIME  LCM  JOB  BG
      ID      FIRST ACTION MESSAGE/LAST DAYFILE MESSAGE
      ID
      3 M06/**      JOBXXXZ 4000 M06/**      0    7    26  PAUS
RESPONSE           PAUSE. DO NOT GIVE A GO. TESTING ON 606

      4 M06/**      JOBXXXZ 4000 M06/**      0    7    26  PAUS
RESPONSE           PAUSE. DO NOT GIVE A GO. TESTING ON 606

      5 M06/**      JOBXXXZ 4000 M06/**      0    7    26  PAUS
RESPONSE           PAUSE. DO NOT GIVE A GO. TESTING ON 606

QUEUE7 COMPLETE

```

Figure 7-1. QUEUE7 Execute Queue Sample Listing

Column headers for the QUEUE7 execute queue listing have the following meanings:

<u>Header</u>	<u>Description</u>
ORD	The job ordinal.
ACTION ID	The station/terminal identifier of the terminal from which the job expects a response; present only when the job requires a response.
JOBNAME	Job name of the executing job.
PRTY	Priority of the job; must be in the range of 0 to 7777g.
JOB ORIGIN	Station/terminal identifier of the terminal from which the job was submitted.
TIME USED	CPU seconds used by the job.
TIME LEFT	CPU seconds remaining for execution.
LCM IN USE	Number of LCM buffers in use by the job.
JOB STAT	Current processing state of the job.
BG	Indicates that the job will be scheduled as a background priority job.

FIRST ACTION MESSAGE/
LAST DAYFILE MESSAGE

If there are action messages for the job, the message is the first action message; otherwise, the message is the last dayfile message.

ACTION Indicates that the above message requests an operator action such as a tape mount.

CFO RESP Indicates that the above message requests the operator to respond with a CFO type entry.

RESPONSE Indicates that the above message requests the operator to respond with a command such as GO or DROP.

```
/queue7,i
SFTS 10/ 172/ 172 H(Z),I,103.

ORD  FILENAME  FILE  PRTY  DEVICE-REQUIREMENTS  TIME  DEPEND  BG
      ORIGIN      MT  NT  YD  YL  SM  SN  SP  LIMIT

107Z  JOBXXXZ  M06/** 1002  0  0  0  0                10    0
110Z  JOBXXXZ  M06/** 1002  0  0  0  0                10    0

STATION  M06  M02  MFF  M05  MSD  MFU  UKWN  TOT
INPUT    2    0    0    0    0    0    0    2
OUT PUT  1    0    0    0    0    0    0    1
PUNCH    0    0    0    0    0    0    0    0

QUEUE7 COMPLETE
```

Figure 7-2. QUEUE7 Input Queue Sample Listing

Column headers for the QUEUE7 input queue listing have the following meanings:

- ORD Input queue suffixed with the last letter of the SCOPE 2 physical identifier.
- FILENAME File name of the input job as specified on the job card.
- FILE ORIGIN Identifier of the station/terminal from which the job was submitted.
- PRTY File priority; must be in the range from 0 to 7777₈.
- DEVICE-
REQUIREMENTS The number of magnetic tape or disk devices assigned to the job.
- TIME LIMIT Job time limit as specified on the job card.
- DEPEND A two-character dependency identifier as specified on the job card.
- BG Indicates that the job will be scheduled as a background priority job.

```

/queue7,o
SFTS 10/ 172/ 172 H(Z),0,103.

ORD  FILENAME      FILE  PRTY  DISP  FORM  FILE  FILE IN
      DESTIN          CODE  CODE  SIZE  TRANSFER

111Z JOBXXXZ  M06/** 4003  PR          1  N

STATION  M06  M02  MFF  M05  MSD  MFU  UKWN  TOT
INPUT    2    0    0    0    0    0    0    2
OUT PUT  1    0    0    0    0    0    0    1
PUNCH    0    0    0    0    0    0    0    0

QUEUE7 COMPLETE

```

Figure 7-3. QUEUE7 Output Queue Sample Listing

Column headers for the QUEUE7 output queue listing have the following meanings:

ORD	Output queue ordinal suffixed with the last letter of the SCOPE 2 physical identifier.
FILENAME	File name of the output file as specified on the job card.
FILE DESTIN	Station/terminal identifier of the terminal to which the output file is to be routed.
PRTY	File priority. Must be in the range from 0 to 7777 ₈ .
DISP CODE	File disposition code.
FORM CODE	Forms code of the file.
FILE SIZE	File length in PRUs.
FILE IN TRANSFER	Indicates whether or not the file is currently being transferred to a station. Possible values are Y for yes, and N for no.

REDO COMMAND

The REDO command allows you to modify and reenter a previously entered command without retyping the entire command.

Format:

REDO,string/GO.

or

R,string/GO.

<u>Parameter</u>	<u>Description</u>
string	Specifies the initial characters of the command to be modified and reentered. string can be up to 10 characters long. If string is specified, the system searches for the most recently entered command beginning with that string. If string contains one or more leading blanks or asterisks, the command to be located may have leading blanks. If string is omitted, the system reenters the command immediately preceding the REDO command.
GO	Reenters the specified command without modification. Display of the OLD:, MOD:, and NEW: prompts is suppressed.

NOTE

The GO option is forced on noninteractive jobs.

When you enter the REDO command, the system prints the characters OLD:, followed by the command to be modified, with leading blanks removed. On the next line, the system prints the characters MOD: as a prompt for you to enter modifications to the line. You enter modifications using the modification symbols described below. When you finish your modifications and enter a carriage return, the system removes leading blanks from the modified line, prints the characters NEW:, followed by the modified line, then executes the command. Following is an example of a CATLIST command that is altered to specify a different file name:

```
/REDO
OLD: CATLIST,LO=F,FN=BTK
MOD:          fac
NEW: CATLIST,LO=F,FN=FAC
```

In this example, the user spaced over to the last three characters in the line and typed in a new file name.

The REDO command modification symbols are as follows:

<u>Symbol</u>	<u>Effect</u>
space	Leaves the character unchanged.
#	Deletes the character and shifts the remainder of the line one character to the left.
&	Replaces the character with a space.
^	Inserts a character string before the character marked by the ^. The inserted string is terminated by a # or a carriage return. If a ^ is followed immediately by a carriage return, the line is redisplayed as modified up to that point, and further modifications are allowed.
!	Deletes the remainder of the line.
other	Any other character entered replaces the original character.

Example:

```
/REDO
OLD: RECLAIM/DUMP,TN=MYVSN,PF=*/ALPHA,GAMMA=BETA
MOD:   ^,Z.# DF   FIL^DT=MS,
NEW: RECLAIM,Z./DUMP,DF=MYFIL,DT=MS,PF=*/ALPHA,GAMMA=BETA
```

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 removable auxiliary pack concurrently. It prevents deadlocks with other jobs that may need the same resources. You do not need to use a RESOURC command if you have only one multivolume tape file or multipack disk file because the system treats each as only one resource unit. Nonremovable auxiliary packs should not be specified on the RESOURC command. (Nonremovable packs are always available to all users.)

NOTE

The RESOURC command should precede the first tape or pack request in a job.

Format:

RESOURC,rt₁=u₁/level₁,rt₂=u₂/level₂,...,rt_n=u_n/level_n.

<u>Parameter</u>	<u>Description</u>
rt _i	Resource type:
	LO 200-cpi, 7-track magnetic tape unit
	HI 556-cpi, 7-track magnetic tape unit
	HY 800-cpi, 7-track magnetic tape unit
	HD 800-cpi, 9-track magnetic tape unit
	PE 1600-cpi, 9-track magnetic tape unit
	GE 6250-cpi, 9-track magnetic tape unit
	MT† Seven-track magnetic tape unit
	NT† Nine-track magnetic tape unit (800/1600-cpi)
	DBm 885-42 Disk Storage Subsystem (full-track; 1<m<3)
	DCm 895 Disk Storage Subsystem (full-track; 1<m<2)
	DDm 834 Disk Storage Subsystem (full-track; 1<m<8)
	DFm 887 Disk Storage Subsystem (full-track; 1<m<3; 4K sector)
	DGm 836 Disk Storage Subsystem (full-track; 1<m<3)
	DHm 887 Disk Storage Subsystem (full-track; 1<m<2; 16K sector)
	DI _m 844-21 Disk Storage Subsystem (1<m<8)
	DJ _m 844-41/44 Disk Storage Subsystem (1<m<8)
	DK _m 844-21 Disk Storage Subsystem (full-track; 1<m<8)
	DL _m 844-41/44 Disk Storage Subsystem (full-track; 1<m<8)
	DM _m 885-11/12 Disk Storage Subsystem (half-track; 1<m<3)
	DN 9853 Disk Storage Subsystem (full-track; 2K sector)
	DQ _m 885-11/12 Disk Storage Subsystem (full-track; 1<m<3)
	DV 819 Disk Storage Subsystem (single-density)
	DW 819 Disk Storage Subsystem (double-density)

The value m denotes the number of units (spindles) required for a particular auxiliary disk file that you access in your job. In this case, the rt_i value must correspond to the R=r specification on the permanent file command that references the disk pack (refer to section 10).

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.

level Specifies the security access level at which you want to access magnetic tape units. The parameter is ignored for auxiliary disk packs. Unless changed by your site, level can be one of the following names:

LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7

The default is the current job access level.

On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information) and for the device to which you are assigning the file.

On an unsecured system, the parameter has no effect. The system does, however, check for valid parameter specifications.

†Retained for compatibility with NOS 1.2. Refer to restrictions described under Tape Units in this section.

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 7-4.)

†For jobs that use only one tape or pack at a time and do not contain a RESOURC command, the system checks validation limits when the request is made.

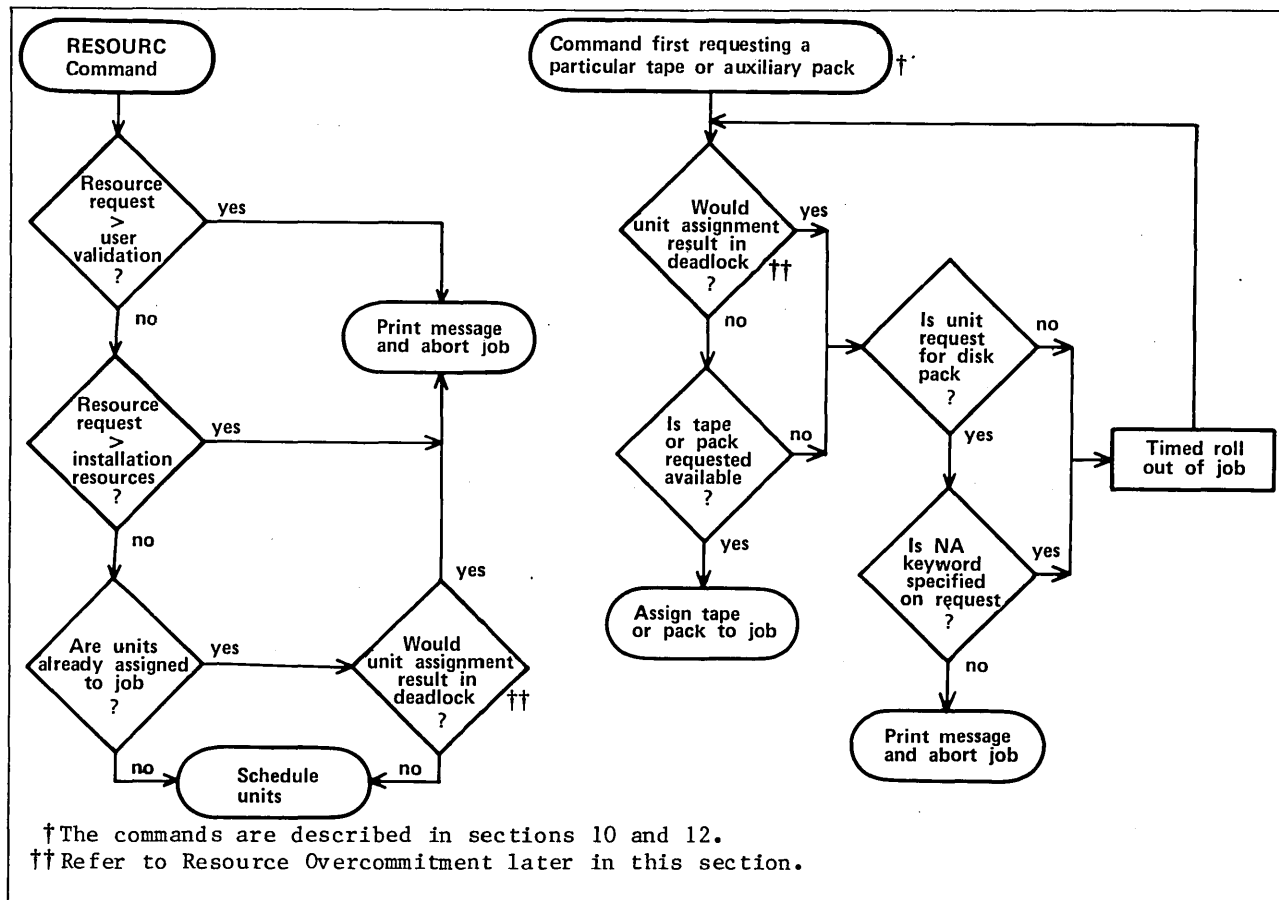


Figure 7-4. Resource Commitment Processing (Simplified)

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.

[†]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.

NOTE

If a job already has a tape or pack assigned when the RESOURC command is issued and the system determines that the assignment would cause a deadlock, the job aborts and the system issues a RESOURCE DEMAND ERROR message.

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

The system supports nine-track magnetic tapes in three different densities. In system commands, you use a 2-character density resource identifier to specify the correct density. The identifiers are as follows:

<u>Identifier</u>	<u>Density</u>
HD	800 cpi
PE	1600 cpi
GE	6250 cpi

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).

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 caused by a potential 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 1600-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 1600-cpi, nine-track tape unit.

```
SAMSJOB,CM50000,T40.  
USER,SJGREEN,WGT.  
CHARGE,*.  
RESOURC,PE=2.  
LABEL,X,D=PE,VSN=TAPE1.  
LABEL,Y,D=PE,VSN=TAPE2.  
RETURN,X,Y.  
RESOURC,DI1=2,PE=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.

Parameter

Description

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 100g.

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 1000g.

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.

<u>Parameter</u>	<u>Description</u>
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.

Typically, a job scheduler delay interval is one second but can be site-determined.

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 (set during CHARGE or USER command processing) is the smaller of the user name and the account validation limits. For interactive jobs, the default limit is 64 SRUs.

Format:

SETASL,s.

<u>Parameter</u>	<u>Description</u>
s	<p>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.</p> <p>If s is an asterisk (*), or has a value of 262136 (777770B) or greater, or has a value greater than your validation limit, the account block SRU limit is then set to your validation limit. Also, if s is less than your current job step SRU limit (refer to the SETJSL command in this section for a description of the job step SRU limit), both the account block and job step SRU limits are set to s.</p> <p>A dayfile message is then issued indicating what the new values are for the account block. SRU limit (ASL) and job step SRU limit (JSL).</p>

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 except for RA+2.

Format:

SETCORE,p.

or

SETCORE,-p.

<u>Parameter</u>	<u>Description</u>
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 RA+2) is set to p. If p is omitted, the system assumes p=0.

Calling the CYBER loader destroys the effect of a SETCORE command since the loader uses memory as a work area.

To preset memory within a load sequence, use a LDSET,PRESET command as described in the CYBER Loader Reference Manual.

SETJAL COMMAND

The SETJAL command changes the security access level of the current job. Initially, your job has the lowest possible access level that is valid for your job (ENQUIRE,B returns this information). Refer to Security Features in Section 3 for a detailed discussion of access levels.

Format:

SETJAL,AL=level.

<u>Parameter</u>	<u>Description</u>
AL=level	Specifies the security access level you want assigned to the job. Unless changed by your site, level can be one of the following names: LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7

On a secured system, the access level you specify must be valid for the job. In addition, to lower the access level of your job, you must be authorized to do so (refer to the LIMITS command).

On an unsecured system, the command has no effect. The system does, however, check for valid parameter specifications.

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 (set during CHARGE or USER command processing) 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 as described in the following paragraphs.

Format:

SETJSL,s.

<u>Parameter</u>	<u>Description</u>
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. The value s must be greater than 0. The asterisk (*), values greater than 262136 (777770B), and values greater than your validated limit set the job step SRU limit at the validation limit. If s is greater than your current account block SRU limit (refer to the SETASL command in this section for a description of the account block SRU limit), both the job step and account block SRU are set to s. Also, if s is greater than your validation limit, both SRU limits are set to the validation limit. A dayfile message is then issued indicating what the new values are for the job step SRU limit (JSL) and the account block SRU limit (ASL).

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 STOP TO END JOB STEP.
TYPED AHEAD INPUT MUST BE REENTERED.
```

is issued. You can enter one of the following:

<u>Entry</u>	<u>Description</u>
S	Increases the SRU limit by 64 units. Job execution continues. The 64-unit increment can be changed by your site.
S,nnnn	Increases the SRU limit by nnnn 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.
STOP	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.

When a limit occurs, any typed ahead input will be discarded since it would be processed as a response to the limit message.

SETPR COMMAND

The SETPR command decreases the CPU priority of a job. A user job cannot increase its priority.

SETPR,p.

<u>Parameter</u>	<u>Description</u>
p	Specifies the priority, $1 \leq p \leq 70$.

Upon job initiation, a job is assigned the maximum priority allowed for its service class. (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). When you do not specify a time limit, the system sets the limit at 64 CPU seconds.† Interactive jobs can increment their job step time limit to complete job step execution (refer to section 8).

Format:

SETTL,t.

<u>Parameter</u>	<u>Description</u>
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. The asterisk (*), values greater than 32760 (77770B), and values greater than your validated time limit set the job step time limit at your validated time limit. A dayfile message is then issued indicating what the new value is for the 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.

† Your site can change this default.

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

```
*TIME LIMIT*
ENTER T TO CONTINUE OR STOP TO END JOB STEP.
TYPED AHEAD INPUT MUST BE REENTERED.
```

is issued 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. The 64 CPU second increment can be changed by your site.
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).
STOP	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.

When a limit occurs, any typed ahead input will be discarded since it would be processed as a response to the limit message.

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 STOP TO END JOB STEP.
TYPED AHEAD INPUT MUST BE REENTERED.
```

t

```
SRU      66.835 UNTS.
```

```
RUN COMPLETE.
```

SHELL COMMAND

The SHELL command enables or disables shell program processing of system commands.

A shell program is a user-written program (usually in COMPASS) which redefines the set of commands that can be entered by a user. A shell program can:

- Limit a user to a subset of NOS commands; the shell program can define which NOS commands the user can or cannot enter.
- Redefine the effects of a NOS command. A shell program redefines a particular command by defining an alternate command or series of commands to be executed whenever the user enters the original command. For example, you could redefine a user's ATTACH command to execute both an ATTACH and a LIBRARY command for the specified file, thus making the file a global library.
- Define user-written commands that are not a part of the standard NOS command set.

In effect, a shell program creates a user-defined subsystem that can be imposed on application users by user name. To enable a shell program for a user name, you can place the SHELL command in the user's prologue. You can also associate a shell program with a user name using the MODVAL utility. (MODVAL is described in the NOS2 Administration Handbook.)

For more information on how to write a shell program, refer to the NOS Reference Set, Volume 4, Program Interface.

Format:

SHELL.

or

SHELL,spn.

or

SHELL,spn,p1,...,p7.

<u>Parameter</u>	<u>Description</u>
spn	Specifies the shell program name to be executed.
P _i	Specifies parameters to control the loading and execution of the shell program.
	<u>P_i</u>
	<u>Description</u>
	E Clears controls if there are errors during the loading of the shell program. If you do not specify E and such errors occur, your job is aborted with no EXIT processing.
	A Clears controls if the shell program aborts.
	L Allows local file loading of the shell program. When loading the program from a local file, the program must be stored in ABS-type records. (Refer to section 15 for a description of record types.)

- G Allows global library load of the shell program.
- S Allows system library load of the shell program.
- T Specifies that the system monitor commands directly from the terminal.
- B Specifies that the system monitor commands outside procedures.
- C Specifies that the system monitor commands inside procedures.

The first format disables shell control.

The second format, specifying only the shell program name, sets the default for p_i as A, S, and B.

The last format specifies the extent of shell program control. To get your shell program loaded, you must specify at least one of the parameters L, G, or S. To enable any type of command monitoring, you must specify at least one of the parameters T, B, or C.

Example:

```

/shell,rmsHELL
SHELL,RMSHELL.
/rewind,*
  9 FILES PROCESSED.
/ascii
UNAUTHORIZED COMMAND.

```

In this example, RMSHELL is a shell program that allows execution of REWIND but not ASCII. The SHELL command causes the system to perform a system library load of RMSHELL, to enable shell control, to clear controls if the shell program aborts, and to monitor commands outside procedures.

The system passes REWIND,* to RMSHELL for examination. Since it is part of the subset of allowed system commands, RMSHELL passes control to the system to process that command exactly as it is. However, when the system passes the ASCII command to RMSHELL, an UNAUTHORIZED COMMAND message is issued, and the command is not passed back to the system.

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 first record of the file must be in 6-bit display code. 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). The dayfile is not written to file OUTPUT. 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 (including the dayfile) 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	Prefix 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 a prefix character is recognized as a reformatting directive. The prefix character is specified on the SUBMIT command (/ by default). Throughout this description, the letter c, preceding a directive, denotes the prefix 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 TRANS and NOTRANS directives).

The system does not process reformatting directives unless the first line of the submit file contains the cJOB directive. Other directives can follow the cJOB directive (that is, they can precede the job and/or USER commands), but the first two commands following the cJOB directive must be a Job command and a USER command, respectively. The USER command can be inserted in the form of a cUSER directive. For example, the following sequences of lines are both valid:

```
ln1      cJOB
ln2      ujn,parameters.
ln3      USER,username,password. (or cUSER)
```

or

```
ln1      cJOB
ln2      cNOSEQ
ln3      ujn,parameters.
ln4      USER,username,password. (or cUSER)
```

ln₁, ln₂, ln₃, and ln₄ 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 cSEQ and cNOSEQ 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 specifying the user name and family of the submitting job. If you specify this directive when you are routing a job to a remote mainframe (ST=lid parameter on the Job command), the destination mainframe must be a NOS mainframe, the same user name and family must be valid on the destination mainframe, and your batch passwords must be identical on both mainframes. You can use this directive only to create the first USER command in a job.
cCHARGE	Inserts a CHARGE command that is identical to the CHARGE command currently in effect for the submitting job. If no CHARGE command is in effect for the submitting job, this directive has no effect on the SUBMIT file.

<u>Directive</u>	<u>Description</u>
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 may be preceded by line numbers and requests that they be removed (default value).
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 that the system not discard (currently existing) succeeding internal end-of-record and end-of-file marks.
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. <ol style="list-style-type: none"> 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.

Directive

Description

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.

cNOTRANS

Reverses the effect of the cTRANS directive and informs the system that the submit file is to be examined on a line-by-line basis. All directives encountered in the submit file while the cNOTRANS directive is in effect are processed. This directive is initially selected by default and remains in effect until a cTRANS directive is encountered in the submit file.

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.

cREAD, lfn

Requests that the system read the contents of the specified file, lfn, and insert that file in place of the cREAD directive in the submit file, during reformatting. Reading terminates when an EOF or EOI is encountered on lfn. If the file to be read is not currently 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 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.

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.

Directive

Description

Each line of that file is also checked for a reformatting directive, unless transparent submit mode is in effect when file lfn is read. 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.

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.

cREWIND,lfn

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.

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 prefix character is to be changed from c₁ (current prefix character) to c₂ (new prefix character). The new prefix character code is used to recognize all subsequent reformatting directives until further changed.

Input lines must not exceed 320 6-bit characters. If a line exceeds 320 characters, the results are unpredictable.

If the submitted job contains an invalid USER command, the job entering the SUBMIT command is aborted. This constitutes a security conflict (refer to the USER command later in this section). The security count for the user name that entered the SUBMIT command is decremented. If this results in a security count of zero, the job is immediately terminated with no EXIT processing (you are logged out if the job is interactive). 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.

SWITCH COMMAND

The SWITCH command sets the sense switches for reference by one of your programs.

Format:

SWITCH,switch₁,switch₂,...,switch_n,jsn.

<u>Parameter</u>	<u>Description</u>
switch _i	Specifies the switches to be set. You must denote each 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.
jsn	Specifies the job sequence name (JSN) of the job whose sense switches will be changed. jsn is an optional parameter, its default is the current job. The system recognizes the jsn parameter by its alphabetic characters; therefore, jsn can appear anywhere in the parameter list and can be omitted without leaving a blank parameter field.

UPROC COMMAND

The UPROC command specifies a user prologue, a program the system executes each time you start a job.

Format:

UPROC,FN=pfile.

or

UPROC,pfile.

<u>Parameter</u>	<u>Description</u>
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. If you omit the FN=pfile parameter or give pfile the value of zero, the system will no longer execute a user prologue.

The system executes your user prologue only after it processes your user validation and charge information but before it initiates job recovery processing. Depending on the length and nature of your user prologue, you may want to initiate recovery processing before your user prologue ends. For example, if your user prologue aborts, the system does not initiate recovery processing. To do this, place the RECOVER command within your user prologue (refer to section 8 for the RECOVER command and section 4 for writing 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.

<u>Parameter</u>	<u>Description</u>
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 72-2x, 73-2x, 74-2x, 174, 720-2, 730-2, 6500, 6700, 830-2, 865-2, or 875-2. 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 an authorized 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,UN=username,PW=password,FM=familyname.

or

USER,username,password,familyname.

<u>Parameter</u>	<u>Description</u>
username	Specifies a one- to seven-character alphanumeric user name.
password	Specifies an alphanumeric password. Its maximum length is seven characters; its minimum length is defined by the installation.
familyname	Specifies an 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. If this is not the first USER command of the job or it is the first USER command from a job submitted from another job, you must not specify a family name different from the current one unless you are so authorized (refer to the LIMITS command).

†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 the system is a secured system, secondary user commands cannot be enabled. Only if secondary USER commands are enabled, may you 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).

On a secured system or on an unsecured system with secondary USER commands enabled, an invalid USER command is processed as a security conflict. Your job aborts and your security count is decremented. If this results in your security count being set to zero, batch jobs will terminate without EXIT processing. At a terminal, you will be immediately logged out with no dayfile messages issued to the terminal. If the security count is not zero, the job step is aborted with appropriate messages to the terminal and the dayfile and with EXIT processing enabled (unless explicitly disabled).

On an unsecured system with secondary USER commands disabled, any secondary USER command will be invalid and aborts the job. This is not a security conflict and the security count is not decremented.

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.

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 JD SMITH with password SPWD could enter either of the following commands.

USER,JD SMITH,SPWD.

USER,JD SMITH,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 JD SMITH could enter either of the USER commands as before because the default family name would still be valid.

COMMANDS FOR INTERACTIVE JOBS

8

There are many system commands that you can use only in interactive jobs or that were designed primarily for interactive use. Unless specified otherwise, these commands cannot be used in procedures. This section describes these commands under the following headings:

- Terminal Control Commands

ASCII	LINE
AUTO	NORMAL
BRIEF	SCREEN
CSET	TDU
EFFECT	TIMEOUT
KEY	TRMDEF

- Subsystem Selection Commands

ACCESS	FORTRAN
BASIC	FTNTS
BATCH	NULL
EXECUTE	

- Interactive Status Commands

ctD
ctE
ctS

- Job Processing Commands

APPSW	LOGIN
BYE	LOGOUT
DIAL	NOSORT
EXPLAIN	RECOVER
GOODBYE	RUN
HELLO	Secure Login
HELLO7	SHOW
HELP	TEXT
HELPME	WHATJSN
LIB	X
LIST	XMODEM

- 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.

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-bit display code. On an ASCII terminal, 6/12-bit 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, you are 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.

The system normally 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 full-ASCII or 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.

You can include this command in a procedure.

AUTO COMMAND

The AUTO command automatically generates five-digit line numbers.

Format:

AUTO,nnnnn,iiii

<u>Parameter</u>	<u>Description</u>
nnnnn	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.
```

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,mode

<u>Parameter</u>	<u>Description</u>
mode	Specifies terminal character set mode.
ASCII	Selects ASCII mode (ASCII 128-character set).
NORMAL	Selects normal mode (ASCII graphic 63/64-character set).

The CSET command may also appear in a procedure file.

NOTE

The CSET command sets only the terminal character set. If you specify the NORMAL parameter, you do not affect auto or brief mode.

EFFECT COMMAND

The EFFECT command enables or disables the output format effectors supplied by IAF. With the IAF effectors disabled, you control terminal output by supplying an appropriate effector as the first character of each output line; the system interprets the first character of each line as an effector and inhibits printing of the character.

Format effectors control carriage positioning on output operations. The effectors for specific terminal classes are described in the Network Access Method Version 1/Communications Control Program Version 3 Terminal Interfaces Reference Manual. This command is valid for all terminals connected through NAM.

NOTE

Control bytes 0001, 0002, 0003, 0005, 0010, 0013, 0014, 0015, and 0016 should not be used with IAF effectors disabled.

The ? prompt, normally supplied by the system, to request input is not displayed when format effectors are user-supplied.

Format:

EFFECT,mode.

<u>Parameter</u>	<u>Description</u>						
mode	Specifies whether or not IAF automatically supplies format effectors. Values that can be entered for mode are as follows:						
	<table><thead><tr><th><u>Value</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>SYSTEM</td><td>Format effectors are supplied by IAF. This is the default value.</td></tr><tr><td>USER</td><td>Format effectors are user-supplied as the first character of each output line.</td></tr></tbody></table>	<u>Value</u>	<u>Description</u>	SYSTEM	Format effectors are supplied by IAF. This is the default value.	USER	Format effectors are user-supplied as the first character of each output line.
<u>Value</u>	<u>Description</u>						
SYSTEM	Format effectors are supplied by IAF. This is the default value.						
USER	Format effectors are user-supplied as the first character of each output line.						
	When format effectors are user-supplied (i.e., EFFECT,USER is specified), all format effectors are supported except those that specify carriage positioning after output. Specifically, the slant (/) and period (.) effectors are not supported.						

You can include this command in a procedure.

KEY COMMAND

The KEY command maintains key definition files which define the function keys for a CDC 721 terminal. KEY is used to create, edit, display, or load a definition file.

The KEY command is valid only for a CDC 721 terminal operating in screen mode. To use this command, select the appropriate option from those listed below and enter the command with that option. The KEY command provides you with interactive help to assist you in creating, editing, or loading the definition file.

Format:

KEY,option,filename.

or

KEY?

<u>Parameter</u>	<u>Description</u>																
option	Any of the following options.																
	<table border="0"> <thead> <tr> <th><u>Option</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>HELP</td> <td>Gives a brief description of each option available. The filename parameter is ignored.</td> </tr> <tr> <td>CHECK</td> <td>Checks to see if function key definitions are already loaded into the terminal. If not, the definitions in file filename are loaded.</td> </tr> <tr> <td>DEFAULT</td> <td>Sets key definitions to the NOS system default keys. The filename parameter is ignored. The default keys are: HELP key = HELP EDIT key = FSE STOP key = CTRL-T followed by carriage return</td> </tr> <tr> <td>DISPLAY</td> <td>Displays the key labels for function keys F1 through F16.</td> </tr> <tr> <td>EDIT</td> <td>Initiates creation or editing of a key definition file. KEY first searches for a local file by the specified name, then for an indirect access permanent file. If filename does not exist, it is created. The edited file is saved or replaced at completion of editing.</td> </tr> <tr> <td>LOAD</td> <td>Loads key definitions into the terminal from the specified file.</td> </tr> <tr> <td>PRINT</td> <td>Creates a local file called KEYPRNT on which the key definitions are listed from the specified filename.</td> </tr> </tbody> </table>	<u>Option</u>	<u>Description</u>	HELP	Gives a brief description of each option available. The filename parameter is ignored.	CHECK	Checks to see if function key definitions are already loaded into the terminal. If not, the definitions in file filename are loaded.	DEFAULT	Sets key definitions to the NOS system default keys. The filename parameter is ignored. The default keys are: HELP key = HELP EDIT key = FSE STOP key = CTRL-T followed by carriage return	DISPLAY	Displays the key labels for function keys F1 through F16.	EDIT	Initiates creation or editing of a key definition file. KEY first searches for a local file by the specified name, then for an indirect access permanent file. If filename does not exist, it is created. The edited file is saved or replaced at completion of editing.	LOAD	Loads key definitions into the terminal from the specified file.	PRINT	Creates a local file called KEYPRNT on which the key definitions are listed from the specified filename.
<u>Option</u>	<u>Description</u>																
HELP	Gives a brief description of each option available. The filename parameter is ignored.																
CHECK	Checks to see if function key definitions are already loaded into the terminal. If not, the definitions in file filename are loaded.																
DEFAULT	Sets key definitions to the NOS system default keys. The filename parameter is ignored. The default keys are: HELP key = HELP EDIT key = FSE STOP key = CTRL-T followed by carriage return																
DISPLAY	Displays the key labels for function keys F1 through F16.																
EDIT	Initiates creation or editing of a key definition file. KEY first searches for a local file by the specified name, then for an indirect access permanent file. If filename does not exist, it is created. The edited file is saved or replaced at completion of editing.																
LOAD	Loads key definitions into the terminal from the specified file.																
PRINT	Creates a local file called KEYPRNT on which the key definitions are listed from the specified filename.																
filename	The name of the key definition file. Default is KEYDEFS.																
?	Lists the options and prompts you for an option and filename.																

The KEY command may appear in a procedure file.

LINE COMMAND

The LINE command specifies that your terminal is to be operated in line (or scrolling) mode, as opposed to screen mode. Commands and features affected by the LINE command include CATLIST, FSE, screen formatting, and the display of NOS procedure parameters.

The LINE command also identifies the terminal model to the system and initiates a search for the terminal definition file for that model (unless the CDC 721 terminal was specified; characteristics of this terminal are preloaded into the system so that no definition file is required). The system uses the following search order in attempting to find the definition file:

- A local file called TERMLIB (created using the TDU utility).
- An indirect access permanent file called TERMLIB under your user name.
- An indirect access permanent file called TERMLIB under user name LIBRARY (UN=LIBRARY). Your site may maintain a system file containing common terminal definitions.

If TERMLIB is found under your user name or under user name LIBRARY, the system copies the terminal definition to a local file called ZZZZTRM for later use by NOS full screen display features. The terminal files listed below for the TM parameter are released with the system and are always available once you have identified your terminal in a LINE or SCREEN command.

Format:

LINE, TM=model.

or

LINE,model.

<u>Parameter</u>	<u>Description</u>
TM=model	Specifies the terminal model. The value specified for model is the terminal mnemonic defined in the terminal definition file or the terminal mnemonic for one of the following preloaded terminal definitions:

<u>model</u>	<u>Description</u>
721	CDC 721 terminal.
721V3	CDC 721 terminal version 3.
722	CDC 722 terminal.
72230	CDC 722-30 terminal.
VT100	DEC VT100 terminal.†
Z19	Zenith Z19 or Z29 terminal or Heathkit H19 terminal.††
ADM3A	Lear Siegler ADM3A terminal.†††
ADM5	Lear Siegler ADM5 terminal.†††
TV924	TeleVideo 924 terminal.†††††
TV950	TeleVideo 950 terminal.†††††
TV955	TeleVideo 955 terminal.†††††
T4115	Tektronix 4115 terminal.††††
3270	IBM 3270 terminal.†††††
PCONxx	IBM PC using CDC CONNECT. xx is the current level of CONNECT.

If you want your terminal to have the type-ahead capability, append T to the model name you are using in the LINE command. Type-ahead allows you to press a function key two or more times in quick succession, rather than pressing it once and waiting for the system to execute it before pressing it a second time.

If this parameter is omitted, the terminal model is left unchanged.

You can include this command in a procedure.

†DEC is a registered trademark of the Digital Equipment Corporation.

††Zenith Z19 and Z29 are products of the Zenith Corporation. Heathkit H19 is a product of the Heath Corporation.

†††Lear Siegler ADM3A and ADM5 are products of the Lear Siegler Corporation.

††††Tektronix 4115 is a product of the Tektronix Corporation.

†††††IBM 3270 is a product of the IBM Corporation.

††††††TeleVideo 924, 950, and 955 are products of TeleVideo Systems, Inc.

NORMAL COMMAND

The NORMAL command reverses the effect of the ASCII; AUTO; BRIEF; CSET,ASCII; and NOSORT commands on both input and output. The system initially assumes that this command has been entered. Normal mode uses the ASCII graphic 63/64-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

SCREEN COMMAND

The SCREEN command specifies that your terminal is to be operated in screen mode, as opposed to line (or scrolling) mode. Commands and features affected by the SCREEN command include CATLIST, FSE, screen formatting, and the display of NOS procedure parameters. The SCREEN command also identifies the terminal model to the system and initiates a search for the terminal definition file for that model (unless the CDC 721 terminal was specified; characteristics of this terminal are preloaded into the system so that no definition file is required). The system uses the following search order in attempting to find the definition file:

- A local file called TERMLIB (created using the TDU utility).
- An indirect access permanent file called TERMLIB under your user name.
- An indirect access permanent file called TERMLIB under user name LIBRARY (UN=LIBRARY). Your site may maintain a system file containing common terminal definitions.

If TERMLIB is found under your user name or under user name LIBRARY, the system copies the terminal definition to a local file called ZZZZTRM for later use by NOS full screen display features. The terminal files are always available once you have identified your terminal in a SCREEN or LINE command.

Format:

SCREEN, TM=model.

or

SCREEN,model.

<u>Parameter</u>	<u>Description</u>
TM=model	Specifies the terminal model. The value specified for model is the terminal mnemonic defined in the terminal definition file or the terminal mnemonic for one of the following preloaded terminal definitions:

<u>model</u>	<u>Description</u>
721	CDC 721 terminal.
721V3	CDC 721 terminal version 3.
722	CDC 722 terminal.
72230	CDC 722-30 terminal.
VT100	DEC VT100 terminal.†
Z19	Zenith Z19 or Z29 terminal or Heathkit H19 terminal.††
ADM3A	Lear Siegler ADM3A terminal.†††
ADM5	Lear Siegler ADM5 terminal.†††
T4115	Tektronix 4115 terminal.††††
3270	IBM 3270 terminal.†††††
PCONxx	IBM PC using CDC CONNECT. xx is the current level of CONNECT.

If you want your terminal to have the type-ahead capability, append T to the model name you are using in the SCREEN command. Type-ahead allows you to press a function key two or more times in quick succession, rather than pressing it once and waiting for the system to execute it before pressing it a second time.

If this parameter is omitted, the terminal model is left unchanged.

You can include this command in a procedure.

†DEC is a registered trademark of the Digital Equipment Corporation.
 ††Zenith Z19 and Z29 are products of the Zenith Corporation. Heathkit H19 is a product of the Heath Corporation.
 †††Lear Siegler ADM3A and ADM5 are products of the Lear Siegler Corporation.
 ††††Tektronix 4115 is a product of the Tektronix Corporation.
 †††††IBM 3270 is a product of the IBM Corporation.

TDU COMMAND

The TDU command calls an interactive procedure to compile a terminal definition file and store the compiled definition (in load capsule form) in a user library. The user library can later be accessed by a SCREEN or LINE command to define terminal characteristics to the system.

The user library to receive the load capsule must be a local file. If the library file you specify does not exist as a local file, TDU creates it. If you do not specify a library file, TDU uses a local file with the name TERMLIB. If no file TERMLIB exists, TDU creates it.

Format:

TDU,I=definition,L=listing,LIB=library,REC=recname,RW=option

<u>Parameter</u>	<u>Description</u>
I=definition	Specifies the terminal definition file. The file must be in 6/12-bit display code. For information on how to create a terminal definition file, refer to the NOS Version 2 Screen Formatting Reference manual.
L=listing	Specifies the listing file. The listing file contains the list of error messages, if any. The default name is OUTPUT.
LIB=library	Specifies the library file to receive the load capsule. The default name is TERMLIB. To be used by the SCREEN and LINE commands, the library file must be a local file or an indirect access file called TERMLIB.
REC=recname	Specifies the name of the terminal definition record to be selected for processing from the terminal definition file. If an asterisk (*) is entered, all records are processed. If REC is omitted, the current record is processed with no regard to the record name.
RW=option	Specifies if the terminal definition file and listing file are to be rewound before and after processing.

<u>option</u>	<u>Description</u>
NO or N	Files are rewound.
YES or Y	Files are rewound.
If RW is omitted, the files are rewound.	

The TDU command may appear in a procedure file.

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 page length or the page-wait attribute). You can also use terminal definition commands (also called network commands) for this purpose. Typically, you would use TRMDEF in a prologue to set your terminal attributes for each login--especially if you tend to log in at the same type of terminal each time. You would use network commands to make on-the-fly changes during a given terminal session.

The network commands for a NAM/CCP network are in appendix J. The CDCNET Terminal Interface Usage manual describes network commands for a NAM/CDCNET network.

Format:

TRMDEF,L=lfn,tc₁=v₁,...,tc_n=v_n

<u>Parameter</u>	<u>Description</u>
L=lfn	Specifies an optional file on which the terminal redefinition information is written. If lfn 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 J-1 in appendix J. 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 J-1). You may not use the mnemonics AR, CH, HN, HS, MS, and TM for tc _i .

The new option, v_i, is any value in the range given for that terminal characteristic in table J-1. It can be a decimal value, a coded value with a special meaning (such as PR for printer), or a single character. You can specify multiple values for DL, EB, EL, and XL if you separate the values with slants (/), but only as specified in appendix J. 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 44g).

<u>Parameter</u>	<u>Description</u>
<u>v_i</u>	<u>Description</u>
\$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 the 8-bit ASCII code for a character (for example, 52B, which would be equivalent to an entry of \$*\$).
vvD	Decimal value of the 8-bit ASCII code for a character (for example, 42D, which would be equivalent to an entry of \$*\$).
Xvv	Hexadecimal value of the 8-bit ASCII code for a 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.

Using TRMDEF with NAM/CDCNET

The CDCNET network defines a larger set of terminal attributes than does the CCP network. This gives you a greater degree of control of your terminal and its connections. For example, associated with the page wait attribute under CCP are two attributes under CDCNET: HOLD_PAGE (HP) and HOLD_PAGE_OVER (HPO). To avail yourself of this added control, you must use the CDCNET mnemonics for the attributes when using TRMDEF. The CDCNET Access Guide describes the CDCNET mnemonics along with any restrictions for parameter specifications with TRMDEF.

If you do choose to use the CCP mnemonics for terminal attributes even though you are using a CDCNET network, the system maps the CCP attributes into the corresponding CDCNET attributes. But again, it will not be a one-to-one mapping. The CDCNET Access Guide shows how the two sets of attributes interrelate.

Examples

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).

TRMDEF,DL=C20/X03/T0 This command defines three different transparent input delimiters.

- the transmission of 20 characters
- the entry of an ETX character code
- a timeout

The first delimiter encountered terminates transparent input mode.

The operating system and the network check the validity of the parameter values you enter. The operating system does the preliminary check (syntax and general range checking) and the network does the specialized parameter checking. If you specify an invalid parameter value, the system informs you and the terminal characteristics in effect before you entered the command remain in effect. In general, the system informs you immediately of the error. The only case where you do not receive immediate notification of the error is where you have specified the L=lfm parameter and the system does not detect the error in its preliminary check. However, the network notifies you of the error as soon as you try to actualize the changes (copy file lfm to the terminal). Also, in this case, the system does not report to EXIT processing if you entered the command from 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, FTNTS, 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 FTNTS 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). You can also set the subsystem in your prologue with the SET 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
```

<u>Parameter</u>	<u>Description</u>
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, fl

<u>Parameter</u>	<u>Description</u>
fl	Specifies the initial running field length for subsequent job steps. Entering this value is equivalent to entering the RFL, fl command (refer to section 7). If fl 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

<u>Parameter</u>	<u>Description</u>
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.

In the following 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.

EXECUTE, OLD, OBJFILE

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=l fn
```

Causes the object code program to be placed on temporary file lfn.

```
SAVE,l fn/SS=EXECUTE
```

Retains file lfn 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
```

<u>Parameter</u>	<u>Description</u>
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 OLD 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

<u>Parameter</u>	<u>Description</u>
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:

ct x

<u>Parameter</u>	<u>Description</u>
ct	ct is the network control character defined for the terminal (refer to the CT command in appendix J). Do not include a space between ct and x. We show one here only for clarity.
x	x can be the letter D, E, or S.

With the exception of the Detach command (ctD), 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
- During the processing of the interruption or termination sequence

DETACH COMMAND (ctD)

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 interruption sequence and then enter the Detach command. Otherwise, ctD alone is sufficient.

When you detach a job, the following message appears at the terminal:

```
JOB DETACHED, JSN=jsn
```

All files assigned to the detached job remain assigned to it. Any commands typed ahead, before the detachment, are discarded.

Subsequent to the detachment, the system creates a new terminal session with a new job sequence name (JSN). Any commands you enter after the detachment 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:

- No more commands remain. 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 specified the OP=TJ parameter (terminate job option) 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.

You may route terminal output from a detached job to the wait queue. To do this you must enter a SETJOB command and an ASSIGN command before detaching the job. The SETJOB command should specify DC=TO. The ASSIGN command should be as follows:

```
ASSIGN,MS,OUTPUT
```

The system limits the number of jobs you can detach (refer to the LIMITS command in section 7).

IMMEDIATE JOB STATUS COMMAND (ctE)

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 ctE command:

```
JSN: AADF SYSTEM: BATCH SRU: 2.095 FILE NAME: SAMPLE2
STATUS: IDLE
```

ABBREVIATED JOB STATUS COMMAND (ctS)

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.

APPSW COMMAND

The APPSW command allows you to temporarily switch to a secondary (alternate) NAM application program. The NAM application you originally logged into is called the primary application. The secondary application that you specify on the APPSW command must be one that is supported by your site and one that you are validated to use. When you log out of the secondary application, your terminal reverts to the primary application. While in the secondary application, your current session is suspended. Upon exiting the secondary application, your session resumes processing at the point where the APPSW command was encountered.

The job issuing the APPSW command must be an interactive terminal job and must be a NAM terminal connection. The APPSW command is invalid for RDF terminals or any origin type other than interactive.

Format:

APPSW,AP=appl,Z.data

or

APPSW,appl,Z.data

<u>Parameter</u>	<u>Description</u>																								
appl	Specifies one of the following NAM applications:																								
	<table><thead><tr><th><u>appl</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>ITF</td><td>Interactive Transfer Facility</td></tr><tr><td>MCS</td><td>Message Control System</td></tr><tr><td>NETOU</td><td>Network Operator Utility</td></tr><tr><td>NJF</td><td>Network Job Entry Facility</td></tr><tr><td>PLATO</td><td>PLATO-NAM Interface</td></tr><tr><td>PSU</td><td>Printer Support Utility</td></tr><tr><td>RBF</td><td>Remote Batch Facility</td></tr><tr><td>TVF</td><td>Terminal Verification Facility</td></tr><tr><td>TLF</td><td>Tieline Facility</td></tr><tr><td>TAF</td><td>Transaction Facility</td></tr><tr><td>VEIAF</td><td>NOS/VE Interactive Facility</td></tr></tbody></table>	<u>appl</u>	<u>Description</u>	ITF	Interactive Transfer Facility	MCS	Message Control System	NETOU	Network Operator Utility	NJF	Network Job Entry Facility	PLATO	PLATO-NAM Interface	PSU	Printer Support Utility	RBF	Remote Batch Facility	TVF	Terminal Verification Facility	TLF	Tieline Facility	TAF	Transaction Facility	VEIAF	NOS/VE Interactive Facility
<u>appl</u>	<u>Description</u>																								
ITF	Interactive Transfer Facility																								
MCS	Message Control System																								
NETOU	Network Operator Utility																								
NJF	Network Job Entry Facility																								
PLATO	PLATO-NAM Interface																								
PSU	Printer Support Utility																								
RBF	Remote Batch Facility																								
TVF	Terminal Verification Facility																								
TLF	Tieline Facility																								
TAF	Transaction Facility																								
VEIAF	NOS/VE Interactive Facility																								
Z	Specifies that any characters following the command terminator are to be forwarded to the secondary application as data.																								
data	Data to be forwarded to the secondary application. data can be up to 50 characters long. Any additional characters are truncated.																								

BYE COMMAND

The BYE command ends your session with IAF and either logs you out of your host and disconnects the terminal or places you under the control of another NAM application.

If you are connected to IAF as your secondary application (refer to the APPSW command description for more information on secondary applications), entering the BYE command with a null application parameter returns your terminal connection to the primary application.

Format:

BYE,appl

<u>Parameter</u>	<u>Description</u>
appl	If IAF is your primary application, appl specifies an application to which you wish to be connected. The following are some supported applications:

<u>appl</u>	<u>Description</u>
ITF	Interactive Transfer Facility.
MCS	Message Control System.
PLATO	PLATO-NAM Interface.
RBF	Remote Batch Facility.
TAF	Transaction Facility.
TVF	Terminal Verification Facility.
VEIAF	NOS/VE Interactive Facility.

If IAF is your secondary application, you have the following options for the appl parameter:

<u>appl</u>	<u>Description</u>
omitted	If you omit the appl parameter, your terminal is logged out of IAF and is returned to the control of the primary application.
ABORT	If you specify ABORT for appl, an abort status message is returned to the primary application, and your terminal is disconnected from both the primary and secondary applications.

Any other application name will be ignored, and you will be returned to your primary application.

If IAF is your primary application and if you select an application that is available and that you are validated to use, you are automatically released from IAF control and placed under the control of the named application. If you do not select another application, you are logged out of the host 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
Characters=xxxxx     KCHS
IAF                   CONNECT TIME      hh.mm.ss.
LOGGED OUT.

```

The following describes the variable items in the logout information:

<u>Item</u>	<u>Description</u>
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.
xxxxxxxxx	A count of the total number of input and output characters read from, or written to, your terminal.

If you switch applications without disconnecting from the network, your terminal's characteristics do not change.

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.

This command can appear in a procedure or in a batch job. If used in this manner, it terminates the job.

DIAL COMMAND

The DIAL command sends a one-line message to another terminal user.

Format:

DIAL,jsn,sss

<u>Parameter</u>	<u>Description</u>
jsn	Specifies the job sequence name (JSN) of the terminal to receive the message.
sss	Specifies the one-line message.

The receiving terminal displays the message in the format:

FROM sjsn:sss

where sjsn is the JSN of the sending job and sss is the 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.

EXPLAIN COMMAND

The EXPLAIN command retrieves an online version of a CDC manual.

Format:

EXPLAIN,M=manual

or

EXPLAIN,manual

<u>Parameter</u>	<u>Description</u>
M=manual	Specifies the manual you want to retrieve. Your site determines what manuals are available to you. If you omit the parameter M=manual, the system generates a menu of the EXPLAIN features.

GOODBYE COMMAND

Same as BYE command.

Format:

GOODBYE,application

This command can appear in a procedure or a batch job. If it does, it terminates the job.

HELLO COMMAND

The HELLO command logs you out of IAF and switches you to another application or reinitiates the login sequence. Your terminal's characteristics do not change.

Format:

HELLO,application

If application is specified or if you are logged into IAF as a secondary application (in which case, the application parameter is ignored), 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.

This command can appear in a procedure or a batch job. If it does, it terminates the job.

HELLO7 COMMAND

The HELLO7 command allows you to submit a job that requires terminal input or output to a linked SCOPE 2 system and to communicate interactively with an executing SCOPE 2 job. If your SCOPE 2 job does not require interactive participation, you can route it to SCOPE 2 with the ROUTE command.

A job you submit through HELLO7 (or the ROUTE command) must be a standard SCOPE 2 job using SCOPE 2 control statements. The job initiates terminal input or output operations by specifying CNF=YES on a FILE control statement or macro or by issuing a CONNECT macro for a file.

There are two modes of operation under HELLO7: control mode and run mode. In control mode, the initial mode on entering HELLO7, you are communicating with the HELLO7 utility. In control mode you can submit a job to SCOPE 2, drop a job, or obtain status information on a job. Once you submit a job to SCOPE 2 (by entering a SUBMIT directive), you enter run mode, in which you interact directly with your job as it executes on the SCOPE 2 system. You remain in run mode until you enter a termination directive.

The following command calls HELLO7 and places the terminal in control mode.

Format:

```
HELLO7.
```

If your NOS system is attached to a single SCOPE 2 mainframe, HELLO7 normally responds with a control mode prompt indicating that it is ready to accept a directive. The control mode prompt consists of two asterisks (**). The run mode prompt, two dollar signs (\$\$), is discussed under the PROMPT directive below. Both the control mode and run mode prompts are followed by a ? prompt. The ? prompt appears at the beginning of the input line.

In a multimainframe environment, the following prompt may appear after you enter the HELLO7 command.

```
MA INFRAME=
```

You respond by entering the logical identifier (LID) of the SCOPE 2 system you want to access. -If the mainframe you request is not available, the following message appears at the terminal.

```
MAINFRAME ID NOT IN ID TABLE
```

In response to this message you may enter an alternate LID, or you may terminate HELLO7 processing by typing END.

HELLO7 Directives

Once you have received the control mode prompt (**), you can submit a job, drop a job, obtain status information, or temporarily exit from HELLO7 using the following HELLO7 directives. The termination directive, *EOF directive, *EOP directive, and *EOS directive can be entered only in run mode. All other directives can be entered only in control mode.

<u>Directive</u>	<u>Function</u>						
ub2 (termination directive)	Suspends terminal communications with the SCOPE 2 job. The terminal returns to control mode while the job continues to run. You can re-establish communications with the job (reenter run mode) later using the RESUME directive. ub2 represents the user break 2 termination sequence recognized by your terminal. The actual termination sequence varies with the terminal class; most terminal classes use) or CTRL/T in the termination sequence. Refer to Terminating Job Step in Section 16 and to the B2 terminal definition command in Appendix J for more information on the termination sequence.						
DROP,J or DR	Drops a SCOPE 2 job prior to normal termination. You can then enter a status request to determine when drop completion has occurred.						
END or E	Disconnects the terminal from HELLO7 and drops any SCOPE 2 job active at the time the END directive is issued.						
*EOF	Inserts an end-of-information designator into terminal input data.						
*EOP	Inserts an end-of-partition designator into terminal input data.						
*EOS	Inserts an end-of-section designator into terminal input data.						
EXIT	Disconnects the terminal from HELLO7 but does not drop an active job you have submitted to SCOPE 2. This allows you to leave HELLO7, execute other commands, eventually return to HELLO7, and resume monitoring the executing job.						
PROMPT,status	Activates or deactivates the automatic run mode prompt which appears at the terminal whenever the SCOPE 2 job requests input from the terminal. The run mode prompt consists of two dollar signs (\$\$). When the automatic prompt is off, terminal prompts must be supplied by the executing SCOPE 2 job. Valid entries for status are as follows.						
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>Status</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>ON</td> <td>The automatic prompt is activated (default).</td> </tr> <tr> <td>OFF</td> <td>The automatic prompt is deactivated.</td> </tr> </tbody> </table>	<u>Status</u>	<u>Description</u>	ON	The automatic prompt is activated (default).	OFF	The automatic prompt is deactivated.
<u>Status</u>	<u>Description</u>						
ON	The automatic prompt is activated (default).						
OFF	The automatic prompt is deactivated.						
RESUME or R	Reestablishes terminal communications with an executing SCOPE 2 job. This directive allows you to reenter run mode after having issued a termination directive. On reentering run mode, the following message appears at your terminal. RESUMING 7000 MODE..						

NOTE

If you have no SCOPE 2 job active when the RESUME directive is issued, the above message will be preceded by the warning message, NO 7000 JOB ACTIVE. The terminal will then wait in run mode indefinitely for connected file input/output with a SCOPE 2 job output.

Directive

Function

STATUS or S

Returns status information on the SCOPE 2 job submitted (using the SUBMIT directive). The response is of the following form.

JOBNAME=jobname USERID=id STATUS=status

where:

jobname Name of last job submitted to SCOPE 2.
id Your user ID.
status Indicates the status of the job identified by jobname. The following values can be returned for status.

<u>status</u>	<u>Description</u>
INPUT	Indicates job waiting for terminal input.
OUTPUT	Indicates job ready for terminal output.
RUN	Indicates job is currently executing.
STOP	Indicates job has aborted.

SUBMIT, lfn

Submits a local file lfn to the SCOPE 2 system for execution and places the terminal in run mode. The SUBMIT directive must be entered with no terminator. HELLO7 responds to the directive with this message.

FORWARDING JOB TO MONITOR

The above message indicates that the job is being validated for account codes. When validation is completed, the following message appears.

ENTERING 7000 MODE .. WAIT

This message indicates that the logical file name has been routed to SCOPE 2 for execution. At this point, you have entered run mode and are waiting for the job to be scheduled. When the job is ready for input, the run mode prompt (\$\$) will appear, unless you have disabled the automatic prompt using the PROMPT,OFF directive.

If the connection to the SCOPE 2 system is inoperative, the following message is returned in response to the SUBMIT directive.

7000 INTERACTIVE NOT ENABLED

Example:

The sample job below reads interactive input from a terminal and writes the input information to the output file. The sample job is contained in a local file called TESTJOB. Following the sample job is an example of an interactive session in which the sample job is submitted to SCOPE 2 through HELLO7.

Sample Job

```
H7JOB,STMFZ,jobcard parameters.
ACCOUNT,account card parameters.
FILE (INPUT,CNF=YES)
FILE (OUT PUT,CNF=YES)
WHILE,EF=0,TAG.
COPYR.
ENDW,TAG.
6/7/8/9
```

Interactive Session

```
/hello7.
MAINFRAME= ? mfz
**
? submit,testjob
FORWARDING JOB TESTJOB TO MONITOR
ENTERING 7000 MODE .. WAIT
**
? 1
1
$$
? 222222
222222
$$
? test abcdefg
TEST ABCDEFG
$$
?
**
end
```

HELP COMMAND

The HELP command generates a menu of the online help features and prompts you to make a selection.

Format:

HELP

There are four online help features:

- A list of all NOS commands, except for compiler calls.
- Help in entering a command (the HELPME feature). HELPME gives a description of the command and its parameters, prompts you for the parameters, and then executes the command.
- Access to an online version of selected CDC manuals (the EXPLAIN feature).
- A list of the NAM/CCP network commands.

HELPME COMMAND

The HELPME command gives a brief description of a command and its parameters, prompts you for parameter values, and then executes the command.

Format:

HELPME,command

<u>Parameter</u>	<u>Description</u>
command	Specifies a command. You can get a list of commands by using the HELP command or by requesting help after entering HELPME.

The HELPME command is an interactive procedure. You can get help at any time during your dialog with HELPME by entering a question mark (or pressing the HELP function key if your terminal is in screen mode). For more information about help on procedure calls, refer to Requesting Help on Procedure Calls in section 4.

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=pfn/PW=password,PN=packname,R=r,NA,ND,WB

<u>Parameter</u>	<u>Description</u>
lfn=pfn	Specifies the local file name lfn given the retrieved copy of permanent file pfn. If lfn= is omitted, the local copy is called pfn.
PW=password	Specifies the password needed to retrieve file pfn if one exists.

<u>Parameter</u>	<u>Description</u>
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=pfn/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 input and output 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=pfn/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=pfn/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

<u>Parameter</u>	<u>Description</u>
F=lfm	Specifies the local file to be listed. The default is the primary file.

LOGIN COMMAND

Same as HELLO command.

Format:

LOGIN,application

This command can appear in a procedure or in a batch job. If used in this manner, the command terminates the job.

LOGOUT COMMAND

Same as BYE command.

Format:

LOGOUT,application

This command can appear in a procedure or in a batch job. If used in this manner, the command terminates the job.

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 or other modifications to the primary file. When followed by the NOSORT command, the 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 dialog.

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:

WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198X.

yy/mm/dd. hh.mm.ss. LEOT1
CDC NETWORK OPERATING SYSTEM.....NOS 2
FAMILY: ,username,password,iaf
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 status messages 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 (system control point) request.
SUSPENDED	The job is no longer executing.
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

The following describes the variable items (in lowercase) in the display:

<u>item</u>	<u>Description</u>
jsn	Job sequence name.
subsystem	Interactive subsystem.

<u>item</u>	<u>Description</u>
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 blank.
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. OUTPUT AVAILABLE. INPUT REQUESTED. IDLE. JOB IN SYSTEM.

If 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. Certain programs (like the FSE editor) recreate the environment that existed at the time of job disruption. 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 reconnected, terminal characteristics for that terminal session are set to the initial values. 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 initial value 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.

WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978,198X.

yy/mm/dd. hh.mm.ss. T02A57
CDC NETWORK OPERATING SYSTEM
FAMILY: ,usernam,passwd,iaf
JSN: AEYT, 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.

? 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.

Text on this page is deleted.

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, FTNLS, and execute subsystems.

Formats:

```
RUN
RUN,I=lfn1
RUN,B=lfn2          or          RUN,C=lfn2
RUN,I=lfn1,B=lfn2    or    RUN,I=lfn1,C=lfn2
RUN,T,q1,q2,...,qn
```

<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 or 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 B and C parameter names are interchangeable.

The fourth format compiles local file lf_{n1} 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 B and C parameter names are interchangeable.

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,OUT PUT,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,OUT PUT,CCC,DDD
```


SECURE LOGIN COMMAND †

The secure login command ensures that your login information goes to the correct host software. You can enter this command at any time you are connected to the network. If you are already connected to a host when you enter this command, you will be unconditionally disconnected and reconnected so that you can proceed with a secure login. This command does not disconnect you from the network.

The secure login command includes a security character that is chosen by your site. The procedure you use to enter this security character depends on your terminal type:

<u>Terminal Type</u>	<u>Secure Login Procedure</u>
Asynchronous (except 2741s)	Press the BREAK key (or ATTN) and type the security character.
X.25 PSN terminal	Press the BREAK key (or ATTN), type the security character, and enter the message transmission signal.
Mode 4	Clear the screen, enter a carriage return, and clear the screen again. Type the security character and press the end-of-block key.
HASP	Type the security character and press the end-of-line key.
2780 or 3780	Enter /* and the security character from the console or a card reader.
2741	Press the NL key twice and the ATTN key once. Type the security character and press the NL key again.
3270	Clear the screen and enter the program attention 1 key once. Type the security character and press the ENTER key.

SHOW COMMAND

The SHOW command calls an interactive procedure to display a screen formatting panel. SHOW is used for testing purposes. It allows you to display a panel without having to write a program that calls the panel.

Format:

SHOW,I=panelname

<u>Parameter</u>	<u>Description</u>
panelname	Specifies the name of a compiled panel in user library PANELIB or in a global library set.

Since the SHOW command is an interactive procedure, you can get help information for the procedure by entering:

SHOW?

† If your terminal is connected to a CDCNET network, refer to the CDCNET Terminal Interface Usage Manual.

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.

Format:

TEXT

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. (Interruption and termination in sequences are discussed in section 16.)

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

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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.

XMODEM COMMAND

The XMODEM command initiates the transfer of files between NOS and a microcomputer using the Christensen protocol.

Format:

XMODEM,fn,td,ft,lf,sp,ec,fm,cf.

XMODEM,FN=fn,TD=td,FT=ft,LF=lf,SP=sp,EC=ec,FM=fm,CF=cf.

Refer to appendix P for more information on the XMODEM command.

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:

<u>Command</u>	<u>Function</u>
ALTER	Changes character strings within lines of the primary file.
DELETE	Deletes lines from the primary file.
DUP	Duplicates lines and moves the duplicates to a specific location in the primary file.
LIST	Selectively lists lines of the primary file.
MOVE	Moves lines from one location to another in the primary file.
READ	Inserts another file at a specified location in the primary file.
RESEQ	Resequences or adds line numbers to the primary file.
WRITE	Copies lines from the primary file to a specified sequenced file.
WRITEN	Copies lines from the primary file to an unsequenced file.

For other methods of file editing, refer to the EDIT, FSE, and XEDIT commands.

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 primary file line number.
Line range	q..r	Specifies a set of sequential lines in the primary 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.
Lines	c ₁ ,c ₂ ,...,c _n	Specifies a nonsequential set of lines and line ranges separated by commas. c _i represents a line or line range. For example, 60..90,120,10..40,350 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.
Line number increment	z	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.
String	/string/	Specifies a sequence of characters, including spaces, delimited by a string delimiter. The string delimiter may not consist of the following: digit , * : letter \$) . @ space The delimiter chosen may not appear in the string. In this subsection, a slant (/) is used as the string delimiter.

<u>Parameter Name</u>	<u>Format</u>	<u>Function</u>
String pair	/string1/string2/	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 //string2/. Refer to the string parameter description for invalid string delimiters.
File name	filename	A file name consisting of 1 to 7 characters where each character is a letter or digit.

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:

**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 primary 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 primary 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
```

```
00100 LET A=1
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
```

```
File is listed to show
corrected version.
```

ALTER COMMAND

The ALTER command enables you to change character strings within specified lines in the primary 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 enter the line correctly.

The following example demonstrates how to change a line by entering the corrected version of the line:

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.
00110 let b=20                           Line is entered changing
                                           B's value from 10 to 20.
                                           READY message is not
                                           given after entry of line.
```

```
list                                     File is listed to show
00100 LET A=1                             corrected version.
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
```

When you enter the lines parameter, and string1 and string2 are null (///), the lines specified in the lines parameter are listed.

Example:

```
list                                     File is listed.
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/                    ALTER command is entered, changing
00110 LET C=20                            B to C in lines 100 through 130.
00120 FOR I=1 to C                          Changed lines are displayed
                                           automatically.

READY.
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
```


DELETE COMMAND

You can delete specified lines from the primary 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
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
```

File is listed.

```
READY.
00105
```

00105 followed by carriage return is entered, deleting line 00105. The READY message is not given after entry of line.

```
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 to show corrected version.

DUP COMMAND

You can duplicate lines and insert them in a specified location in the primary 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 primary file using the LIST command.

Format:

```
LIST,c1,c2,...,cn,/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 primary 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 NEXT 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 primary file with the use of the MOVE command. The MOVE command moves lines from one location in the primary 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 primary file.

If the new line numbers overlap existing line numbers in the primary file, line resequencing occurs as described in the Line Number Overlap subsection.

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.
move,100,110,10
INSERT LINES OVERLAP EXISTING LINES, AT 120.
ENTER Y TO CONTINUE OR N TO STOP
? y
00120 LET A=1
00130 FOR I=1 TO B
00140 A=A*I
00150 PRINT "FACTORIAL ";I,"IS ";A
00160 NEXT I
```

MOVE command is entered, moving line 100 and inserting it immediately following line 110, incrementing subsequent lines by 10. Affected lines are automatically listed.

```
READY.
list
00110 LET B=10
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
```

File is listed to show corrected version.

READ COMMAND

The READ command inserts the lines of a file (read file) at a specific location in the primary 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.

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 primary 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 primary file and incremented by z from the last line number of the primary 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 primary 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 primary file.

If the new line numbers overlap existing lines in the primary file, line resequencing occurs as described in the Line Number Overlap subsection.

In the BASIC subsystem, when inserting a sequenced read file where no overlapping of line numbers occurs, the system handles line references within the read file in the following manner. The system automatically updates line references within statements if the referenced lines are within the read file. It does not alter line references within statements if the lines referenced lie outside of the read file.

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
```

Primary file is listed.

```
READY.
get,skid
```

Read file is made local.

```
READY.
read,skid,150,10
INSERT LINES OVERLAP EXISTING LINES,
AT 200.
ENTER Y TO CONTINUE OR N TO STOP
? y
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
```

READ command is entered. With line increments of 10, the read file is inserted after line 150.

Y is entered to continue command processing.

Inserted 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 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
```

Primary file is listed to show corrected version.

RESEQ COMMAND

Refer to the RESEQ command in section 9.

WRITE COMMAND

You can also append lines and line numbers of the primary 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 primary file containing the specified string are copied to the destination file. If both the lines and string parameters are omitted, all lines of the primary 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
```

Primary 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 primary file to an unsequenced destination file, enter the WRITEN command. The WRITEN command copies specified lines of the primary file to the destination file while removing the line numbers; resulting in an unsequenced destination file.

Format:

```
WRITEN,filename,c1,c2,...,cn,/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
```

Primary 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

LET A=1
LET B=10
FOR I=1 TO B
A=A*I
PRINT "FACTORIAL ";I,"IS ";A
NEXT I
END
```

FILE3 is listed.

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	EVICT	RENAME	SKIPF
BKSP	FCOPY	REQUEST	SKIPFB
CLEAR	LIST80	RESEQ	SKIPR
COPY	LOCK	RETURN	SORT
COPYBF	LO72	REWIND	TCOPY
COPYBR	MFQUEUE	ROUTE	TDUMP
COPYCF	NEW	SCOPY	UNLOAD
COPYCR	OUT	SECHDR	UNLOCK
COPYEI	OVWRITE	SETFAL	VERIFY
COPYSBF	PACK	SETFS	WRITEF
COPYX	PRIMARY	SKIPEI	WRITER
DOCUMENT			

The commands in this section position files, copy data from one file to another, specify the method and format of input or 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. The only commands recommended for copying CRM files are COPY and COPYEI.

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,AL=level.

<u>Parameter</u>	<u>Description</u>																																										
nn	Specifies the 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:																																										
	<table><thead><tr><th><u>Type</u></th><th><u>Equipment</u></th></tr></thead><tbody><tr><td>DB</td><td>885-42 Disk Storage Subsystem (full-track).</td></tr><tr><td>DC</td><td>895-1/2 Disk Storage Subsystem (full-track).</td></tr><tr><td>DD</td><td>834 Disk Storage Subsystem (full-track).</td></tr><tr><td>DE</td><td>Extended memory.</td></tr><tr><td>DF</td><td>887 Disk Storage Subsystem (4K sector; full-track).</td></tr><tr><td>DG</td><td>836 Disk Storage Subsystem (full-track).</td></tr><tr><td>DH</td><td>887 Disk Storage Subsystem (16K sector; full-track).</td></tr><tr><td>DI</td><td>844-21 Disk Storage Subsystem (half-track).</td></tr><tr><td>DJ</td><td>844-41/44 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>844-41/44 Disk Storage Subsystem (full-track).</td></tr><tr><td>DM</td><td>885-11/12 Disk Storage Subsystem (half-track).</td></tr><tr><td>DN</td><td>9853 Disk Storage Subsystem (2K sector; full-track).</td></tr><tr><td>DP</td><td>Distributive data path to extended memory.</td></tr><tr><td>DQ</td><td>885-11/12 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><tr><td>MS</td><td>Mass storage device.</td></tr><tr><td>NE</td><td>Null equipment.</td></tr><tr><td>TT†</td><td>Interactive terminals.</td></tr></tbody></table>	<u>Type</u>	<u>Equipment</u>	DB	885-42 Disk Storage Subsystem (full-track).	DC	895-1/2 Disk Storage Subsystem (full-track).	DD	834 Disk Storage Subsystem (full-track).	DE	Extended memory.	DF	887 Disk Storage Subsystem (4K sector; full-track).	DG	836 Disk Storage Subsystem (full-track).	DH	887 Disk Storage Subsystem (16K sector; full-track).	DI	844-21 Disk Storage Subsystem (half-track).	DJ	844-41/44 Disk Storage Subsystem (half-track).	DK	844-21 Disk Storage Subsystem (full-track).	DL	844-41/44 Disk Storage Subsystem (full-track).	DM	885-11/12 Disk Storage Subsystem (half-track).	DN	9853 Disk Storage Subsystem (2K sector; full-track).	DP	Distributive data path to extended memory.	DQ	885-11/12 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.
<u>Type</u>	<u>Equipment</u>																																										
DB	885-42 Disk Storage Subsystem (full-track).																																										
DC	895-1/2 Disk Storage Subsystem (full-track).																																										
DD	834 Disk Storage Subsystem (full-track).																																										
DE	Extended memory.																																										
DF	887 Disk Storage Subsystem (4K sector; full-track).																																										
DG	836 Disk Storage Subsystem (full-track).																																										
DH	887 Disk Storage Subsystem (16K sector; full-track).																																										
DI	844-21 Disk Storage Subsystem (half-track).																																										
DJ	844-41/44 Disk Storage Subsystem (half-track).																																										
DK	844-21 Disk Storage Subsystem (full-track).																																										
DL	844-41/44 Disk Storage Subsystem (full-track).																																										
DM	885-11/12 Disk Storage Subsystem (half-track).																																										
DN	9853 Disk Storage Subsystem (2K sector; full-track).																																										
DP	Distributive data path to extended memory.																																										
DQ	885-11/12 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.																																										

†This device type applies only to interactive origin jobs.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the 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>
CB	Each dump is written at the BOI of lfn.
CK	Each dump is written at the previous EOI of lfn.

AL=level Specifies the security access level to assign to the file. Unless changed by your site, level can be one of the following names:

LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7

The default is the current access level of the job.

On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information) and for the device to which you are assigning the file.

On an unsecured system, the parameter has no effect. The system does, however, check for valid parameter specifications.

When using this command to assign a file for checkpoint dumps on a secured system, use the AL=level parameter to assign the file at the highest access level valid for the job (ENQUIRE,B returns this information). The CKP command aborts the job if the checkpoint file is not at this access level.

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 input or output 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	Specifies the name of the file to be backspaced.
n	Specifies the number of logical records (decimal) to backspace; if this parameter is omitted, the system assumes n=1. The maximum for n is 262143.
m	Specifies the 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 checkpoint or no-auto-drop status. Certain system scratch files (for example, ZZZZCO, ZZZZC1, ZZZZC2, and ZZZZLD) have no-auto-drop 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.

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	
Input (I=lf _{n1})	Mass Storage or Terminal	Yes	Yes	Yes	Yes	Yes	No	
	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, ns

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, NS=ns.

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 (1 to 7 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 the input and output files are coded files.	

<u>c</u>	<u>Description</u>	<u>Default</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, 1- to 7-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)
-------	--	-------------------------------------

<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, 1000g for S tape copy and 2000g 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
NS=ns	Noise size. Any input block containing fewer than ns characters is considered to be noise and is discarded. This parameter is valid for input tapes in S, L, or F format. The maximum value that can be specified for ns is 41. If NS=0 is specified, the default value of 18 is used.	

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 receive the 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 500 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-bit display code require 12 bits rather than 6, this parameter may produce unforeseen results when copying a file containing 6/12-bit 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 500 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 500 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 TCOPI 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 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-bit display code require 12 bits rather than 6, this parameter may produce unforeseen results when copying a file containing 6/12-bit 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 500 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 and inserts a space before the first character of each line of the file. For a file without printer control characters, insert spaces in the file with this command before routing the file to a printer. The system assumes that a file routed to a printer has a printer control character as the first character of each line. You might get undesirable results if you do not ensure that a printer control character appears first. Refer to appendix H for more information on printer control characters.

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 500 (6-bit) characters. Any characters beyond 500 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 500 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.

<u>Parameter</u>	<u>Description</u>										
x	Provides additional copy specifications. If omitted, one record is copied. The value for x may be one of the following:										
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>x</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>n</td> <td>Copy n records, when n is a decimal number.</td> </tr> <tr> <td>00</td> <td>Copy all records up to and including first zero-length record.</td> </tr> <tr> <td>name</td> <td>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).</td> </tr> <tr> <td>type/name</td> <td>Copy all records up to and including record of specified type and name (refer to Library Record Types in section 15 for list of valid record types).</td> </tr> </tbody> </table>	<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 15 for list of valid record types).
<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 15 for list of valid record types).										
b	Specifies the backspace control. If omitted, 0 is assumed.										
	<table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>b</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Do not backspace.</td> </tr> <tr> <td>1</td> <td>Backspace file lfn₁ one record after copying completes.</td> </tr> <tr> <td>2</td> <td>Backspace file lfn₂ one record after copying completes.</td> </tr> <tr> <td>3</td> <td>Backspace files lfn₁ and lfn₂ one record after copying completes.</td> </tr> </tbody> </table>	<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.
<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, 1- to 7-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 an appropriately documented COMPASS or FORTRAN program. For output listing files not connected to your terminal, the DOCUMENT command honors the page size and print density set for your job using the SET command and the PS and PD symbolic names.

Format:

DOCUMENT, P₁, P₂, ..., P_n.

The parameters P_i can be in any order, but 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 parameter P_i can assume the following values (the default values are presented in a table after the descriptions):

<u>P_i</u>	<u>Description</u>
I=lf _{n1}	Specifies the name of the file that contains the page footing information; this must be a single statement in the following format.

<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 _{n2}	Specifies the 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 _{n3}	Specifies the name of the file on which the output is to be written.				
N=nn	Specifies the number of copies to be produced.				
T=type	Specifies the documentation type: <table border="1" data-bbox="487 1722 1401 1869"> <tbody> <tr> <td>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	Specifies the octal display code of the key character for documentation.				

<u>Pi</u>	<u>Description</u>
P=pp	Specifies the number of print lines per page.
NR	Disables rewind on the S (source) file.
NT	Negates table generator.
TC	Lists 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	47(*)	03(C)	Octal display code of check character (two octal digits).
P	60	80	Number of print lines per page. This is either the system default or the user-set value of the PS symbolic name (refer to the SET command in section 6).
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.

<u>Parameter</u>	<u>Description</u>
lfn _i	Names of the files 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.
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.

†Refer to the ROUTE command in this section.

FCOPY COMMAND

The FCOPY command converts a file from one character code set to another.

Format:

FCOPY,P=lf_{n1},N=lf_{n2},PC=cs₁,NC=cs₂,PL=lt₁,NL=lt₂,FL=f_l,LB=l_b,R,A.

<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.

When copying from lf_{n1}, the system normally reads lf_{n1} to EOI, preserving all EOR and EOF marks unless lf_{n1} or lf_{n2} is an S- or L-format tape. If an S- or L-format tape is specified for lf_{n1}, the system reads lf_{n1} only to the first tape mark. If an S- or L-format tape is specified for lf_{n2}, the system reads lf_{n1} only to the first EOR, EOF, or EOI encountered. If you want to copy an entire file to an S- or L-format tape, you can do so by first removing all EORs and EOFs using the PACK command.

PC=cs₁ Specifies the code set of lf_{n1}. The default value for cs₁ is ASCII.

<u>cs₁</u>	<u>Description</u>
ASCII	Specifies the 6/12-bit display code that supports the ASCII graphic 63- or 64-character set present on the current system.
ASCII8	Specifies the 7-bit ASCII code, right-justified in 12-bit bytes.
ASCII88	Specifies the 8-bit ASCII code, right-justified in 8-bit bytes.
ASCII63	Specifies the 6/12-bit display code that supports the ASCII graphic 63-character set.
ASCII64	Specifies the 6/12-bit display code that supports the ASCII graphic 64-character set.
ASCFL	Specifies the 8-bit ASCII code on S- and L-format tapes (with fixed line length and no line terminators).
DIS	Specifies the 6-bit display code that supports the CDC graphic 63- or 64-character set present on the current system.
DIS63	Specifies the 6-bit display code that supports the CDC graphic 63-character set.
DIS64	Specifies the 6-bit display code that supports the CDC graphic 64-character set.
EBCFL	Specifies the 8-bit EBCDIC code set on S- and L-format tapes (with fixed line length and no line terminators).

Parameter

Description

- NC=cs₂ Specifies the code set of lfn₂. The default value for cs₂ is ASCII8. The possible values for cs₂ are the same as for cs₁.
- PL=lt₁ Specifies the line terminator used for file lfn₁. Table 9-2 shows valid line terminators for each code set. Values that can be specified for lt₁ are as follows:

<u>Value</u>	<u>Line Terminator</u>
CR	Carriage return.
CRLF	Carriage return and line feed.
FF	Form feed.
LF	Line feed.
LFCR	Line feed and carriage return.
RS	Record separator.
US	Unit separator.
ZB	Zero byte.
n	Specified octal character value (See table 9-2).

The default value for lt₁ for most code sets is ZB. For ASCII88, the default is US. ASCFL and EBCFL do not use line terminators.

- NL=lt₂ Specifies the line terminator to be used in file lfn₂. Table 9-2 shows valid line terminators for each code set. Values that can be specified for lt₂ are the same as for the PL=lt₁ parameter. The default value for lt₂ for most code sets is ZB. For ASCII88, the default is US. ASCFL and EBCFL do not use line terminators.

Table 9-2. Code Set-Line Terminator Reference Chart

Code Set	ZB	CR	FF	LF	US	RS	CRLF	LFCR	n
DIS	Y	N	N	N	N	N	N	N	0-77B
DIS63	Y	N	N	N	N	N	N	N	0-77B
DIS64	Y	N	N	N	N	N	N	N	0-77B
ASCII	Y	Y	Y	Y	Y	Y	N	N	0-77B, 7400-7477B, or 7600-7677B
ASCII63	Y	Y	Y	Y	Y	Y	N	N	0-77B, 7400-7477B, or 7600-7677B
ASCII64	Y	Y	Y	Y	Y	Y	N	N	0-77B, 7400-7477B, or 7600-7677B
ASCII8	Y	Y	Y	Y	Y	Y	Y	Y	0-177B
ASCII88	N	Y	Y	Y	Y	Y	Y	Y	0-377B
ASCFL	N	N	N	N	N	N	N	N	N
EBCFL	N	N	N	N	N	N	N	N	N

- FL=f1 Specifies the length of fixed length lines for S- and L-format tapes. The default value is 80 characters. This parameter is valid only if ASCFL or EBCFL was specified for cs₁ or cs₂.

<u>Parameter</u>	<u>Description</u>
LB=lb	Specifies the number of lines per block for S- and L-format tapes. The default value is 3840 divided by the fixed line length (FL). (For example, if FL=80, the default for LB is 3840/80 or 48 lines per block.) For S-format tapes, the maximum value of FL*LB is 3840. This parameter is valid only if ASCFL or EBCFL was specified for cs ₁ or cs ₂ .
R	Rewinds lfn ₁ and lfn ₂ before and after the conversion. The default is no rewinding.
A	Aborts the FCOPY command if errors occur during the conversion. The default is not to abort when errors occur.

FCOPY reads lfn₁ to its EOI, preserving its EOR and EOF marks on the converted file unless lfn₁ or lfn₂ is an S- or L-format tape. If lfn₁ is an S- or L-format tape, FCOPY reads it only to the first tape mark. If lfn₂ is an S- or L-format tape, FCOPY reads lfn₁ only to the first EOR, EOF, or EOI encountered. 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.

Files converted to 7-bit ASCII code can be listed on a local batch printer (refer to the ROUTE command) but cannot be listed at an 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-bit display code (FILE2) to a file containing 7-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. For output listing files not connected to your terminal, the LIST80 command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

Format:

LIST80, lfn₁, lfn₂, NR.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the file from which to copy; if this parameter is omitted, file LIST is assumed.
lfn ₂	Specifies the file to which to copy; if this parameter is omitted, file OUTPUT is assumed.
NR	Indicates that lfn ₁ 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, ..., lfn_n.

<u>Parameter</u>	<u>Description</u>
lfn ₁	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	Specifies that the reformat parameters are on file INPUT.								
I=lf _{n1}	Reformat parameters are on file lf _{n1} .								
I=0	There is no input file of reformat parameters. If the I parameter is omitted, I=0 is assumed.								
S	Specifies that the data to be reformatted is on file SCR.								
S=lf _{n2}	Specifies that the data to be reformatted is on file lf _{n2} . If the S parameter is omitted, SCR is assumed.								
L	Specifies that the reformatted data is listed on file OUTPUT.								
L=lf _{n3}	Specifies that the reformatted data is listed on file lf _{n3} . lf _{n3} is always rewound unless the NR parameter is specified. If the L parameter is omitted, OUTPUT is assumed.								
T	Specifies that the file to be reformatted is of type B.								
T=x	Specifies that the file to be reformatted is of type x.								
	<table><thead><tr><th><u>x</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>M</td><td>Modify source data.</td></tr><tr><td>C</td><td>COMPASS source data.</td></tr><tr><td>B</td><td>Other source data.</td></tr></tbody></table>	<u>x</u>	<u>Description</u>	M	Modify source data.	C	COMPASS source data.	B	Other source data.
<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	Specifies that the number of characters per output line is 72.								
H=xxx	Specifies that the number of characters per output line is xxx (maximum allowed is 150 characters). xxx must be greater than or equal to the number of characters being moved (Nx) plus the starting column number of the destination field (Ox). If the H parameter is omitted, 72 is assumed.								
LP	Specifies that the output is formatted for the line printer.								
NR	Specifies that the 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 1 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 
ARGUMENT          VALUE
INPUT FILE NAME:          ? 
SOURCE FILE NAME:        SCR      ? source 
OUT PUT FILE NAME:       OUT PUT ? out 
SOURCE FILE TYPE:        BATCH ? b 
OUT PUT LINE LENGTH:     72 CHARS.? 90 
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. ? n4=1 
? i4=2 
? o4=75 
? 
LO72 COMPLETE.
```

Batch job entry:

```
LO72,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.

MFQUEUE COMMAND

The MFQUEUE command sends a local file from your host to the input or output queue of a remote host. The MFQUEUE command functions much like the ROUTE command. For most file transfers between mainframes, ROUTE is the preferred command. Unlike the ROUTE command, MFQUEUE provides the capability of sending a separate routing directive with the file or of indicating that the file has a nonstandard data format. The remote host processes the file according to the routing directive. Only when you need special routing specifications not provided by the ROUTE command do you need to use the MFQUEUE command. You must be authorized to use the MFQUEUE command (refer to the LIMITS command).

Format:

MFQUEUE, lfn, ST=lid, DD=dd, I=dirfile, PC=c, DC=dc.

PARAMETER DESCRIPTIONS

Parameters lfn and ST=lid are required and lfn must be the first parameter; the other parameters are optional and order-independent.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the local file to be routed to the remote host. The file name must be one to seven alphanumeric characters.
ST=lid	Specifies the logical identifier (LID) of the remote host to which the file is to be routed. The LID specification must be three alphanumeric characters. ST=lid may not specify the local host.
DD=dd	Specifies the data format of the file to be transferred. The DD=dd parameter both describes the character set of lfn on the host where it currently resides and also specifies the character set lfn is to have after the transfer. For file transfers between NOS mainframes and between NOS/BE and NOS mainframes this parameter is effectively ignored. The file will have the same format on both mainframes. The interpretation of the DD=dd parameter depends on the operating system(s) involved in the file transfer. The following paragraphs describe how NOS interprets the DD=dd parameter. For its interpretation by other operating systems, refer to the Remote Host Facility Usage manual.

Parameter

Description

On NOS, you can specify the following values for dd:

<u>dd</u>	<u>Description</u>
C6	The file contains character data in a 64-character (or less) subset of the ASCII 128-character set. The character code of lfn is 6-bit display code and its lines are zero-byte terminated (Z-type records).
C8	The file contains character data in a larger than 64-character subset of the ASCII 128-character set. The character code of lfn is the 7-bit ASCII code (right-justified in 12-bit fields) and its lines are zero-byte terminated (Z-type records).
US	Specifies that the file contains binary data in structured format with EORs and EOFs. The file is transmitted as a continuous string of binary data, broken only by EORs and EOFs as appropriate.
UU	Specifies that the file contains binary data without EORs and EOFs. NOS treats UU as US.

If the file is character data, you do not need to specify a DD=dd parameter. The remote host determines the format in which to write the file. If NOS is the remote host and the DD=dd parameter has not been specified, NOS assumes character data and reads/writes the file in C6 format.

I=dirfile Specifies the local file which contains the routing directive that the system sends to the remote host. The file name must be one to seven alphanumeric characters. If you specify only the keyword I, file INPUT becomes the directive file. If you omit the I=dirfile parameter, the system reads directives from the lines immediately following the MFQUEUE command. The lines must begin with the prefix character defined by the PC=c parameter. The system reads the lines until it comes to a line without the prefix character. It then strips off the prefix character from each line and concatenates the lines into one single directive to send to the remote host. This directive must not exceed the 256-character limit.

PC=c Specifies the prefix character for the lines of the routing directive. The default is an asterisk (*). The prefix character can be any 6-bit display code character, except a closing parenthesis, a dollar sign, a period, or a colon. To avoid ambiguity between commands and routing directives, use alphabetic values for c cautiously.

<u>Parameter</u>	<u>Description</u>
DC=dc	Specifies the disposition codes for the file to be transferred. You can specify one of the following codes for dc:

<u>dc</u>	<u>Description</u>
CP	Punch the file in coded format.
IN	Queue the file for input.
IX	Queue the file for input but do not return the output unless explicitly directed to do so.
LP	Print the file.
PB	Punch the file in system binary format.
PH	Same as CP.
PR	Same as LP.
PU	Same as CP.
P8	Punch in 80-column binary format.
SB	Same as PB.
SP	Plot in binary format.

Example:

To route a 7-bit ASCII file (MYFILE) to a remote mainframe (with REM as its LID) for execution, enter this command:

MFQUEUE,MYFILE,ST=REM,DC=IN,DD=C8.

THE ROUTING DIRECTIVE

The routing directive provides additional routing specifications for the queued file. The I=dirfile parameter determines the residence of this directive. The host strips the prefix characters from the lines of this directive and forwards the directive unaltered (except for prefix characters) to the remote host for processing. Hence, the directive must be a command recognized by the remote host.

This section documents only the routing directive recognized by a NOS 2 remote host. For information on routing directives for other remote hosts, refer to the Remote Host Facility Usage manual.

The routing directive accepted by a NOS 2 remote host closely resembles its ROUTE command.

Format:

ROUTE, lfn, p₁, p₂, ..., p_n.

The parameter descriptions for this directive are the same as those for the ROUTE command with the following exceptions:

- The lfn parameter is not required on the ROUTE directive and can be omitted.

- Parameters p_i can be any of the following as described for the ROUTE command:

DC=dc	FM=fm	SC=sc
DO=lid	IC=ic	SCL=scl
EC=ec	ID=id	TID=tid
FC=fc	PI=pi	UJN=ujn
FID=fid	REP=rep	UN=un

EXAMPLE:

Assume you are at mainframe MFA. To route local file RBPRINT from mainframe MFA to a remote batch printer logged in with user name ABC on NOS mainframe MFB, enter a command similar to the following:

```
MFQUEUE,RBPRINT,ST=MFB.
*ROUTE,DC=PR,UN=ABC.
```

You can accomplish the same task with the following ROUTE command:

```
ROUTE,RBPRINT,DC=PR,UN=ABC,ST=MFB.
```

NEW COMMAND

The NEW command creates a primary file.

Format:

```
NEW,lfn/ND.
```

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the name of the file to be made the primary file.
ND	Specifies that 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 checkpoint or no-auto-drop status. Certain system files (for example, ZZZZCO, ZZZZC1, ZZZZC2, and ZZZZLD) always have no-auto-drop 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 local 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 binary format.

OVWRITE COMMAND

The OVWRITE command destroys the contents of files by overwriting them. This command does not apply to tape files or to files on which you cannot write (for example, direct access files attached in a mode besides WRITE or files that have been locked). The command overwrites the entire file.

Formats:

OVWRITE,lfn₁,lfn₂,...,lfn_m/OP=p₁p₂.

or

OVWRITE,*,lfn₁,lfn₂,...,lfn_m/OP=p₁p₂.

<u>Parameter</u>	<u>Description</u>
lfn _i	Specifies a file assigned to your job. This file cannot be a tape file.
OP=p ₁ p ₂	Specifies how the files are to be overwritten and whether the files are to be released from the job after the overwriting. The value p _i can be any of the following:

<u>Parameter</u>	<u>Description</u>
<u>P_i</u>	<u>Description</u>
Z	Overwrite the files with zeroes. This is the default. You cannot specify both Z and X.
X	Overwrite the files first with zeroes, then with ones, and then with an alternating pattern of ones and zeroes. You cannot specify both Z and X.
R	Release the files from the job after overwriting them.

The first format shown overwrites files lfn₁. The second format overwrites all files assigned to the job except files lfn₁.

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.

<u>Parameter</u>	<u>Description</u>
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 local 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. 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 catalog. 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, AL=level. 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.						
AL=level	Specifies the security access level to assign to the file. Unless changed by your site, level can be one of the following names: LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7 The default is the current access level of the job. On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information) and for the device to which you are assigning the file. On an unsecured system, the parameter has no effect. The system does, however, check for valid parameter specifications. When using this command to assign a file for checkpoint dumps on a secured system, use the AL=level parameter to assign the file at the highest access level valid for the job (ENQUIRE,B returns this information). The CKP command aborts the job if the checkpoint file is not at this access level.						

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.
```

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.

† Any mass storage file used as a checkpoint file must have write permission.
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 line number increment. 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 so that the system changes the line number references within the source statements. The system supplies 5-digit line numbers and line number references; excess surrounding blanks are used in the expansion of line number references.

In order to resequence a BASIC program written in 6/12-bit display code, you must call RESEQ from an interactive job which is in ASCII mode.

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.
```

<u>Parameter</u>	<u>Description</u>
lfn _i	Specifies the name of a file assigned to your job.

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 checkpoint or no-auto-drop status. Certain system scratch files (for example, ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD) have no-auto-drop 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.

If the file is a magnetic tape file 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 returns 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 returns 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 returns the tape.

Refer to Magnetic Tape Files in section 2 and Tape Management command in section 12 for further information about tape files, and to appendix G for a description of an EOF1 label.

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.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Specifies the name of a file assigned to your job.

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.

With the proper validation, you can use the ROUTE command to queue files on other mainframes. Enter the LIMITS command to display the limits and user permissions for your user name. The MFQUEUE command also queues files on other mainframes, but use the ROUTE command for this purpose unless you need special routing specifications not provided by ROUTE.

PARAMETER DESCRIPTIONS

Format:

ROUTE, lfn, P₁, P₂, ..., P_n.

<u>Parameter</u>	<u>Description</u>
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.

<u>P_i</u>	<u>Description</u>																
DC=dc	Specifies the disposition of the queued file.																
	<table><thead><tr><th><u>dc</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>IN</td><td>Queue the file with an input disposition. Normal job input file format is required and the first record of the file must be in 6-bit display code. If the Job command within the file is in error, the file is not queued and remains a local file.</td></tr><tr><td>LP</td><td>Queue the file for printing on any printer.</td></tr><tr><td>LR</td><td>Queue the file for printing on 580-12 printer.</td></tr><tr><td>LS</td><td>Queue the file for printing on 580-16 printer.</td></tr><tr><td>LT</td><td>Queue the file for printing on 580-20 printer.</td></tr><tr><td>LX</td><td>Queue the file for printing on 5870 Non-Impact Printer.</td></tr><tr><td>NO</td><td>Queue the file with an input disposition. Output not explicitly routed by the job is discarded at job completion. The dayfile is not written to file OUTPUT. Normal job input file format is required and the first record of the file must be in 6-bit display code. If the Job command within the file is in error, the file is not queued and remains a local file.</td></tr></tbody></table>	<u>dc</u>	<u>Meaning</u>	IN	Queue the file with an input disposition. Normal job input file format is required and the first record of the file must be in 6-bit display code. If the Job command within the file is in error, the file is not queued and remains a local file.	LP	Queue the file for printing on any printer.	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.	LX	Queue the file for printing on 5870 Non-Impact Printer.	NO	Queue the file with an input disposition. Output not explicitly routed by the job is discarded at job completion. The dayfile is not written to file OUTPUT. Normal job input file format is required and the first record of the file must be in 6-bit display code. If the Job command within the file is in error, the file is not queued and remains a local file.
<u>dc</u>	<u>Meaning</u>																
IN	Queue the file with an input disposition. Normal job input file format is required and the first record of the file must be in 6-bit display code. If the Job command within the file is in error, the file is not queued and remains a local file.																
LP	Queue the file for printing on any printer.																
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.																
LX	Queue the file for printing on 5870 Non-Impact Printer.																
NO	Queue the file with an input disposition. Output not explicitly routed by the job is discarded at job completion. The dayfile is not written to file OUTPUT. Normal job input file format is required and the first record of the file must be in 6-bit display code. If the Job command within the file is in error, the file is not queued and remains a local file.																

Pi

Description

<u>dc</u>	<u>Meaning</u>
PB	Same as SB.
PH	Same as PU.
PL	Queue the file for plotting.
PR	Same as LP.
PU	Queue the file for punching in coded format.
P8	Queue the file for punching in 80-column binary format.
SB	Queue the file for punching system binary format.
SC	Rescind prior routing and change the file type to local. If no prior routing exists, SC is ignored.
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.

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.

If DC is omitted and lfn is: ROUTE assumes DC=dc is:

OUTPUT	DC=LP
PUNCH	DC=PU
PUNCHB	DC=SB
P8	DC=P8

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. The security count for the user name that did the ROUTE is decremented accordingly. If you specify a remote host (ST=lld₂) that does not perform USER command validation, the local host does not check your queued file for a USER command.

If a ROUTE command sends a file to a remote host with an input disposition not supported on that host (for example, DC=TO or DC=NO to a non-NOS 2 host), the remote host accepts the input file but treats it as though it were routed with DC=IN. Hence, job output from a job routed to a non-NOS 2 host with DC=NO or DC=TO returns and prints locally.

Pi

Description

On a dual-state system, if DC=NO is specified when routing a NOS/VE batch job from NOS, the job output is discarded at termination of the NOS/VE job. If DC=TO is specified, job output is queued with a wait disposition.

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, DC=NO, or DC=TO.

DO=lid₁ Specifies the logical identifier (LID) of the mainframe to receive the output generated by the routed file. The LID can be any three-character, alphanumeric string defined by your site or you can specify an asterisk.

By default, output returns to the originating mainframe, including cases where your routed file submits jobs to still other mainframes. For example, you are at mainframe MAA and you route job X to mainframe MBB. Job X routes job Y to mainframe MCC. Then the output from both jobs X and Y return to mainframe MAA by default.

DO=* returns the output to the mainframe that processed the ROUTE command on which DO=* appears. For example, you are at mainframe MAA and you route job A to mainframe MBB. Job A routes job B to mainframe MCC with DO=* specified. Then output from job A returns to MAA but output from job B returns to MBB.

In both preceding examples, all the mainframes must be directly connected; that is, no two mainframes have to rely on a third for file transfers.

You must have special authorization to use this parameter (refer to the LIMITS command).

EC=ec Defines external characteristics for files queued for printing or punching.

For files queued for printing, ec can be the following. (For more information on print trains associated with various EC codes, refer to Jobs Using Line Printers in appendix A.)

<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 (except for CYBER model 810 systems using a PSU printer; to route files to a PSU printer, use the FC parameter). File lfn must be a 7-bit ASCII file. Refer to the FCOPY command earlier in this section for information on how to convert a file to 7-bit ASCII.

Pi

Description

<u>ec</u>	<u>Meaning</u>
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=value This parameter has two possible uses depending on your site or host configuration:

For routing of output requiring special forms, specify FC=fc, where fc is a 2-character code assigned by the system operator to an output device. This prevents output of a file before its special forms are placed in the output device. In this case, omission of the parameter results in a null value.

Pi

Description

For sites supporting two 533/536 or 585 printers, an ASCII graphic 64-character printer and an ASCII graphic 95-character printer, you can use the FC parameter to select either printer. Specify FC=AS to select the 95-character printer. In this case, omission of the parameter results in routing to the 64-character printer.

FID=ujn FID is a NOS/BE parameter included for compatibility. Under NOS, this parameter performs the same function as the UJN parameter.

FM=fm Indicates routing to a remote batch or interactive terminal logged in with the family name specified. You must have special authorization to specify an alternate family name (refer to the LIMITS command). 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	6-bit display code.
ASCII	7-bit ASCII code.
BIN	Binary.

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

<u>Pi</u>	<u>Description</u>
ID=id	Selects the device locally associated 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).
PI=pi	Selects a print train image denoted either by a digit from 1 to 7 or by a one- to seven-character symbolic name assigned by your site.
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 63; therefore, the number of copies that can be sent ranges from 1 to 64. Values for rep beyond its range are set to zero, an informative message is sent to the dayfile, and one copy is routed to the destination. The default value is zero; only one copy is routed.
SC=sc	Specifies a programmable format control (PFC) array for the 580 PFC printer, or a vertical format definition for a 533/536, 537, or 585 printer. The system is released with two PFC arrays, a default (SC=0), and an alternate (SC=1). If your site has 533/536, 537, or 585 printers, the system contains six vertical format definitions for these printers with a default of SC=3. For more information on PFC arrays and vertical format definitions, refer to appendix H and the NOS Version 2 Analysis Handbook.

For 533/536, 537, and 585 printers, you may choose one of the following sc values:

<u>sc</u>	<u>Description</u>
1	12-inch forms at 6 lines/inch.
2	12-inch forms at 8 lines/inch.
3	11-inch forms at 6 lines/inch.
4	11-inch forms at 8 lines/inch.
5	8.5-inch forms at 6 lines/inch.
6	8.5-inch forms at 8 lines/inch.

If no sc value is specified, the system default (SC=3) is used. Refer to NOS Version 2 Analysis Handbook for more information on 533/536, 537, and 585 printers.

<u>Pi</u>	<u>Description</u>
SCL=scl	Specifies the service class to be assigned to this routed file. The value scl must be one of the following:

<u>scl</u>	<u>Description</u>
BC	Local batch.
CT	Communication task.
DF	The default service class according to the origin type of the routed file. The LIMITS command shows your default service class assignments.
DI	Detached interactive.
In	Installation class n ($0 \leq n \leq 3$).
MA	Maintenance.
NS	Network supervisor.
RB	Remote batch.
SY	System.
TS	Interactive.

When using the ROUTE command to queue a file to a remote host, the service class for your job must be valid on the remote host and all mainframes in the link to the remote host.

For a file queued for output, you can only specify a service class valid for your current job. The default service class for such a file is the service class of the current job.

For a file queued for input (DC=IN, DC=NO, and DC=TO), you can only specify a service class valid for local batch jobs. The default service class in this case is the service class the system assigns by default to any of your local batch jobs. If you specify a service class with this parameter for a file queued for input and the file's Job command also has a service class specification, the service class specified on the ROUTE command takes precedence. Refer to Job Service Classes in section 3 for more information on service classes.

ST=lid ₂	Specifies the logical identifier (LID) of the mainframe to which you are routing the specified file. The LID can be any three-character alphanumeric string defined by your site or you can specify an asterisk.
---------------------	--

By default, the mainframe processing the ROUTE command also processes the routed file if it is queued for input. This mainframe also processes the routed file if it is queued for output unless the job containing this ROUTE command emanated from another job that is diverting the output elsewhere.

ST=* ensures that the mainframe processing the ROUTE command always processes the routed file.

<u>Pi</u>	<u>Description</u>
	In a dual-state environment, you can route an input file to NOS/VE by specifying the NOS/VE LID on this parameter. The file must be in ASCII (6/12-bit display code) format.
	The ST=lid ₂ on the ROUTE command overrides the ST=lid parameter on the Job command of the routed file.
	You must have special authorization to use this parameter (refer to the LIMITS command).
TID=tid or TID=C	The TID=tid form of this parameter functions the same as the UN=un parameter and is included for compatibility with NOS/BE. The TID=C form of the parameter specifies routing to the central site. If you specify only the keyword TID, implicit remote routing is selected (refer to Implicit Routing below).
UJN=ujn	Specifies a user job name (UJN) for the file to be routed. Except for input dispositions, 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 default 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.

You can use the ENQUIRE command to determine whether your routed file is destined for the central site or a remote batch terminal. If you specify the JSN=jsn or UJN=ujn parameter, ENQUIRE returns either RB (remote batch routing) or BC (routing to the batch devices at the central site) in the DS field of the display.

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 500 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 TCOPY 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 be 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, the file is copied to EOF.

<u>Parameter</u>	<u>Description</u>
fchar	Specifies the first character position (1 to 500 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 500 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 500.†
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-bit 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-bit display code require 12 bits rather than 6, this parameter may produce unforeseen results when copying a file containing 6/12-bit 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 500, 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.
```


SECHDR COMMAND

The SECHDR command adds security-related information to a file. You can choose to have this information placed on banner pages throughout the file, placed at the top and bottom of each page, or both.

Format:

SECHDR, lfn, FF, OP=p₁p₂p₃p₄.

The parameter lfn is required and must be the first parameter. The other parameters are optional and order-independent.

<u>Parameter</u>	<u>Description</u>										
lfn	Specifies a file assigned to your job.										
FF	Specifies that the file has already been formatted for printing; that is, carriage control characters (such as those controlling page ejects and page density) have already been inserted.										
OP=p ₁ p ₂ p ₃ p ₄	Specifies where the security information will be inserted in the file. OP=B is the default. You have the following choices:										
	<table><thead><tr><th><u>Pi</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>B</td><td>Print the security information on two banner pages, one at the beginning of the file and one at the end of the file.</td></tr><tr><td>F</td><td>Print the security information on a banner page at the beginning and the end of every logical file within file lfn.</td></tr><tr><td>R</td><td>Print the security information on a banner page at the beginning and the end of every logical record within file lfn.</td></tr><tr><td>P</td><td>Print the access level of the file at the top and bottom of every page of output.</td></tr></tbody></table>	<u>Pi</u>	<u>Description</u>	B	Print the security information on two banner pages, one at the beginning of the file and one at the end of the file.	F	Print the security information on a banner page at the beginning and the end of every logical file within file lfn.	R	Print the security information on a banner page at the beginning and the end of every logical record within file lfn.	P	Print the access level of the file at the top and bottom of every page of output.
<u>Pi</u>	<u>Description</u>										
B	Print the security information on two banner pages, one at the beginning of the file and one at the end of the file.										
F	Print the security information on a banner page at the beginning and the end of every logical file within file lfn.										
R	Print the security information on a banner page at the beginning and the end of every logical record within file lfn.										
P	Print the access level of the file at the top and bottom of every page of output.										

You can specify any combination of these options.

The banner pages have the following header lines:

FILE NAME = file name

DATE PRINTED = yy/mm/dd

USER NAME = user name

where the lowercase characters represent job-specific information. Following those header lines, the system prints the security access level of the file in large block characters.

Example:

The following command specifies that security information is to be placed both in a header and a footer on each page of the file and also on banner pages that are to be printed at the beginning and ending of each logical file within FILE213:

```
SECHDR,FILE213,OP=FP.
```

SETFAL COMMAND

The SETFAL command changes the security access level of a local file. In most cases, a local file receives the access level of the job that creates it. Refer to Security Features in section 3 for a detailed discussion of file access levels.

Format:

```
SETFAL, lfn, AL=level.
```

The two parameters are required and are order-dependent.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies a local file. The file cannot be a direct access permanent file or a magnetic tape file.
AL=level	Specifies the security access level you want assigned to the file. Unless changed by your site, level can be one of the following names: LVLO, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7

On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information). In addition, to lower the access level of a file, you must be authorized to do so (refer to the LIMITS command).

On an unsecured system, the system does not use this access level assignment in monitoring access to the file. It does, however, become an attribute of the file.

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, ..., lfn_n/FS=fs.

or

SETFS, *, lfn₁, lfn₂, ..., lfn_n/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, ZZZZC0, 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.

<u>Parameter</u>	<u>Description</u>
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. The maximum for n is 262143.
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. The maximum for n is 262143.
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>
lfn	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. The maximum for n is 262143.
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. The default is 0.
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, lfn, NC=n.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the file to be sorted. lfn 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. $n < 10$. 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 I 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, ns.

Order-independent format:

TCOPY, I=lfn₁, O=lfn₂, F=format, TC=tc, N=copycnt, CC=charcnt, EL=erlimit, PO=p₁p₂, L=lfn₃, NS=ns.

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="0"> <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 17 ₈ 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=Dor TC=EOD)
	F or EOF When this TC=tc value is set, the N keywords 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. An EL=U specification allows an unlimited number of errors before an abort. This includes parity errors and block-too-large errors which are returned by the tape subsystem after completing recovery procedures. It also includes invalid block format errors (invalid byte-count coded format and/or unused bitcount) 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). <ul style="list-style-type: none"> E Input blocks with parity errors or block-too-large errors are processed (copied). 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. 	Error blocks are skipped. Lines exceeding the maximum line size are split into multiple blocks.
L=lf _{n3}	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 lf _{n3} must not be the same as lf _{n2} .	OUTPUT
NS=ns	Noise size. Any input block containing fewer than ns characters is considered to be noise and is discarded. This parameter is valid only for E format to B format conversions. The maximum value that can be specified for ns is 41. If NS=0 is specified, the default value of 18 is used.	

Example:

The following TCOY 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 hexadecimal, 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-bit 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, ..., Pn.

<u>Pi</u>	<u>Description</u>
I=lf _{n1}	Specifies the local file to be dumped (default is TAPE1).
L=lf _{n2}	Specifies the local file to which the output is written (default is OUTPUT). If lf _{n2} is not a local file, TDUMP creates it. It does not rewind lf _{n2} before or after the dump.
H	Specifies a hexadecimal dump only.
O	Specifies an octal dump only.
A	Specifies an alphanumeric dump only.
	If more than one of the H, O, or A parameters are specified, the last one overrides. If no H, O, or A parameter is specified, TDUMP lists a combined octal and alphanumeric dump.
R=rcount	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=fcount	Specifies the maximum decimal number of files to be dumped. If F is omitted, the dump continues to EOI. If F=0, the 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 _{n1} before the dump (default is to rewind lf _{n1}).
CW	Specifies that control words as well as data are to be dumped. If CW is omitted, control words are not dumped.

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. hh.mm.ss. 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 --
```

File X was then dumped to file Y using the CW parameter, producing the following output. Note that when the CW parameter is used, EOR and EOF markers are dumped along with the data with no marked separation.

```

- FILE DUMP - TDUMP,I=X,L=Y,CW.
yy/mm/dd. hh.mm.ss. PAGE 1.
F 1 R 1 W 0- 0000 0100 0000 0000 0036 0102 0304 0506 0710 1112 1314 1516 1720 2122
2324 2526 2730 3132 0000 0000
      A                               3 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
F 1 R 1 W 4- 0102 0304 0506 0710 1112 1314 1516 1720 2122 2324 2526 2730 3132 0000
0000 0000 0000 0010 4676 0001
      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                               H -
-- END OF INFORMATION --
-- END OF DUMP --

```

File X was then converted to ASCII8 format using FCOPY to create file AX. File AX was then dumped to file Y using the H parameter, producing the following output.

```

- FILE DUMP - TDUMP,I=AX,L=Y,H.
yy/mm/dd. hh.mm.ss. PAGE 1.
F 1 R 1 W 0- 041 042 043 044 045 046 047 048 049 04A 04B 04C 04D 04E 04F 050 051 052 053 054
F 1 R 1 W 4- 055 056 057 058 059 05A 000 000 000 000 041 042 043 044 045 046 047 048 049 04A
F 1 R 1 W 10- 04B 04C 04D 04E 04F 050 051 052 053 054 055 056 057 058 059 05A 000 000 000 000
-- 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, ..., lfnn.
```

or

```
UNLOAD, *, lfn1, lfn2, ..., lfnn.
```

Parameter

Description

lfn_i Specifies the name of a file assigned to your job.

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 checkpoint or no-auto-drop status. Certain system scratch files (for example, ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD) have no-auto-drop 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. It does not decrease the number of tapes or packs that were scheduled for the job with the RESOURC command. If the file is a magnetic tape file 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.

<u>Parameter</u>	<u>Description</u>
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 9-3. For output listing files not connected to your terminal, the VERIFY command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

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>P_i</u>
	<u>Description</u>
	N=x
	In the case of multifile files, specifies the number of files to verify. The default is N=1.
	If you specify just the keyword N, verification proceeds until the end of information is reached on both files.
	If you specify N=0, verification terminates whenever an empty file is encountered on either file.
	E=y
	Specifies the maximum number of errors to list. The default is E=100. Specifying E has the same effect as E=0.
	L=lfn ₃
	List errors on file lfn ₃ . If L is omitted, the system assumes L=OUTPUT.

<u>Pi</u>	<u>Description</u>
A	Abort after verification completed if errors occurred.
R	Rewind both files before and after the verification.
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 ₈ words, and for an L tape, it is 2000 ₈ 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.

NOTE

When printing output for text files, VERIFY uses the first seven characters of the first word of the record as the record name. VERIFY assumes that the record is in 6-bit display code. Therefore, if you attempt to display VERIFY output for a record in 6/12-bit display code while your terminal is in ASCII mode, the output display may be garbled. To avoid this problem, you should set your terminal to NORMAL mode before listing VERIFY output.

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 9-3. Compatible File Structures for the VERIFY Command

		SECOND FILE (lfn ₂)						
		Mass Storage	Tape Formats					
			I	SI	S	L	F	
Mass Storage		Yes	Yes	Yes	No	No	No	
FIRST FILE (lfn 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. The maximum for x is 262143.

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. The maximum for x is 262143.

PERMANENT FILE COMMANDS

10

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.
DROPDS	Releases the disk space associated with a permanent file for which a valid alternate storage copy exists. The local file copy (if any) is also returned.
FTP	Provides file transfer capabilities between connected TCP/IP hosts.
GET	Retrieves a copy of an indirect access permanent file as a local file.
MFLINK	Transfers a file from one mainframe to another or alters the attributes of a file on another mainframe.
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.
RECLAIM	Creates and manages tape and disk backup copies of permanent files.
REPLACE	Copies a local file over an indirect access permanent file.
SAVE	Creates an indirect access permanent file.
SETPFAC	Changes the security access category set of a permanent file.
SETPFAL	Changes the security access level of a 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.

On a secured system, your job may not access a permanent file unless the job is validated for the access level and access category set assigned to the file. For an alternate user access, such an access attempt is treated as a file-not-found error; for an access to one of your own files, it is treated as a security conflict.

In addition, your job may not change or access certain types of permanent file information. The following limitations apply:

- CHANGE, PERMIT, and SETPFAC requests are not allowed on files with an access level lower than that of the job.
- CATLIST requests for permit information are not allowed on files with an access level higher than that of the job. Normal CATLIST requests are allowed, but information on the password expiration date and access category set is not included in the CATLIST output. If a password exists, it is returned as seven asterisks (*****).

For more information on permanent files, refer to Permanent Files, Mass Storage File Residence, and Alternate Storage 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 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.

When a slash is used, it 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 permit category (CT), user name (UN), file access password (PW), security access level (AL), 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>
AL=level	<p>Specifies the security access level you want assigned to the file. Unless changed by your site, level can be one of the following names:</p> <p style="padding-left: 40px;">LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7</p> <p>When used with the DEFINE command, the default for this parameter is the current job access level. When used with the SAVE command, the default is the access level of the local file being saved.</p>

BR=br	<p>Specifies the backup copy requirement. You can request that two backup copies, one backup copy, or no backup copy be kept of your permanent file. Depending on your site's configuration and procedures, a backup copy can be one of the following:</p>
-------	--

- A cartridge alternate storage (MSE) copy.
- A tape alternate storage copy.
- A tape alternate storage secondary copy.
- A PFDUMP tape copy.

The following values can be specified for br:

<u>br</u>	<u>Backup Requirement</u>
Y	Two backup copies of the file must be kept.
MD	One backup copy of the file must be kept.
N	No backup copy of the file need be kept.

If BR=br is omitted, BR=Y is used by default. For more information, refer to Alternate Storage in section 2.

<u>Parameter</u>	<u>Description</u>
CT=ct	Specifies the file permit category. If CT=ct is omitted when the file is created, the file is private. The file category can be changed later.

<u>ct</u>	<u>Category</u>
P, PR, or PRIVATE	<p>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).</p> <p>For each permitted user, the system records the number of accesses made and the date and time of the last access by each user. This information may be displayed with the CATLIST(LO=FP) command.</p> <p>The system also records the total number of accesses by all users, the date and time of the last access by any user, and the date and time of the last modification by any user. This information may be displayed with the CATLIST(LO=F) command.</p>
S or SPRIV	<p>Semiprivate files; available for access by any user who knows the file name, user name, and password. The file owner specifies the default permitted mode of access for all users. The file owner may also specify different permissions for different users with the PERMIT command.</p> <p>For each user who accesses the file, the system records the user name, the number of accesses made, and the date and time of the last access. This information may be displayed with the CATLIST(LO=FP) command.</p> <p>The system also records the total number of accesses by all users, the date and time of the last access by any user, and the date and time of the last modification by any user. This information may be displayed with the CATLIST(LO=F) command.</p>

<u>Parameter</u>	<u>Description</u>	<u>Category</u>
<u>ct</u> PU or PUBLIC	<p>Public files; available for access by all users who know the file name, user name, and password. The file owner specifies the default permitted mode of access for all users. The file owner may also specify different permissions for different users with the PERMIT command.</p> <p>For each explicitly permitted user, the system records the number of accesses made and the date and time of the last access. This information may be displayed with the CATLIST(LO=FP) command.</p> <p>The system also records the total number of accesses by all users, the date and time of the last access by any user, and the date and time of the last modification by any user. This information may be displayed with the CATLIST(LO=F) command.</p>	
NA	<p>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.</p> <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. 	

You may not specify both the NA and WB parameters.

<u>Parameter</u>	<u>Description</u>												
PN=packname	<p>Specifies the 1- to 7-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.</p> <p>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.</p> <p>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.</p>												
PR=pr	<p>Specifies the preferred residence. If PR=pr is omitted, no preference is assumed. For more information, refer to Alternate Storage in section 2.</p> <table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>pr</u></th> <th style="text-align: left;"><u>Preference</u></th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">D</td> <td>You prefer that the file remain on disk between accesses. However, the system might move the file to alternate storage despite the specified preferred reference.</td> </tr> <tr> <td style="vertical-align: top;">L</td> <td>The file is locked to disk and will not be moved to alternate storage. Special validation is required to specify this value.</td> </tr> <tr> <td style="vertical-align: top;">M</td> <td>You prefer that the file be moved to alternate storage between accesses. However, the system might not move the file to alternate storage despite the specified preference.</td> </tr> <tr> <td style="vertical-align: top;">N</td> <td>You have no preference.</td> </tr> <tr> <td style="vertical-align: top;">T</td> <td>You prefer that the file be moved to tape alternate storage between accesses. However, the system might not move the file to tape alternate storage despite the specified preference.</td> </tr> </tbody> </table>	<u>pr</u>	<u>Preference</u>	D	You prefer that the file remain on disk between accesses. However, the system might move the file to alternate storage despite the specified preferred reference.	L	The file is locked to disk and will not be moved to alternate storage. Special validation is required to specify this value.	M	You prefer that the file be moved to alternate storage between accesses. However, the system might not move the file to alternate storage despite the specified preference.	N	You have no preference.	T	You prefer that the file be moved to tape alternate storage between accesses. However, the system might not move the file to tape alternate storage despite the specified preference.
<u>pr</u>	<u>Preference</u>												
D	You prefer that the file remain on disk between accesses. However, the system might move the file to alternate storage despite the specified preferred reference.												
L	The file is locked to disk and will not be moved to alternate storage. Special validation is required to specify this value.												
M	You prefer that the file be moved to alternate storage between accesses. However, the system might not move the file to alternate storage despite the specified preference.												
N	You have no preference.												
T	You prefer that the file be moved to tape alternate storage between accesses. However, the system might not move the file to tape alternate storage despite the specified preference.												
PW=password	<p>Specifies the 1- to 7-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.</p> <p>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.</p>												

<u>Parameter</u>	<u>Description</u>
R=r	Specifies the device type on which the file resides or is to reside.
<u>r</u>	<u>Device</u>
DBi	885-42 Disk Storage Subsystem (1<i<3; full-track).
DCi	895-1/2 Disk Storage Subsystem (1<i<2; full-track).
DDi	834 Disk Storage Subsystem (1<i<8; full-track).
DE	Extended memory.
DFi	887 Disk Storage Subsystem (1<i<3; full-track; 4K sector).
DGi	836 Disk Storage Subsystem (1<i<3; full-track).
DHi	887 Disk Storage Subsystem (1<i<2; full-track; 16K sector).
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	885-11/12 Disk Storage Subsystem (1<i<3; half-track).
DN	9853 Disk Storage Subsystem (full-track; 2K sector).
DQi	885-11/12 Disk Storage Subsystem (1<i<3; full-track).
DP†	Distributive data path to extended memory.
DV	819 Disk Storage Subsystem (single-density).
DW	819 Disk Storage Subsystem (double-density).

The value i denotes the number of units (spindles) required for a particular auxiliary disk file that you will access in your job.

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).

†The job must be of system origin or you must have system origin privileges.

<u>Parameter</u>	<u>Description</u>
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
FTNTE	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.

<u>Parameter</u>	<u>Description</u>
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.
XD=expiredate	<p>Specifies the expiration date for a file password or file permit. The value expiredate must be a six-digit string of the form yymmdd; where yy, mm, dd are two-digit year, month, and day designators, respectively. The latest date you can specify and the default expiration date are site-determined.</p> <p>You must have special authorization to specify an expiration date for a file password or permit (refer to the LIMITS command).</p>
XT=expireterm	<p>Specifies the life of a file password or permit in days. If you specify 0, the file password or permit expires immediately. If you specify 4095 or *, the file password or permit does not expire. The default expiration term and the maximum expiration term are site-determined (no site can choose a value greater than 4095, however).</p> <p>You must have special authorization to specify an expiration term for a file password or permit (refer to the LIMITS command).</p>

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,lf_n₁,lf_n₂,...,lf_n_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.
lf _n _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 a removable 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, ..., lfnn=pfnn /M=m, UN=username, PW=password,
PN=packname, R=r, NA, RT, WB.
```

<u>Parameter</u>	<u>Description</u>
lfn _i =pfn _i	Specifies that 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. 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).

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.																		
	<table border="0"> <tr> <td>If m is:</td> <td>You can:</td> <td>Concurrently, other users can:</td> </tr> <tr> <td>E or EXECUTE (execute mode)</td> <td>Execute the file.</td> <td>Read or execute the file.</td> </tr> <tr> <td>R or READ (read mode)</td> <td>Read or execute the file.</td> <td>Read or execute the file.</td> </tr> <tr> <td>RU or READUP† (read update mode)</td> <td>Read or execute the file</td> <td>Read or execute the file; one user can rewrite the file.</td> </tr> <tr> <td>RA or READAP† (read append mode)</td> <td>Read or execute the file.</td> <td>Read or execute the file; one user can lengthen the file.</td> </tr> <tr> <td>RM or READMD† (read modify mode)</td> <td>Read or execute the file.</td> <td>Read or execute the file; one user can lengthen or rewrite the file.</td> </tr> </table>	If m is:	You can:	Concurrently, other users can:	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.	RU or READUP† (read update mode)	Read or execute the file	Read or execute the file; one user can rewrite 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.
If m is:	You can:	Concurrently, other users can:																	
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.																	
RU or READUP† (read update mode)	Read or execute the file	Read or execute the file; one user can rewrite 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.																	

† Special care should be taken when using modes RM, RA, or RU. Some programs do not anticipate that one user may be updating a file while other users are reading the same file. Specifically, CRM/AAM (refer to the CYBER Record Manager Advanced Access Methods Reference Manual) is not designed to handle concurrent reading and updating. If a program using CRM/AAM attempts to access a file attached in RM, RA, or RU mode while a second program using CRM/AAM is accessing the file in M, A, or U mode, one or both programs may be aborted with an error message stating that the file was ruined. In fact, the file may still be intact, but CRM cannot guarantee this and does not allow the access.

<u>Parameter</u>	<u>Description</u>	
	If m is:	You can: Concurrently, other users can:
	U or UPDATE (update mode)	Read, execute, or rewrite the file. Read or execute 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 rewrite 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 update, 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 update, append, modify, or write mode, the current date is recorded as the last modification date even if the file is not altered.

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 a removable 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 file from alternate storage 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 alternate storage, job processing is suspended while the file is staged to disk and assigned to the job. For more information on alternate storage, refer to Alternate Storage in section 2.

† You cannot rewrite the file using the copying commands described in section 9. To rewrite the file while in modify or update mode, you must write your own program using the REWRITE macro (refer to Volume 4, Program Interface).

Parameter

Description

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 or WB is specified.

Assuming that no user has the file attached in append, update, modify, or write mode, 16380 users can attach the file concurrently (4095 in read mode, 4095 in read append mode, 4095 in read update 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 attaches the file in update mode, 8190 other users can attach the file (4095 in read update 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	Update	Read	Read Modify	Read Append	Read Update	Execute
Write	Busy	Busy	Busy	Busy	Busy	Busy	Busy	Busy	Busy
Modify	Busy	Busy	Busy	Busy	Busy	Read Modify	Busy	Busy	Busy
Append	Busy	Busy	Busy	Busy	Busy	Read Modify	Read Append	Busy	Busy
Update	Busy	Busy	Busy	Busy	Busy	Read Modify	Busy	Read Update	Busy
Read	Busy	Busy	Busy	Busy	Read	Read Modify	Read Append	Read Update	Execute
Read Modify	Busy	Modify	Append	Update	Read	Read Modify	Read Append	Read Update	Execute
Read Append	Busy	Busy	Append	Busy	Read	Read Modify	Read Append	Read Update	Execute
Read Update	Busy	Busy	Busy	Update	Read	Read Modify	Read Append	Read Update	Execute
Execute	Busy	Busy	Busy	Busy	Read	Read Modify	Read Append	Read Update	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. For interactive terminals operating in screen mode, the system automatically reformats CATLIST output according to the page length set for your terminal.

Format:

CATLIST, DN=dn, EF=ef, FN=fn, L=lfm, LO=option, NA, PN=packname, R=r, UN=username, WB.

<u>Parameter</u>	<u>Description</u>
DN=dn	Specifies the device number assigned to a device at system initialization. If specified, the CATLIST command lists catalog information from that device.
EF=ef	<p>Specifies one or more excluded permanent file names. This parameter excludes these permanent files from being listed.</p> <p>An asterisk (*) present in a file name indicates that a set of files is to be excluded. For example, EF=***OPL will exclude all the files with 6 character file names that end in OPL; EF=M***** will exclude all the files with file names that start with the letter M.</p> <p>Multiple file names may be specified, separated by slashes. For example, EF=A*****/B*****/C***** will exclude all the files with names that begin with the letters A, B, or C.</p> <p>The EF parameter may be specified together with the FN parameter. For example, FN=AB*****, EF=ABC***** will list information about all files with names that start with the letters AB, except for those files with names that start with the letters ABC.</p> <p>This parameter is not allowed when listing permit information (LO=FP, LO=P), nor when listing security access categories (LO=X).</p>

<u>Parameter</u>	<u>Description</u>
FN=fn	<p>Specifies one or more permanent file names. This parameter specifies that catalog information is to be listed only for these permanent files.</p> <p>An asterisk (*) present in a file name indicates that information for a set of files is to be listed. For example, FN=***OPL will list all the files with 6 character file names that end in OPL; FN=M***** will list all the files with file names that start with the letter M.</p> <p>Multiple file names may be specified, separated by slashes. For example, FN=A*****/B*****/C***** will list all the files with names that begin with the letters A, B, or C.</p> <p>The FN parameter may be specified together with the EF parameter. For example, FN=AB*****,EF=ABC**** will list information about all files with names that start with the letters AB, except for those files with names that start with the letters ABC.</p> <p>If a single file name with no asterisk (*) characters is specified together with LO=0, a message (fn FOUND.) indicates whether or not the specified file is found in the specified catalog. If such a file name is specified together with LO=P and UN=username, a message (username FOUND.) indicates whether or not permission to access that file has been granted to that user name.</p> <p>This parameter is required when listing permit information (LO=FP, LO=P) or security access categories (LO=X). When LO=FP, LO=P, or LO=X is specified, only a single file name can be specified and no asterisk (*) characters can be used.</p>
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>

<u>Parameter</u>	<u>Description</u>
LO=option	One of the following list options. If LO is omitted, LO=0 is assumed.

<u>option</u>	<u>Description</u>
F	<p>Lists detailed information about the file specified by the FN=fn parameter. If you do not specify an FN=fn parameter, the system lists information about each file in your catalog. An example of this list option is given following the parameter descriptions.</p> <p>If you specify an alternate user name (UN=username), you receive a listing of files you can access in the other user's catalog; however, the only files listed are those that the owner has explicitly permitted to be listed (using the AC parameter on a CHANGE, DEFINE, or SAVE command). The passwords for files in another user's catalog are not listed. Passwords must be obtained from the file owner. In addition, charge and project number information also is not listed in an alternate user CATLIST unless the calling user has system origin privileges.</p>
FP	<p>Lists the access permissions granted for the file specified on the FN=fn parameter. If a user name is also specified (UN=username), only the file permission granted to that user is listed.</p> <p>The user name, mode of permitted access, permission expiration date, number of accesses, and the date and time of the last access is listed for each user that has been granted explicit permission to access the file. For a semiprivate file, this information is also listed for all other users that have accessed the specified file. (User names are not recorded for accesses to public files.) An asterisk follows the user name/permission mode if explicit permission has been granted to this user.</p>

Parameter

Description

option

Description

P Lists the user names of users who have been granted explicit permission to access the specified file. For a semiprivate file, this option also lists the user names of all other users who have accessed the specified file. This option requires that a file name be specified (FN=fn).

X Lists the security access categories associated with the file specified on the FN=fn parameter as well as the information returned by the LO=F parameter. You must specify the FN=fn parameter if you specify LO=X.

For security reasons, the security access categories associated with a file will not be listed unless your current job security access level is at least as high as the current permanent file security access level of the file. Refer to the Security Features topic in section 3 of this manual for more information on security access levels and categories.

0 (zero) 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.

A file name surrounded by parentheses indicates that the file resides on alternate storage rather than on disk.

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.

- EOI was altered during mass storage recovery.
- Error exists in file data or permit entries.

To clear an error status flag, refer to the CHANGE command in this section.

<u>Parameter</u>	<u>Description</u>
NA	Specifies the no abort option. If NA is specified, processing errors do not terminate the job. Do not specify both NA and WB.
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).
UN=username	Specifies the user name. This parameter has two purposes: <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>
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs. Do not specify both NA and WB.

Example 1:

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

```

CATALOG OF STUDENT          FM/NOSCLSH yy/mm/dd. hh.mm.ss.

INDIRECT ACCESS FILES

ADD      (EXAM)  GRADES  ID      MODIFY2  (RESEQ)  XX
CAPITAL  FIND   *HEROFTN LIST    PRIME    T

DIRECT ACCESS FILES

(DIRFILE)  DRFILE TV
  11 INDIRECT ACCESS FILES ON DISK.  TOTAL PRUS =  14.
   2 DIRECT ACCESS FILES ON DISK.    TOTAL PRUS =   2.
   2 INDIRECTS ON ALTERNATE STORAGE. TOTAL PRUS =   2.
   1 DIRECT ON ALTERNATE STORAGE.    TOTAL PRUS =   2.

```

An asterisk preceding a file name indicates that the error flag is set. A file name surrounded by parentheses indicates that the file resides on alternate storage rather than on disk.

Example 2:

Listing of files that begin with the letters PROC in the catalog of FREE007. The command is entered by user FREE007 in the form: CATLIST,FN=PROC***.

```
CATALOG OF FREE007          FM/NOSCLSH yy/mm/dd. hh.mm.ss.
      FILE NAME PROC***

INDIRECT ACCESS FILES

PROCART  PROCFIL  PROC1    PROC1A

      4 INDIRECT ACCESS FILES ON DISK.      TOTAL PRUS =          15.
```

Example 3:

Listing of files with 6-character and 7-character file names that end in BIN in the catalog of XYZ3320. The command is entered by user XYZ3320 in the form: CATLIST,FN=***BIN/****BIN.

```
CATALOG OF XYZ3320          FM/NOSCLSH yy/mm/dd. hh.mm.ss.
      SELECTED FILES

DIRECT ACCESS FILES

CATLBIN  PFA5BIN  PFTBIN  PL14BIN  PURGBIN  QACBIN  SPFLBIN
MAGBIN   PFA6BIN  PL13BIN

      10 DIRECT ACCESS FILES ON DISK.      TOTAL PRUS =          390.
```

Example 4:

Listing of files with 6-character and 7-character file names that end in BIN, except for those files with names that start with PF, in the catalog of XYZ3320. The command is entered by user XYZ3320 in the form: CATLIST,FN=***BIN/****BIN,EF=PF*****.

```
CATALOG OF XYZ3320          FM/NOSCLSH yy/mm/dd. hh.mm.ss.
      SELECTED FILES

DIRECT ACCESS FILES

CATLBIN  MAGBIN  PL13BIN  PL14BIN  PURGBIN  QACBIN  SPFLBIN

      7 DIRECT ACCESS FILES ON DISK.      TOTAL PRUS =          269.
```

Example 5:

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

```
CATALOG OF STUDENT          FM/NOSCLSH yy/mm/dd. hh.mm.ss.
      FILE NAME PRIME
      USER NAMES
```

```
CMK2011  JLX2015  KXK4277
```

```
3 USERS
```

Example 6:

Listing of detailed information about files in the catalog of SMITH. The command is entered by user SMITH in the form: CATLIST,LO=F.

```
CATALOG OF SMITH          FM/NOSCLSH yy/mm/dd. hh.mm.ss. PAGE 1
```

```
FILE NAME  FILE TYPE      LENGTH DN CREATION  ACCESS  DATA MOD
PASSWORD  COUNT          PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
EXPIRES   LEVEL  PR BR AC RS
CHARGE NO. PROJECT NUMBER
```

```
1 CATX     IND. PRIVATE      1   84/04/19. 84/04/20. 84/04/19.
          1          READ   09.10.23. 08.30.16. 09.10.23.
          LVLO  N Y N D
          2222  PRJ2201
```

```
2 TESTLIB  DIR. PUBLIC      1 41 84/04/19. 84/04/19. 84/04/19.
          0          WRITE  09.11.05. 09.11.05. 09.11.05.
          LVLO  N Y Y D
          2222  PRJ2201
```

```
1 INDIRECT ACCESS FILE ON DISK.      TOTAL PRUS =      1.
1 DIRECT ACCESS FILE ON DISK.        TOTAL PRUS =      1.
```

Example 7:

Listing of detailed information about the file XFILE, as well as listing of the security access categories associated with the file XFILE. The command is entered by user MLSUSER in the form: CATLIST, FN=XFILE, LO=X.

```
CATALOG OF MLSUSER          FM/NOSCLSH yy/mm/dd. hh.mm.ss.

FILE NAME FILE TYPE      LENGTH DN CREATION ACCESS  DATA MOD
PASSWORD COUNT          PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
EXPIRES  LEVEL  PR BR AC RS
CHARGE NO. PROJECT NUMBER

1 XFILE  DIR. PRIVATE      212 * 88/02/25. 88/02/25. 88/02/25.
          45          READ      11.11.48. 11.11.48. 11.11.48.
          LVL3    N Y N D
          1280    73CW5770

FILE ACCESS CATEGORY SET

CAT00  CAT01  CAT02  CAT04  CAT06
```


Column headings are printed when the LO=F or LO=X parameter is specified. For CATLIST listing files not connected to a terminal, you can control the print density and page length using the SET command and the PL and PD symbolic names.

The following are the column headings and their meanings.

<u>Heading</u>	<u>Meaning</u>
FILE NAME	Permanent file name.
FILE TYPE	The permanent file type, direct access (DIR) or indirect access (IND), followed by the file permit category (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. If the system is a secured system and if the file has an access level higher than that of your job, a series of seven asterisks (*****) replaces the file password.
COUNT	The number of times the file has been accessed.
PERM.	Permission mode allowed the user. The entry is WRITE, MODIFY, APPEND, UPDATE, READ, READMD, READAP, READUP, EXECUTE, or NULL.
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 date and time in the following format. yy/mm/dd. hh.mm.ss.
ACCESS DATE/TIME	Date and time of the last access to the file.
DATA MOD DATE/TIME	Date and time of the last modification of the file data.
EXPIRES	The expiration date of the file password. It is not listed if the file does not have a password, the password does not expire, or if the file belongs to another user.
LEVEL	The security access level of the file. On a secured system, it is always listed if the file belongs to you. On an unsecured system, it is only listed if the level is not null (an assignment has been made) and the file belongs to you.
PR	Preferred residence for this file. The possible values are L, D, M, N, and T. For more information, refer to the description of the PR parameter at the beginning of this section.

<u>Heading</u>	<u>Meaning</u>
BR	Requested backup copy for this file. Y means tape backup, MD means alternate storage 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.
AC	Alternate CATLIST permission. Specifies whether alternate users may obtain information about the file using CATLIST. Y means that information on this file will appear in an alternate user's CATLIST display; N means that an alternate user cannot get information on this file.
RS	Actual residence of the file. The possible values are D, DC, DT, DCT, CT and T. If a D is present, a copy of the file exists on disk. If a C is present, a copy of the file exists on cartridge alternate storage (MSE). If a T is present, a copy of the file exists on tape alternate storage. For more information, refer to Alternate Storage in section 2.
CHARGE NO.	Charge number associated with the file.
PROJECT NUMBER	Project number associated with the file.

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,XD=expiredate,XT=expireterm,AC=ac,CP.

<u>Parameter</u>	<u>Description</u>
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 permit category for the file. Entries are P, PR, or PRIVATE for private; S or SPRIV for semiprivate; and PU or PUBLIC for public.
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.

<u>Parameter</u>	<u>Description</u>
BR=br	Specifies the new backup copy selection; entries are Y (tape), MD (alternate storage), and N (no backup).
PR=pr	Specifies the new preferred residence; entries are M (alternate storage) and N (no preference).
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.
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 alternate storage file errors.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.
XD=expiredate	Specifies the new expiration date for the file password. The value expiredate must be a six-digit string of the form yymmdd; where yy, mm, and dd are two-digit year, month, and day designators, respectively.
XT=expireterm	Specifies the new life of the file password in days. If you specify 0, the file password expires immediately. If you specify 4095 or *, the file password does not expire.
AC=ac	Specifies whether alternate users may obtain information about the file using CATLIST. Entries are Y (yes) and N (no). The default value is N.
CP	Specifies that the charge and project numbers for the file are to be replaced by the charge and project numbers currently in effect for your job.

COMMON COMMAND

The COMMON command accesses the system file SYSTEM.

Format:

COMMON,SYSTEM.

You must be authorized to access library files. If a file with the name SYSTEM 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.

DEFINE COMMAND

The DEFINE command creates an empty direct access permanent file. It can also change a file of the local file type into a direct access file. In this case, the file must reside on a device on which you can place direct access files (refer to secondary masks and MSAL CMRDECK entries in the NOS 2 Analysis Handbook or contact site personnel). When you create a file using the DEFINE command, the file is automatically saved under the charge number and project number currently in effect for your job.

Format:

```
DEFINE, lfn1=pfn1, lfn2=pfn2, ..., lfnn=pfnn / PW=password, CT=ct, M=m, BR=br, PR=pr,  
PN=packname, R=r, S=space, NA, WB, AL=level, XD=expiredate, XT=expireterm, AC=ac.
```

<u>Parameter</u>	<u>Description</u>
lfn _i =pfn _i	The DEFINE command creates an empty direct access permanent file if pfn _i and lfn _i do not exist. If lfn _i is omitted, pfn _i is assumed to be the local file name and the permanent file name. Each file name can be from one to seven characters. 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.

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 1- to 7-character password that others must specify to access the file.
CT=ct	Specifies the permit 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. Other users can attach the file in the following modes (refer to the ATTACH command description): If m is: E or EXECUTE Execute. R or READ Read or execute.

<u>Parameter</u>	<u>Description</u>
RU or READUP†	Read update, read, or execute.
RA or READAP†	Read append, read, or execute.
RM or READMD†	Read modify, read append, read update, read, or execute.
U or UPDATE	Update.
A or APPEND	Append.
M or MODIFY	Modify, append, update, read modify, read append, read update, read, or execute.
W or WRITE	Write, modify, append, update, read modify, read append, read update, read, or execute.
N or NULL	None.
	After a file is defined, it is always assigned to the job in write mode.
BR=br	Specifies the backup copy requirement; Possible values are MD (one backup copy required), N (no backup required), or Y (two backup copies required). If BR=br is omitted, BR=Y is assumed.
PR=pr	Specifies the preferred file residence; possible values are D (disk residence preferred), L (locked to disk), M (alternate storage residence preferred), N (no preference), or T (tape alternate storage residence preferred). If PR=pr is omitted, PR=N 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.
R=r	Specifies the device type on which the permanent file is to reside. Specify this parameter if a removable auxiliary device with a device type other than the installation-defined default is used. 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 invalid device type 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.

†Special care should be taken when using modes RM, RA, or RU. Some programs do not anticipate that one user may be updating a file while other users are reading the same file. Specifically, CRM/AAM (refer to the CYBER Record Manager Advanced Access Methods Reference Manual) is not designed to handle concurrent reading and updating. If a program using CRM/AAM attempts to access a file attached in RM, RA, or RU mode while a second program using CRM/AAM is accessing the file in M, A, or U mode, one or both programs may be aborted with an error message stating that the file was ruined. In fact, the file may still be intact, but CRM cannot guarantee this and does not allow the access.

<u>Parameter</u>	<u>Description</u>
	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.
AL=level	Specifies the security access level you want assigned to the file. Unless changed by your site, level can be one of the following names: LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7 The default for this parameter is the current job access level.
XD=expiredate	Specifies the expiration date for the file password. The value expiredate must be a six-digit string of the form yymmdd; where yy, mm, and dd are two-digit year, month, and day designators, respectively. You can specify this parameter only if you also specify PW=password.
XT=expireterm	Specifies the life of the file password in days. If you specify 0, the file password expires immediately. If you specify 4095 or *, the file password does not expire. You can specify this parameter only if you also specify PW=password.
AC=ac	Specifies whether alternate users may obtain information about the file using CATLIST. Entries are Y (yes) and N (no). The default is N.

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.

DROPDS COMMAND

The DROPDS command releases the disk space associated with one or more of your permanent files, provided that a valid alternate storage copy (on cartridge (MSE) or tape alternate storage) exists for each file. The local file copies of these files (if they are present) are also returned by this command.

Format:

DROPDS, lfn₁=pfn₁, lfn₂=pfn₂, ..., lfn_n=pfn_n / PN=packname, R=r, NA, WB.

<u>Parameter</u>	<u>Description</u>
lfn _i =pfn _i	Specifies that the permanent file disk space associated with the file named pfn _i should be released. The local file named lfn _i will be returned at the same time.
	If lfn _i is omitted, the name of the local file is assumed to be pfn _i .

The full descriptions of the following optional parameters are given at the beginning of this section.

<u>Parameter</u>	<u>Description</u>
PN=packname	Specifies the auxiliary device name. Specified if the permanent file to be released resides on an auxiliary device.
R=r	Specifies the device type. Specified if a removable 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 abort the job.
WB	Specifies the wait-if-busy option. The job waits for the mounting of removable auxiliary disk packs.

FTP COMMAND

The File Transfer Protocol (FTP) is a DDN-defined protocol that provides file transfer capabilities between connected TCP/IP hosts. FTP allows you to access directories/files on a remote host and to perform common operations. Such operations include list and change working directories, list files at various levels, and rename directories and files.

Format:

FTP,I=input,O=output,P=ftpprlg.

or

CRECFC,I=input,O=output,P=ftpprlg.

<u>Parameter</u>	<u>Description</u>
I=input	Specifies the local file which contains the directives that FTP sends to the remote host. The file name must be from 1- to 7-alphanumeric characters. The system reads directives from the current position on the file to the next EOR. If you omit the I=input parameter, the default is to use local file INPUT.
O=output	Specifies the local file to receive the output generated by FTP. The file name must be from 1- to 7-alphanumeric characters. If you omit the O=output parameter, the default is to use local file OUTPUT. If you specify O=0, then no output file is created.
P=ftpprlg	Specifies the prolog file. Values specified in the prolog file take precedence over those specified by FTP. The file name must be from 1- to 7-alphanumeric characters. Before being read, the file is rewound and cannot be assigned to a terminal. If the file is not local to the job, FTP attempts to acquire a permanent file of the same name. If the file is acquired, it is returned after prolog file processing is completed. If omitted, the default is to use file FTPPRLG. If you specify P=0, no prolog file is used.

FTP DIRECTIVES

The action taken by FTP depends on the directives you specify. Alternatively, you can specify a directives file with the I=input parameter.

For more information on the FTP command, refer to the CDCNET TCP/IP Usage manual.

APPEND_FILE Directive

The APPEND_FILE directive appends data from a local host file to a corresponding remote host file. If the remote host file does not exist, then one is created by the remote host's server.

NOTE

When you are specifying the embedded parameters (CS, FSC, and READEOI) found on the LOCAL_FILE parameter, and the destination host is a non-NOS system, it is necessary to explicitly specify the REMOTE_FILE parameter. Allowing the embedded parameters to default without specifying the REMOTE_FILE parameter may result in a file name error. The following example appends file PROCL from NOS to file PROCL on NOS/VE:

```
APPEND_FILE 'PROCL,CS=A,FSC=1,READEOI' PROCL
```

Format:

```
APPEND_FILE or  
APPF or  
APPEND or  
APPE  
LOCAL_FILE=local_name or  
LOCAL_FILE='local_name,CS=cset,FSC=n,READEOI'  
REMOTE_FILE=local_name
```

<u>Parameter</u>	<u>Description</u>
LOCAL_FILE or Name of the local host file. LF	
CS	Specifies the code set, data character size and format of the file on the NOS host. Files transmitted by FTP/NOS are mapped to network ASCII from the specified representation. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command, is ASCII (A).

The following NOS FCOPY command compatible values are supported:

<u>Value</u>	<u>Description</u>
ASCII or A	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 or 8	ASCII 8/12 file with TELNET end-of-line terminators.
ASCII88 or 88	ASCII 8/8 file with TELNET end-of-line terminators.
DIS or D	Display-coded file with zero-byte-terminated records.

If this parameter is omitted, the default is ASCII (A).

<u>Parameter</u>	<u>Description</u>
FSC	Specifies the number of files to be skipped before starting data transfer to the remote host. If this parameter is specified, n NOS end-of-file (EOF) marks are skipped before data transfer begins. Data transfer begins at file n+1 of a multifile set (BOI) of the file. If this parameter is omitted, the default is zero. Data transfer starts from the beginning-of-information (BOI) of the file.
READEOI	Specifies read to end-of-information (EOI). If this parameter is specified, all files of a multifile set from the current file position to the EOI of the file are transferred with NOS EOF marks ignored.
REMOTE_FILE	Specifies the name of the remote file; same name as specified by the LOCAL_FILE or LF parameter. This parameter is required if specifying the embedded parameters CS, FSC, and READEOI, and the remote host is a non-NOS system.

CHANGE_AUTO_LOGIN_MODE Directive

The CHANGE_AUTO_LOGIN_MODE directive controls automatic login operations when a connection to a remote host has been established.

Format:

```
CHANGE_AUTO_LOGIN_MODE or
CHAALM
MODE=boolean
```

<u>Parameter</u>	<u>Description</u>
MODE or M	Setting the MODE parameter to ON, TRUE, or YES enables auto login mode. Setting the MODE parameter to OFF, FALSE, or NO disables auto login mode. The default is ON.

CHANGE_BELL_MODE Directive

The CHANGE_BELL_MODE directive controls the sending by FTP of an ASCII BEL character to interactive terminals with the input prompt.

Format:

```
CHANGE_BELL_MODE or
CHABM or
BELL
MODE=boolean
```

<u>Parameter</u>	<u>Description</u>
MODE or M	Setting the MODE parameter to ON, TRUE, or YES enables bell mode. Setting the MODE parameter to OFF, FALSE, or NO disables bell mode. The default is OFF.

CHANGE_DEBUG_MODE Directive

The CHANGE_DEBUG_MODE directive controls the display of debug mode output.

Format:

CHANGE_DEBUG_MODE or
CHADM or
DEBUG
MODE=boolean

<u>Parameter</u>	<u>Description</u>
MODE or	Setting the MODE parameter to ON, TRUE, or YES enables debug mode.
M	Setting the MODE parameter to OFF, FALSE, or NO disables debug mode. The default is OFF.

CHANGE_FILE_NAME Directive

The CHANGE_FILE_NAME directive renames a file on the remote host.

Format:

CHANGE_FILE_NAME or
CHAFN or
RENAME
FILE=current file name
NEW_FILE_NAME=new file name

<u>Parameter</u>	<u>Description</u>
FILE or	Name of a remote host file.
F	
NEW_FILE_NAME	Specifies a new name for the remote host file.
or NFN	

CHANGE_INPUT_FILE Directive

The CHANGE_INPUT_FILE directive allows the specification of a new default input file.

Format:

CHANGE_INPUT_FILE or
CHAIF
INPUT=file

<u>Parameter</u>	<u>Description</u>
INPUT or	Specifies the input source file. The default is INPUT.
I	

CHANGE_OUTPUT_FILE Directive

The CHANGE_OUTPUT_FILE directive allows the specification of a new output file.

Format:

```
CHANGE_OUTPUT_FILE or  
CHAOF  
OUTPUT=file
```

<u>Parameter</u>	<u>Description</u>
OUTPUT or 0	Specifies the new output file. The current file is flushed by writing an end-of-record (EOR) to it. The default is OUTPUT.

CHANGE_PORT_COMMAND_MODE Directive

The CHANGE_PORT_COMMAND_MODE directive specifies if the FTP PORT command should be sent to the remote host prior to a data transfer.

Format:

```
CHANGE_PORT_COMMAND_MODE or  
CHAPCM  
MODE=boolean
```

<u>Parameter</u>	<u>Description</u>
MODE or M	Setting the MODE parameter to ON, TRUE, or YES enables port command mode. Setting the MODE parameter to OFF, FALSE, or NO disables port command mode. The default is ON.

CHANGE_TRANSFER_TYPE Directive

The CHANGE_TRANSFER_TYPE directive defines to FTP the data of the file being transferred. When an FTP session is started, the default transfer type is ASCII.

Format:

```
CHANGE_TRANSFER_TYPE or  
CHATT or  
TYPE  
TYPE=keyword
```

<u>Parameter</u>	<u>Description</u>
TYPE	Specifies the data type of the file to be transferred.

<u>keyword</u>	<u>Description</u>
ASCII or A	ASCII 6/12 file with zero-byte-terminated records.
BINARY or B or IMAGE or I	A mass storage file containing binary data, represented in 8-bit bytes.

The default is ASCII.

CHANGE_VERBOSE_MODE Directive

The CHANGE_VERBOSE_MODE directive controls the display status of FTP processing. For more information, refer to the FTP directive DISPLAY_LOCAL_HELP.

Format:

```
CHANGE_VERBOSE_MODE or  
CHAVM or  
VERBOSE  
MODE=boolean
```

<u>Parameter</u>	<u>Description</u>
MODE or M	Setting the MODE parameter to ON, TRUE, or YES enables VERBOSE_MODE. Setting the MODE parameter to OFF, FALSE, or NO disables VERBOSE_MODE. The default is OFF.

CREATE_HOST_CONNECTION Directive

The CREATE_HOST_CONNECTION directive establishes a connection with a remote host. An existing host connection must be deleted with the DELETE_HOST_CONNECTION directive before a new connection to another host can be made.

Format:

```
CHANGE_HOST_CONNECTION or  
CREHC or  
OPEN  
HOST=name  
AUTO_LOGIN_MODE=boolean
```

<u>Parameter</u>	<u>Description</u>
HOST or H	The name of the remote host where a connection is to be established.
AUTO_LOGIN_ MODE or ALM	Specifies whether an automatic login attempt should be made when a connection to a host has been established. Setting AUTO_LOGIN_MODE to ON, TRUE, or YES enables AUTO_LOGIN_MODE. Setting AUTO_LOGIN_MODE to OFF, FALSE, or NO disables AUTO_LOGIN_MODE. If automatic login is enabled and no login information for the specified host has been made available by the DEFINE_AUTO_LOGIN directive, then a prompt is issued for user name, password, and if required, account. Automatic login information is established with the DEFINE_AUTO_LOGIN directive. For more information, refer to the DEFINE_AUTO_LOGIN directive. If this parameter is omitted, the default is ON.

DEFINE_AUTO_LOGIN Directive

The DEFINE_AUTO_LOGIN directive associates login information with a host name, so when a connection to that host is established, the user can be automatically logged in.

Format:

```
DEFINE_AUTO_LOGIN or
DEFAL
  HOST=name
  USER=username
  PASSWORD=password
  FAMILY=family
  ACCOUNT=account
  PROJECT=project
```

<u>Parameter</u>	<u>Description</u>
HOST or H	Name of the remote host associated with the login information.
USER or U	The login user name.
PASSWORD or PW	Password associated with the login user name.
FAMILY or F	Family name associated with the login user name if different from the default family name used on the remote host.
ACCOUNT or A	Account associated with the login. This parameter specifies the nondefault account to which the resource usage on the remote host is charged.
PROJECT or P	Project associated with the login. This parameter specifies the nondefault project to which the resource usage on the remote is charged. If this parameter is omitted, the default established by the server's host administrator is used.

DELETE_FILE Directive

The DELETE_FILE directive deletes a file on the remote host.

Format:

```
DELETE_FILE or
DELF or
DELETE or
DELE
  NAME=name
```

<u>Parameter</u>	<u>Description</u>
NAME or N	Name of the remote host file to be deleted.

DELETE_HOST_CONNECTION Directive

The DELETE_HOST_CONNECTION directive terminates an existing connection with a remote host without terminating FTP.

Format:

DELETE_HOST_CONNECTION or
DELHC or
CLOSE

DISPLAY_COMMAND_INFORMATION Directive

The DISPLAY_COMMAND_INFORMATION directive displays information regarding a specific FTP directive.

Format:

DISPLAY_COMMAND_INFORMATION or
DISCI
COMMAND=FTP command

<u>Parameter</u>	<u>Description</u>
COMMAND	Name of an FTP directive for which information is required.

DISPLAY_DIRECTORY Directive

The DISPLAY_DIRECTORY directive displays a detailed list of files in the specified remote directories. The default is to display the current directory. If the name specified is a file name on the remote server host, current information on the file is displayed. To display the local directory, use the NOS/VE command DISPLAY_CATALOG or the NOS command CATLIST.

Format:

DISPLAY_DIRECTORY or
DISD or
DIR or LIST
NAME=name
OUTPUT=local_name

<u>Parameter</u>	<u>Description</u>
NAME or N	Remote host file name.
OUTPUT or O	Allows the specification of an output file other than the default. The default is OUTPUT.

DISPLAY_FILE_NAMES Directive

The DISPLAY_FILE_NAMES directive displays a list of file names in a remote directory.

Format:

DISPLAY_FILE_NAMES or
DISFN or
LS or
NLST

NAME=name
OUTPUT=local_name

<u>Parameter</u>	<u>Description</u>
NAME or N	Remote host file name.
OUTPUT or O	Allows the specification of an output file other than the default. The default is OUTPUT.

DISPLAY_FTP_OPTIONS Directive

The DISPLAY_FTP_OPTIONS directive allows the user to obtain current information about the existing FTP connection.

Format:

DISPLAY_FTP_OPTION or
DISFO

DISPLAY_LOCAL_HELP Directive

The DISPLAY_LOCAL_HELP directive provides information about each command. The output contains the FTP command, a brief description of the command, and the aliases of the command.

Format:

DISPLAY_LOCAL_HELP or
DISLH or
HELP

FTP/NOS directly supports the DISPLAY_COMMAND_INFORMATION directive to display parameter information for a specific directive.

DISPLAY_REMOTE_HELP Directive

The DISPLAY_REMOTE_HELP directive provides help information from the remote FTP server.

Format:

```
DISPLAY_REMOTE_HELP or  
DISRH or  
REMOTEHELP  
COMMAND=ftp_command
```

<u>Parameter</u>	<u>Description</u>
COMMAND or C	Identifies a specific FTP command for which a user wants help.

GET_FILE Directive

The GET_FILE directive retrieves a file from the remote host.

NOTE

When specifying the embedded parameters (CS and TRUNC) found on the LOCAL_FILE parameter, and the destination host is a non-NOS system, it is necessary to explicitly specify the REMOTE_FILE parameter. Allowing the embedded parameters to default without specifying the REMOTE_FILE parameter may result in a file name error.

The following example gets a file named PROCL on NOS/VE and makes it a local file on NOS.

```
GET_FILE PROCL `PROCL,CS=A,`
```

Format:

GET_FILE or
GETF or
GET or
RECV or
RETR

REMOTE_FILE=name
LOCAL_FILE=local_name or
LOCAL_FILE="local_name,CS=cset,TRUNC"

Parameter

Description

REMOTE_FILE Specifies the name of the remote file; same name as specified by the LOCAL_FILE or LF parameter. This parameter is required if specifying the embedded parameters CS and TRUNC, and the remote host is a non-NOS system.

LOCAL_FILE or Name of a local host file.

LF

CS Specifies the code set, data character size, and format of the file on the NOS host. Files transmitted by FTP/NOS are mapped to network ASCII from the specified representation. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command, is ASCII (A).

The following NOS FCOPY command compatible values are supported:

<u>Value</u>	<u>Description</u>
ASCII or A	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 or 8	ASCII 8/12 file with TELNET end-of-line terminators.
ASCII88 or 88	ASCII 8/8 file with TELNET end-of-line terminators.
DIS or D	Display-coded file with zero-byte-terminated records.

If this parameter is omitted, the default is ASCII (A).

TRUNC

Specifies the deletion of pad bits from the file being received by FTP/NOS. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command, is IMAGE (I).

LOGIN_USER Directive

The LOGIN_USER directive sends login information to the remote host after a connection has been established.

LOGIN_USER or
LOGU or
USER

USER=username
PASSWORD=password
FAMILY=family
ACCOUNT=account
PROJECT=project

<u>Parameter</u>	<u>Description</u>
USER or U	Login user name.
PASSWORD or PW	Password associated with the login user name.
FAMILY or F	Family name associated with the login user name if it is other than the default family for the remote host. The FAMILY parameter value is sent to the remote host in the PASS FTP protocol element (FTP command) by prefixing it with a comma acting as a delimiter and suffixing the result string to the PASSWORD parameter value. The FAMILY parameter is only applicable if the remote host is a CYBER host.
ACCOUNT or A	Account associated with the login. This parameter specifies the nondefault account to which the resource usage on the remote is charged.
PROJECT or P	Project associated with the login. This parameter specifies the nondefault to which the resource usage on the remote host is charged. If this parameter is omitted, the default established by the server's host administrator is used. This parameter is only valid if the remote host is a CYBER host.

QUIT Directive

The QUIT directive terminates execution of FTP.

Format:

QUIT or
QUI or
BYE or Q
ABORT=boolean

<u>Parameter</u>	<u>Description</u>
ABORT	Specifies whether to abort the current connection before terminating the execution of FTP. Setting the ABORT parameter to ON, TRUE, or YES causes FTP to abort the current connection before terminating; setting the ABORT parameter to OFF, FALSE, or NO causes FTP to close any existing connection before terminating. If this parameter is omitted, by default FTP tries to close any existing connection before terminating.

REPLACE_FILE Directive

The REPLACE_FILE directive sends data from a file on the local host to a file on the remote host. If the file does not exist on the remote host, it is created.

NOTE

When specifying the embedded parameters (CS, FSC, and READEOI) found on the LOCAL_FILE parameter, and the destination host is a non-NOS system, it is necessary to explicitly specify the REMOTE_FILE parameter. Allowing the embedded parameters to default without specifying the REMOTE_FILE parameter may result in a file name error.

The following example replaces file PROCL on NOS/VE with file PROCL on NOS.

```
REPLACE_FILE 'PROCL,CS=A,FSC=1,READEOI' PROCL
```

Format:

REPLACE_FILE or
REPF or
PUT or
SEND or
STOR

LOCAL_FILE=local_name or
LOCAL_FILE='local_name, CS=cset, FSC=n, READEOI'
REMOTE_FILE=name

<u>Parameter</u>	<u>Description</u>
LOCAL_FILE or Name of a local host file. LF	
CS	Specifies the code set, data character size, and format of the file on the NOS host. Files transmitted by FTP/NOS are mapped to network ASCII from the specified representation. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command, is ASCII (A).

The following NOS FCOPY command compatible values are supported:

<u>Value</u>	<u>Description</u>
ASCII or A	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 or 8	ASCII 8/12 file with TELNET end-of-line terminators.

<u>Parameter</u>	<u>Description</u>						
	<table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>ASCII88 or 88</td> <td>ASCII 8/8 file with TELNET end-of-line terminators.</td> </tr> <tr> <td>DIS or D</td> <td>Display-coded file with zero-byte-terminated records.</td> </tr> </tbody> </table> <p>If this parameter is omitted, the default is ASCII (A).</p>	<u>Value</u>	<u>Description</u>	ASCII88 or 88	ASCII 8/8 file with TELNET end-of-line terminators.	DIS or D	Display-coded file with zero-byte-terminated records.
<u>Value</u>	<u>Description</u>						
ASCII88 or 88	ASCII 8/8 file with TELNET end-of-line terminators.						
DIS or D	Display-coded file with zero-byte-terminated records.						
FSC	Specifies the number of files to be skipped before starting data transfer to the remote host. If this parameter is specified, n NOS end-of-file (EOF) marks are skipped before data transfer begins. Data transfer begins at file n+1 of a multifile set (BOI) of the file.						
READEOI	Specifies read to end-of-information (EOI). If this parameter is specified, all files of a multifile set from the current file position to the EOI of the file are transferred with NOS EOF marks ignored.						
REMOTE_FILE	Specifies the name of the remote file; same name as specified by the LOCAL_FILE or LF parameter. This parameter is required if specifying the embedded parameters CS, FSC, and READEOI, and the remote host is a non-NOS system.						

SEND_FTP_COMMAND Directive

The SEND_FTP_COMMAND directive allows the sophisticated FTP user to directly send an FTP command to a remote host. It receives a single reply from the remote server. This command is used for debugging purposes by users who are familiar with the FTP protocol. This command does not work for FTP commands that initiate a data transfer (LIST, NLST, STOR, RETR, and APPE).

Format:

```
SEND_FTP_COMMAND or
SENFEC or
QUOTE
COMMAND=ftp_command
PARAMETERS=command_parameters
```

<u>Parameter</u>	<u>Description</u>
COMMAND	Sends 3- or 4-character FTP commands.
PARAMETERS	Sends additional parameters required for the given FTP command.

GET COMMAND

The GET command retrieves copies of indirect access permanent files for use as local files.

Format:

GET, lfn₁=pfn₁, lfn₂=pfn₂, ..., lfn_n=pfn_n / UN=username, PW=password, PN=packname,
R=r, NA, RT, WB.

<u>Parameter</u>	<u>Description</u>
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.
RT	Specifies the real-time processing option. If RT is specified, the job continues processing after requesting staging of a file from alternate storage 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 GET command to assign the file to the job. If RT is omitted and the file resides only on alternate storage, job processing is suspended while the file is staged to disk and assigned to the job. For more information on alternate storage, refer to Alternate Storage in section 2.
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.

MFLINK COMMAND

The MFLINK command optionally transfers a file to or from another mainframe or creates, purges, or otherwise changes the attributes of a permanent file on another mainframe. You must be authorized to use the MFLINK command (refer to the LIMITS command).

Format:

MFLINK,lfn,ST=lid,I=dirfile,DD=dd,EP,RT,PC=c.

Parameter lfn is order-dependent, it must be the first parameter. If you omit lfn, you must have two consecutive delimiters after MFLINK. The other parameters are order-independent.

<u>Parameter</u>	<u>Description</u>
lfn	Specifies the optional local file name to be used in any file transfers. The file name must be one to seven alphanumeric characters and lfn must reside on disk. If the transfer is from the local host to the remote host, lfn must be a local file before you enter the MFLINK command. If the transfer is from the remote host to the local host, MFLINK either writes over the existing lfn, or if there is no local file, MFLINK creates a new file with the name lfn. If you omit lfn, but subsequent directives call for a file transfer, MFLINK uses LFILE as the default lfn. Some directives, such as CHANGE, PERMIT, and PURGE (for a NOS remote host) do not require an lfn, because they do not cause a file transfer. MFLINK rewinds lfn before and after the transfer.
ST=lid	Specifies the logical identifier (LID) of the remote host to which MFLINK is to send the directives record. The LID must be a three-character string defined by your site. You must specify the ST parameter on the first, and only the first, MFLINK command of a series of MFLINK commands (an MFLINK session) that are for the same remote host. The ST=lid specification remains in effect during the entire session. On subsequent MFLINK commands in the same session, any other parameters not specified revert to their default values. The occurrence of the ST=lid parameter on an MFLINK command initiates a new MFLINK session with the specified remote host. Once you have entered an MFLINK command with an ST=lid parameter, you can resume that MFLINK session at any time during your job by entering an MFLINK command without an ST=lid parameter.

Parameter

Description

I=dirfile Specifies the local file which contains the directives that the system sends to the remote host. The file name must be one to seven alphanumeric characters. The system reads directives from the current position on the file to the next EOR. If you specify only the keyword I, file INPUT becomes the directives file. If you omit the I=dirfile parameter, the system reads directives from the lines immediately following the MFLINK command. In this case, the directives must all begin with a prefix character and the system reads directives until it comes to a line without the prefix character.

DD=dd Specifies the data format of the file to be transferred. The DD=dd parameter both describes the character set of lfn on the host where it currently resides and also specifies the character set lfn is to have after the transfer.

For file transfers between NOS mainframes and between NOS/BE and NOS mainframes this parameter is effectively ignored. The file will have the same format on both mainframes.

The interpretation of the DD=dd parameter depends on the operating system(s) involved in the file transfer. The following paragraphs describe how NOS interprets the DD=dd parameter. For its interpretation by other operating systems, refer to the Remote Host Facility Usage manual.

On NOS, you can specify the following values for dd:

<u>dd</u>	<u>Description</u>
C6	The file contains character data in a 64-character (or less) subset of the ASCII 128-character set. The character code of lfn is 6-bit display code and its lines are zero-byte terminated.
C8	The file contains character data in a larger than 64-character subset of the ASCII 128-character set. The character code of lfn is the 7-bit ASCII code (right-justified in 12-bit fields) and its lines are zero-byte terminated.
US	Specifies that the file contains binary data in structured format with EORs and EOFs. The file is transmitted as a continuous string of binary data, broken only by EORs and EOFs as appropriate.
UU	Specifies that the file contains binary data without EORs and EOFs. NOS treats UU as US. If you send a file to a non-NOS or non-NOS/BE remote host with DD=UU specified and the file contains EORs or EOFs, the remote host may ignore them.

<u>Parameter</u>	<u>Description</u>
	If the file is character data, you do not need to specify a DD=dd parameter. The remote host determines the format in which to read/write the file. If you transfer a binary file to or from a mainframe with a different word size than the model 170, the mainframe may add extra bits at EORs or EOFs.
EP	Specifies the error processing that ensues if network problems cause a loss of the connection during a file transfer. If you omit the parameter EP, the system retries the request. The EP parameter inhibits this retry. The system attempts to correct data transmission errors, regardless of the EP parameter.
RT	Specifies the real-time action the system is to take when network resources are temporarily unavailable. If you specify RT and some network resource necessary for you to connect to a remote host is unavailable, MFLINK aborts without any retries. If you do not specify RT, after an installation-defined period of time MFLINK retries until the connection is successful.
PC=c	Specifies the prefix character for MFLINK directives. The default prefix character is an asterisk (*). The prefix character can be any 6-bit display code character except a closing parenthesis, a dollar sign, a period, comma, or a colon. To avoid confusion between commands and routing directives, use caution in selecting alphabetic characters for the prefix character.

MFLINK DIRECTIVES

The action taken by MFLINK depends on the directives you specify. You can specify directives immediately following the MFLINK command if you prefix each directive with the prefix character. The system discards the prefix characters before forwarding the directives to the remote host. Alternatively, you can specify a directives file with the I=dirfile parameter. The directives in the directives file do not require the prefix character. The directives must be commands recognized by the remote host; for example, if the remote host is a NOS/BE system, your directives must be NOS/BE commands. NOS allows only a small subset of its commands to be used as MFLINK directives and, further, may require some directives to have a slightly different form or function from their command counterparts.

This section documents only those MFLINK directives valid for a NOS remote host. For other remote hosts, refer to the Remote Host Facility Usage manual.

NOS, as a remote host, processes directives of up to 80 characters. A directive longer than 80 characters terminates the processing of the current directives record. A directive can span more than one line as long as it does not exceed the 80-character limit. You indicate that the next line is a continuation line by ending the current line with a comma.

The MFLINK directives fall into the three categories of recovery directives, file transfer directives, and file alteration directives as follows:

<u>Recovery</u>	<u>File Transfer</u>	<u>File Alteration</u>
USER	APPEND	CHANGE
CHARGE	ATTACH	PERMIT
	DEFINE	PURGE
	GET	PACKNAM
	REPLACE	
	SAVE	

Recovery directives are directives that a host must send a remote host any time you initiate or resume an MFLINK session. File transfer directives are directives that transfer a file from one host to another. There can be only one file transfer directive in a given directives record.

The directives operate in a manner similar to that of their NOS command counterparts. One exception is the function of the NA keyword.

The NA option on an MFLINK directive works in a manner very similar to the NA option on NOS permanent file commands. The main difference is that the system does not suspend MFLINK processing indefinitely while it waits for a temporary error condition to clear. If APLO (removable pack option) is set to NONZERO (refer to the description of COMSPFM parameters in the Installation Handbook) and the temporary error condition persists beyond an installation-defined timeout period, MFLINK issues a message indicating that the timeout occurred and then continues. If APLO is set to ZERO, MFLINK will not wait for a pack to be mounted but will continue immediately.

The following directive descriptions show the format of each directive and describe the function of each directive whose function differs from that of its NOS command counterpart.

APPEND Directive

The APPEND directive adds the file specified on the MFLINK command (lfn) to the end of pfn.

Format:

APPEND,pfn/UN=username,PW=password,PN=packname,R=r,NA,WB.

ATTACH Directive

The ATTACH directive retrieves a copy of pfn from the remote host and makes that copy a temporary file with the name specified on the MFLINK command. No interlock is maintained between lfn and pfn, except during the file transfer (as if M=READ had been specified).

Format:

ATTACH,pfn/UN=username,PW=password,PN=packname,R=r,NA,RT,WB.

CHANGE Directive

The CHANGE directive operates the same as the NOS command.

Format:

CHANGE,nfn=ofn/PW=password,CT=ct,M=m,BR=br,PR=pr,PN=packname,
R=r,NA,CE,WB,XD=expiredate,XT=expireterm,AC=ac,CP.

CHARGE Directive

The CHARGE directive operates the same as its NOS command counterpart, including the handling of default charges.

Format:

CHARGE,chargenumber,projectnumber.

If the CHARGE directive is required, it must immediately follow the USER directive.

DEFINE Directive

The DEFINE directive creates a direct access file named pfn on the remote host and copies onto it the file specified by lfn on the MFLINK command. There is no provision to copy over an existing direct access file; you must purge the existing file. No interlock is maintained between lfn and pfn, except during the file transfer (as if M=READ had been specified).

Format:

DEFINE,pfn/PW=password,CT=c,M=m,BR=br,PR=pr,PN=packname,R=r,S=space,NA,WB,
AL=level,XD=expiredate,XT=expireterm,AC=ac.

DROPDS Directive

The DROPDS directive releases the current copy of a permanent file from mass storage when a backup copy has been saved on a storage device.

Format:

DROPDS,pfn/PN=packname,R=r,NA,WB.

GET Directive

The GET directive retrieves a copy of a permanent file from the remote host and gives it the name lfn, specified on the MFLINK command.

Format:

GET,pfn/UN=username,PW=password,PN=packname,R=r,NA,RT,WB.

PACKNAM Directive

The PACKNAM directive operates the same as its NOS command counterpart. If you use the PACKNAM directive, it must precede any file transfer directive in the same directives record. The PACKNAM directive only applies to the directive record in which it appears.

Format:

PACKNAM,PN=packname,R=r.

or

PACKNAM,packname,R=r.

If you are authorized to access auxiliary devices, you can always access files on an auxiliary device that is already mounted. If you attempt an MFLINK access of a file on an unmounted auxiliary device, you get a DEVICE UNAVAILABLE message unless your site has enabled the wait-for-unmounted-device installation option. If this option is present on more than one mainframe in an RHF network, a deadlock situation is possible. If such a deadlock occurs, MFLINK would time-out the connection as discussed previously.

PERMIT Directive

The PERMIT directive operates the same as its NOS command counterpart.

Format:

PERMIT,pfn,username=m/PN=packname,R=r,NA,WB,XD=expiredate,XT=expireterm.

PURGE Directive

The PURGE directive operates the same as its NOS command counterpart.

Format:

PURGE,pfn/UN=username,PW=password,PN=packname,R=r,NA,WB.

REPLACE Directive

The REPLACE directive replaces pfn on the remote host with file lfn, as specified on the MFLINK command.

Format:

REPLACE,pfn/UN=username,PW=password,PN=packname,R=r,NA,WB.

SAVE Directive

The SAVE directive creates an indirect access file named pfn on the remote host and copies onto it the file specified by lfn on the MFLINK command.

Format:

SAVE,pfn/PW=password,CT=ct,M=m,BR=br,PR=pr,PN=packname,R=r,NA,WB,AL=level,
XD=expiredate,XT=expireterm,AC=ac.

USER Directive

The USER directive specifies validation information for the remote host. It operates the same as its NOS command counterpart. The USER directive must be the first directive associated with an MFLINK command that has ST=lid specified. You do not need to specify this directive at any other time.

Format:

USER,username,password,familyname.

If the remote host is a NOS Version 2.2 or higher system, the password specified must be the batch password.

INTERACTIVE USE OF MFLINK

When you use MFLINK in an interactive job, the system obtains the directives in one of the following ways:

- If you specify I=dirfile on the MFLINK command, the system reads the local file named dirfile. If dirfile is assigned to the terminal, the system prompts the terminal using the prefix character.
- If you omit the I=dirfile parameter, the system also prompts the terminal using the prefix character. If the MFLINK command is part of a procedure, the system reads the lines immediately following the command. If the first line following the command does not begin with the prefix character, the system prompts the terminal.

When reading from the terminal, the system prompts for directives until you enter an empty line (only a carriage return) or the prefix character in response to a prompt. The system then sends the directives (minus the prefix character) to the remote host. The remote host responds with messages indicating the success or failure of the operations.

Any time you enter an MFLINK command that has the ST=lid specified, the system initiates a new MFLINK session. If you enter the command without the ST=lid parameter, the system resumes the session with the remote host currently connected.

Examples: (NOS to NOS)

1. Assume you are at mainframe MFB. To transfer a file from mainframe MFA to MFB and purge it on MFA, you must enter commands and directives similar to the following:

```
MFLINK,NEWFILE,ST=MFA.  
*USER,XYX,ABC.  
*ATTACH,OLDFILE/PN=USER.  
*PURGE,OLDFILE/PN=USER.
```

2. Assume you are on mainframe MFB. To transfer four files from mainframe MFA, enter commands and directives similar to the following:

```
MFLINK,FILE1,ST=MFA,I=DIRECTS.  
MFLINK,FILE2,I=DIRECTS.  
MFLINK,FILE3,I=DIRECTS.  
MFLINK,FILE4,I=DIRECTS.
```

The file DIRECTS has the following contents:

```
USER,XYZ,ABC.  
CHARGE,1234,345N355.  
GET,LFN1.  
-EOR-  
GET,LFN2.  
-EOR-  
ATTACH,PFN1/PN=USER.  
-EOR-  
ATTACH,PFNS.  
-EOI-
```


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, RT.

<u>Parameter</u>	<u>Description</u>
lfn=pfm	Specifies the one- to seven-character file name lfn 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 checkpoint or no-auto-drop status.
WB	Specifies the wait-if-busy option. WB causes the job to wait for the mounting of disk packs.
RT	Specifies the real-time processing option. If RT is specified, the job continues processing after requesting staging of a file from alternate storage 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 OLD command to assign the file to the job.

If RT is omitted and the file resides only on alternate storage, job processing is suspended while the file is staged to disk and assigned to the job. For more information on alternate storage, refer to Alternate Storage in section 2.

If an auxiliary device has been previously specified by a PACKNAM command, the system attempts to find the permanent file, pfn, 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 checkpoint or no-auto-drop status. Certain system files (for example, ZZZZC0, ZZZZC1, ZZZZC2, and ZZZZLD) always have no-auto-drop 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 a removable 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 auxiliary device. However, to request permanent files on another auxiliary device, you must specify the PN=packname parameter on the file request or enter another PACKNAM command before the request (you must also specify the R=r parameter if the device type is different). 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 or public file.

Format:

PERMIT,pfn,username₁=m₁,username₂=m₂,...,username_n=m_n/PN=packname,R=r,NA,WB,
XD=expiredate,XT=expireterm.

<u>Parameter</u>	<u>Description</u>
pfn	Name of the 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 the DEFINE Command or SAVE Command description 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 a removable auxiliary device on a device type other than the installation-defined default is to be used.

<u>Parameter</u>	<u>Description</u>
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.
XD=expiredate	Specifies the expiration date for the file permit. The value expiredate must be a 6-digit string of the form yymmdd; where yy, mm, and dd are 2-digit year, month, and day designators, respectively.
XT=expireterm	Specifies the life of the file permit in days. If you specify 0, the file permit expires immediately. If you specify 4095 or *, the file permit does not expire.

PURGALL COMMAND

The PURGALL command purges all permanent files in your catalog that satisfy the criteria specified by the parameters.

Format:

PURGALL,AD=ad,AF,CD=cd,CT=ct,DF,DN=dn,EF=ef,FN=fn,MD=md,NA,PN=packname,R=r,TM=tm,TY=ty,WB.

<u>Parameter</u>	<u>Description</u>
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.
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.
CD=cd	Specifies the creation date; its format is yymmdd. All files created before this date are purged, unless the AF parameter is specified.
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.
DF	Specifies that a dayfile message will be issued for each file that is purged.
DN=dn	Specifies the device number assigned to a device during system initialization. Only files on that device are purged.

<u>Parameter</u>	<u>Description</u>
EF=ef	<p>Specifies one or more excluded permanent file names. This parameter excludes these permanent files from being purged.</p> <p>An asterisk (*) present in a file name indicates that a set of files is to be excluded. For example, EF=***OPL will exclude from purging all the files with 6-character names that end in OPL; EF=M***** will exclude from purging all the files with names that start with the letter M.</p> <p>Multiple file names may be specified, separated by slashes. For example, EF=A*****/B*****/C***** will exclude from purging all the files with names that begin with the letters A, B, or C.</p> <p>The EF parameter may be specified together with the FN parameter. For example, FN=AB*****,EF=ABC**** will purge all the files with names that start with the letters AB, except for those files with names that start with the letters ABC.</p>
FN=fn	<p>Specifies one or more permanent file names. This parameter restricts the purge to these permanent file names.</p> <p>An asterisk (*) present in a file name indicates that a set of files is to be purged. For example, FN=***OPL will purge all the files with 6-character names that end in OPL; FN=M***** will purge all the files with names that start with the letter M.</p> <p>Multiple file names may be specified, separated by slashes. For example, FN=A*****/B*****/C***** will purge all the files with names that begin with the letters A, B, or C.</p> <p>The FN parameter may be specified together with the EF parameter. For example, FN=AB*****,EF=ABC**** will purge all the files with names that start with the letters AB, except for those files with names that start with the letters ABC.</p>
MD=md	<p>Specifies the last modification date; its format is yymmdd. All files last modified before this date are purged, unless the AF parameter is specified.</p>

<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.
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).
TM=tm	Specifies the time of day on the date specified by the AD, MD, or CD parameter; its format is hhmmss.
TY=ty	Specifies the file type to be purged.

<u>ty</u>	<u>Action</u>
A or ALL	Purges all files.
D or DIRECT	Purges all direct access files.
I or INDIR	Purges all indirect access files.

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.

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, DF, DN, EF, FN, NA, PN, R, TM, TY, and one date (either AD, CD, or MD) 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.

<u>Parameter</u>	<u>Description</u>
pfn ₁	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 a removable 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.

RECLAIM COMMAND

The RECLAIM command performs a number of functions related to the transfer of NOS files between your job and a dump file. The dump file can be a magnetic tape or a mass storage (disk) file. Using RECLAIM you can selectively dump local or permanent files, retrieve files from a dump file, and list information about files previously dumped to tape or mass storage.

RECLAIM provides you with an easy way of making backup copies of local files or of direct and indirect access permanent files. It also gives you greater control over the size of your permanent file disk space. RECLAIM can be entered from a batch or interactive job.

Dumped files can be reloaded as local or permanent files. Reloaded permanent files have the same permissions, modes, permit categories, and so forth that they had when dumped. Note, however, that file history information for reloaded files (that is, file creation date, last date modified, last date accessed, etc.) is set to the date and time of loading. Files can be reloaded from tape by the file owner or by an alternate user.

Information about dumped files (such as file name, dump date, tape VSN, dump number, and so forth) can be stored in a dump database. This database is maintained as a direct access file in your permanent file catalog. RECLAIM automatically creates a dump database when you enter a DUMP directive unless you specify that you do not want a database for the dump

(using the DB=0 parameter). If you do not specify a name for the database, RECLAIM uses the default name RECLDB. To get information on individual files in the database, enter a RECLAIM command with the LIST directive.

Multiple users can share a dump database. For example, alternate users can access a database in your catalog by entering a RECLAIM command specifying your database file name (on the DB parameter) and your user name (on the UN parameter). As with any other permanent file, alternate users must have WRITE permission to access and use your database file.

MAGNETIC TAPES

As previously mentioned, you can use RECLAIM to dump or retrieve files to or from magnetic tapes as well as mass storage media. You should be aware of the following special rules that apply when using tapes with RECLAIM:

- You do not need to enter a REQUEST command for RECLAIM dump tapes. RECLAIM issues its own REQUEST command internally when you enter a COMPACT, DUMP, LOAD, or COPY directive (RECLAIM directives are described below). RECLAIM uses the default parameter values for the REQUEST command.

NOTE

Since RECLAIM issues the REQUEST command internally, you must be prepared to wait for the tape mount if you enter a COMPACT, DUMP, LOAD, or COPY directive from an interactive job. Other alternatives are to submit the RECLAIM command from a batch job or to detach your interactive job using the Detach (ctD) command.

- RECLAIM dump tapes must be labeled; unlabeled tapes are not permissible.
- If you enter a LOAD, COPY, or LIST directive for a tape that is not already defined to the RECLAIM database, RECLAIM reads the entire tape and adds all files on the tape, or all files matching the specified user index if the UI processing option is specified, to the database. This feature lets you add the entire contents of a tape to the database without having to name each file individually.
- The tape processing options listed under Processing Options in section 12 are valid for RECLAIM tapes. Processing options are specified with the PO option.
- The permanent file tapes are written by the RECLAIM utility in a format compatible with both PFDUMP and PFLOAD. The NOS 2 Analysis Handbook describes these permanent file utilities.

COMMAND FORMAT

The parameters of the RECLAIM command are order-independent. The directives and the options are appended to the end of the command.

NOTE

For sites using the Tape Management System (TMS), there are additional directive options available on the RECLAIM command which are not described in this manual. For a description of the RECLAIM command under TMS, refer to the NOS Version 2 Tape Management System (TMS) User Reference manual.

Format:

RECLAIM, p_1, p_2, \dots, p_n / directive₁, options₁ / directive₂, options₂ / directive_n, options_n.

Parameter

Description

P_i

Specifies any of the following optional parameters.

Parameter

Description

DB=pfm

Specifies the name of the direct access file that contains the RECLAIM database. The default file name for pfm is RECLDB. If a value of 0 is entered for pfm, no database is created or maintained.

Do not attach the database file before you call RECLAIM; RECLAIM attaches the file for you.

I=lfm₁

Specifies the name of the local file containing the RECLAIM input directives to be processed. The default file name for lfm₁ is INPUT.

L=lfm₂

Specifies the name of the local file to receive output listing. The default file name for lfm₂ is OUTPUT.

NA

No abort option. Inhibits RECLAIM abort in case of a program error.

NH

Inhibits RECLAIM from printing a header on file lfm₂.

NV

Specifies that a dump is to be written at end-of-information (EOI), regardless of what precedes the EOI. If this parameter is omitted, RECLAIM will not write a dump beginning at EOI unless at least one valid RECLAIM dump precedes the EOI.

<u>Parameter</u>	<u>Description</u>
<u>Parameter</u>	<u>Description</u>
PN=packname	Specifies the auxiliary device name. You specify packname only if the database file resides on an auxiliary device.
PW=password	Specifies the file password for the database file; applicable only if the database file resides in an alternate user catalog.
R=devtype	Specifies the mass storage device type for database residence. This parameter is applicable only if the PN parameter is also specified. If PN is specified but R is not, the system default device type specified by IPRDECK entry DFPT is used.
UN=username	Specifies the user name of the database file owner; applicable only if the database file resides in an alternate user catalog. You must have WRITE permission to access files under an alternate user name.
Z	Specifies that the RECLAIM command contains input directives following the terminator. If the Z parameter is specified, the I parameter is ignored. Use of the Z parameter eliminates the need for an input file (lfn ₁) when only a few directives are required.
directive _i	Specifies how RECLAIM is to manipulate files that meet the criteria set forth in the directive options. (RECLAIM directives and options are described below.)
options _i	Specifies a file characteristic or attribute RECLAIM is to use in selecting the file or files to be processed. (RECLAIM directives and options are described below.) Use a comma to separate individual options.

RECLAIM directives define the operation to be performed on a specified file or files. All directives except END can have one or more associated options. A comma is used as the separator between a directive and its associated options.

When you enter a RECLAIM command with the Z parameter, you type in the directives following the command terminator. The first character following the terminator must be the separator character used to separate individual directives. The separator character can be any character that does not appear within the directives. (We use the / character in the examples in this section.)

Following are descriptions of the RECLAIM directives. The description of the options for the directives immediately follows the description of the directives.

Directive

Description

COMPACT

Rewrites a dump file using only the active files from the database. All files logically deleted (using the DELETE directive) from your database are physically deleted from the dump file and the database. Options may also be specified which apply additional file selection criteria. Files which do not meet these additional criteria are also physically deleted. The database is updated to remove all inactive entries and update the dumpfile number for the remaining entries. COMPACT cannot be used on dump files that have no database file.

The active files rewritten by the COMPACT directive are consolidated into a single dump before being rewritten; that is, the new dump file contains only one dump, unless existing dumps on the new dump file are preserved by specifying the EI or EI=Y option. Files are written to the new dump file at beginning-of-information (file is rewound) unless EI or EI=Y is specified. If EI or EI=Y is specified and RECLAIM determines that a dump file is empty or does not contain a RECLAIM dump, it sends the following message to the dayfile and the terminal:

UNKNOWN DUMP FILE WILL BE REWRITTEN.

If the COMPACT directive was entered interactively, RECLAIM issues the prompt:

IS THIS OK (YES OR NO)?

If you enter YES, the COMPACT directive proceeds normally. For a NO response, RECLAIM ignores the directive and prompts for the next one. In a noninteractive job, RECLAIM proceeds with the COMPACT directive.

You must specify one of the following options to indicate where the dump file to be created by COMPACT is to be stored:

CF=clfn
CN=cpfn
CT=vsfn
OV

Either or both of the CF=clfn and CN=cpfn options must be specified in conjunction with the DT=MS option if a new mass storage file is to receive the dump.

The CT=vsfn option must be specified if a new tape is to receive the dump. If you intend to enter a COMPACT directive specifying a new tape, you must issue a RESOURC command to allow RECLAIM to request the two tapes at once.

Directive

Description

You must specify the OV option if the dump is to be rewritten on the original tape or file. To rewrite a dump tape, the compacted dump is first written to a disk scratch file, then copied to the original file.

If the OV option is specified, the EI or EI=Y option should not be used. If it is used, it will be ignored.

If you know or expect that a dump will require multiple reels, you must enter a VSN command before entering RECLAIM. The VSN command should specify as many VSNs as necessary and should associate the VSNs with the local file name specified by the CF option. If no CF option is specified, the default local file name NTAPE is used.

COPY

Creates local file copies of all dumped files that meet the criteria specified by the COPY directive options. (The COPY directive is the same as the LOAD directive except that the LOAD directive creates permanent files from dumped files.) The copied files are created using their original permanent file names unless they are explicitly renamed.

The COPY directive performs its own tape requests internally. If a copy is initiated for several files resident on a single tape, that tape is only requested once. If the copy specifies files on multiple tapes, multiple tape requests are initiated.

The COPY directive can be used to read files from a tape that is not listed in the RECLAIM database. When used in this way, RECLAIM reads all files from the tape (regardless of how many files are specified by the COPY directive options) and adds all files to the RECLAIM database.

The latest (most recently dumped) version of each file is copied unless you specify other characteristics. The following are examples of a COPY directive:

RECLAIM,Z./COPY,PF=ALPHA/COPY,PF=BETA/COPY,PF=SIGMA

or

RECLAIM,Z./COPY,PF=*/ALPHA,BETA,SIGMA

Either of the above commands copies the latest (most recently dumped) versions of files ALPHA, BETA, and SIGMA and makes them local files.

DELETE

Disables all files in your database that meet the criteria specified by the DELETE directive options. Deleted files are not physically deleted from the database and can be restored to active status using the RESET directive. Use the LIST directive with the DE option to list deleted files. You can use DELETE to temporarily disable selected files, thus allowing you to load all files except those disabled. All files disabled by DELETE are physically removed from the dump file if a COMPACT directive is entered before the disabled files are reactivated (using RESET). The following is an example of a DELETE directive:

RECLAIM,Z./DELETE,DD=851204

The above example instructs RECLAIM to disable all files dumped on the date 12/4/85.

Directive

Description

DUMP

Dumps to tape or mass storage all permanent files that meet the criteria specified by the DUMP directive options. Information that RECLAIM requires to perform future reloads from the dump file is stored in a database file within the specified user catalog. If the specified database file does not exist, RECLAIM creates it. You can inhibit the creation or use of a database file by specifying the DB=0 parameter.

Files are dumped in the order they appear in your permanent file catalog unless you specify the PF or FN option. Files are then dumped in the order they are listed more for PF or FN. If a file is listed more than once for PF or FN, multiple copies are dumped, in the specified order.

Files dumped are written to the dump file at EOI unless EI=N is specified. If RECLAIM determines that a dump file is empty or does not contain a RECLAIM dump, it sends the following message to the dayfile and the terminal:

UNKNOWN DUMP FILE WILL BE REWRITTEN.

If the DUMP directive was entered interactively, RECLAIM issues the prompt:

IS THIS OK (YES OR NO)?

If you enter YES, the DUMP directive proceeds normally. For a NO response, RECLAIM ignores the directive and prompts for the next one. In a noninteractive job, RECLAIM proceeds with the DUMP directive.

The following is an example of a RECLAIM DUMP directive:

```
RECLAIM,Z./DUMP,TY=D,TN=001442
```

This example instructs RECLAIM to dump all direct access permanent files in your catalog to a magnetic tape with the VSN of 001442. If a database named RECLDB does not currently exist, RECLAIM creates it.

If you know or expect that a dump will require multiple reels, you must enter a VSN command before entering the RECLAIM command. The VSN command should specify as many VSNs as necessary and should associate the VSNs with the local file name specified by the DF option. The default name is TAPE. For example:

```
VSN,TAPE=001442/001501/001995.  
RECLAIM,Z./DUMP,TN=001442,TY=D
```

The above example duplicates the previous example, but allows for a multi-reel dump.

<u>Directive</u>	<u>Description</u>
END	<p>Ends the current RECLAIM session. At an interactive terminal, a blank line followed by a carriage return is equivalent to an END. In a directive file, an EOR has the same effect as an END directive.</p> <p>You can use the F tape processing option (refer to Processing Options in section 12) to instruct RECLAIM to unload a tape when the END directive is entered. To do this, enter the following SET directive before you initiate any operation that would cause a tape to be mounted (such as a COPY, DUMP, or LOAD):</p> <p style="margin-left: 40px;">SET,PO=F</p>
LIST	<p>Lists RECLAIM database information about all dumped files that meet the criteria specified by the accompanying directive options. Files are listed in alphabetical order except for dump files without a database. Dump files with no database are listed in sequential order.</p> <p>The LIST directive can be used to list files on a tape that is not listed in the RECLAIM database. When used in this way, RECLAIM adds all files on the tape to the RECLAIM database, regardless of how many files are specified by the LIST directive options.</p>
LOAD	<p>Loads into your permanent file catalog all of the files that meet the criteria specified by the LOAD directive options. (The LOAD directive is the same as the COPY directive except that the COPY directive creates only local files.) The latest (most recently dumped) version of each file is loaded unless you specify other characteristics. If a file name specified in a LOAD directive already exists in the catalog, the file will not be loaded unless the RP or RP=Y option is also specified.</p> <p>The LOAD directive can be used to load files from a tape that is not listed in the RECLAIM database. When used in this way, RECLAIM loads all files from the tape (regardless of how many files are specified by the LOAD directive options) and adds all files to the RECLAIM database.</p> <p>The LOAD directive performs its own tape requests internally. If a load is initiated for several files resident on a single tape, that tape is only requested once. If the load specifies files on multiple tapes, multiple tape requests are initiated. Following are examples of the LOAD directive:</p> <p style="margin-left: 40px;">RECLAIM,Z./LOAD,PF=MYFILE</p> <p>The above LOAD directive loads into your catalog the most recently dumped version of MYFILE.</p> <p style="margin-left: 40px;">RECLAIM,Z./LOAD,DD=840112,NF=595</p> <p>The above LOAD directive loads up to 595 files dumped on the date 1/12/84.</p>

Directive

Description

RECLAIM,Z./LOAD,PF=ALPHA,NN=BETA,MD=831114

or

RECLAIM,Z./LOAD,PF=*,MD=831114/BETA=ALPHA

Either of the previous LOAD directives loads the latest version of file ALPHA with a modification date of 11/14/83. File ALPHA is loaded into your catalog as file BETA. If a permanent file called BETA already exists in your catalog, the file is not loaded. If a permanent file named ALPHA exists in the catalog, it does not affect, and is not affected by, this operation.

QUIT

Equivalent to END.

REMOVE

Permanently removes a tape volume serial number (VSN) from the database. This purges from the database all entries for the specified tape.

RESET

Reactivates all files previously disabled by the DELETE directive that meet the criteria specified by the RESET directive options.

SET

Redefines the RECLAIM defaults for any directive options. Typically, this directive is used during a long RECLAIM session to establish criteria for subsequent processing. If a SET directive is encountered without an accompanying option, it has no effect on current defaults.

The options that can be associated with RECLAIM directives are described below.

<u>Option</u>	<u>Description</u>												
AA=yymmdd	Instructs RECLAIM to process only those files accessed after a specified date. The AA option may be used in conjunction with AB to specify a date range. The option is used with DUMP. If DB=0, it may also be used with COPY, LIST, and LOAD.												
AB=yymmdd	Instructs RECLAIM to process only those files accessed before a specified date. The AB option may be used in conjunction with AA to specify a date range. The option is used with DUMP. If DB=0, it may also be used with COPY, LIST, and LOAD.												
AD=yymmdd	Instructs RECLAIM to process only those files accessed on a specified date. The AD option is used with DUMP. If DB=0, it may also be used with COPY, LIST, and LOAD.												
CF=clfn	Specifies the name of the local file to receive the compacted dump. If clfn is omitted, the local file name is the name specified for the CN parameter. If the CN parameter is omitted, the default name NTAPE is used for clfn. This option is used with COMPACT.												
CN=cpfn	Specifies the permanent file to receive the compacted dump if you do not want to overwrite the original dump file. (Use the OV option to overwrite the original dump file.) The CN option allows you to specify different names for the local and permanent file copies of the dump file. This option does not create, save, or replace the permanent file. These operations are the user's responsibility. This option is used with COMPACT.												
CT=vsn	Specifies the VSN of the tape to which the compacted dump is to be written. The CT option is used with COMPACT.												
D=den	Specifies tape density. Values that can be specified for den are as follows:												
	<table border="1"> <thead> <tr> <th><u>den</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>HI</td> <td>556 cpi; 7-track tapes</td> </tr> <tr> <td>HY</td> <td>800 cpi; 7-track tapes</td> </tr> <tr> <td>HD</td> <td>800 cpi; 9-track tapes</td> </tr> <tr> <td>PE</td> <td>1600 cpi; 9-track tapes</td> </tr> <tr> <td>GE</td> <td>6250 cpi; 9-track tapes</td> </tr> </tbody> </table>	<u>den</u>	<u>Description</u>	HI	556 cpi; 7-track tapes	HY	800 cpi; 7-track tapes	HD	800 cpi; 9-track tapes	PE	1600 cpi; 9-track tapes	GE	6250 cpi; 9-track tapes
<u>den</u>	<u>Description</u>												
HI	556 cpi; 7-track tapes												
HY	800 cpi; 7-track tapes												
HD	800 cpi; 9-track tapes												
PE	1600 cpi; 9-track tapes												
GE	6250 cpi; 9-track tapes												
	The D option is used with COMPACT and DUMP.												
DA=yymmdd	Instructs RECLAIM to process only those files dumped after a specified date. DA may be used in conjunction with DB to specify a date range. The DA option is used with COMPACT, COPY, DELETE, LIST, LOAD and RESET.												
DB=yymmdd	Instructs RECLAIM to process only those files dumped before a specified date. DB may be used in conjunction with DA to specify a date range. The DB option is used with COMPACT, COPY, DELETE, LIST, LOAD and RESET.												
DD=yymmdd	Instructs RECLAIM to process only those files dumped on a specified date. The DD option is used with COMPACT, COPY, DELETE, LIST, LOAD and RESET.												

<u>Option</u>	<u>Description</u>
DE	Instructs RECLAIM to process only those files that have been disabled by DELETE. If this option is omitted, only undeleted files are processed. The DE option is used with COMPACT, COPY, LIST, LOAD and RESET.
DF=dlfn	Specifies the name of the local file copy of a dump file. The default name for dlfn is TAPE. The DF option is used with DUMP.
DN=dpfn	Specifies the file to or from which files are to be processed. Only files on this dump file that meet other specified selection criteria are selected. The DN option is logically equivalent to and interchangeable with the TN option except if a DT, MT, or NT option is not specified. In this case, DN implies a mass storage dump file while TN implies a tape dump file (therefore, TN causes RECLAIM to issue an internal REQUEST command, whereas DN does not). The DN option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD, and RESET. If the DN option (or DT=MS) is used for a DUMP operation, RECLAIM does not make the mass storage dump file permanent. If you wish to make the dump file permanent, you may do so. However, the dump file must be local before you can use it with RECLAIM.
DT=dtype	Specifies the type of dump file to be created. Possible values for dtype are:

<u>dtype</u>	<u>Description</u>
MS	The dump is placed on a mass storage (disk) file.
MT	The dump is placed on 7-track magnetic tape. (Same as MT option.)
NT	The dump is placed on 9-track magnetic tape. (Same as NT option.)

The DT option is used with COMPACT and DUMP.

EI=option	Specifies whether RECLAIM is to write dumped files at the beginning of the dump file or write dumped files at the end of the dump file.
-----------	---

<u>option</u>	<u>Description</u>
Y	Files are to be written to the dump file at end-of-information.
N	Files are to be written to the dump file at beginning-of-information, thus destroying any existing information on the dump file.

If EI appears alone, it is equivalent to EI=Y. If EI is omitted, DUMP processing assumes EI=Y; COMPACT processing assumes EI=N. The EI option is used with COMPACT and DUMP.

<u>Option</u>	<u>Description</u>
EX=option	Specifies the use of exception processing for RECLAIM options. Values that can be specified for option are Y or N. Y indicates that exception processing is used; that is, only files that fail one or more selection criteria are processed. N indicates that normal processing is used; that is, only files that meet all selection criteria are processed. The EX option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD, and RESET.
F=format	Specifies any valid NOS tape format. The F option is used with COMPACT or DUMP.
FI=nn	Specifies the dump number (1 to 63) to which the dump file is to be positioned. A dump is the set of user files dumped to a file by a single DUMP or COMPACT directive. The FI option can be used in conjunction with the RC option to position the tape or file pointer to a particular file within a specified dump. (The FI value associated with a particular file is listed in the FI column of the RECLAIM output listing.) The FI option is used with COPY and LOAD.

<u>Option</u>	<u>Description</u>
FN=filename	Processes files in the same manner as the PF option with one exception: When used with a DUMP directive, it first checks to see if a named file exists as a local file. If so, the local file is dumped. The TY option can be used with FN to specify whether named local files are to be dumped as direct access or indirect access files. The default is direct access. The FN option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD, and RESET. For all directives except DUMP, the FN and PF options are interchangeable.
FT	Foreign tape. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.
GT=length	Instructs RECLAIM to process only those files that have a length in PRUs greater than the length specified. GT may be used in conjunction with LT to specify a size range. The GT option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.
LT=length	Instructs RECLAIM to process only those files that have a length in PRUs less than the length specified. LT may be used in conjunction with GT to specify a size range. The LT option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.
LV	Specifies that the most current version of the specified file is to be processed. This is the default for the COPY and LOAD directives. The LV option is used with COMPACT, COPY, DELETE, LIST, LOAD and RESET.
MA=yyymmdd	Instructs RECLAIM to process only those files modified after a specified date. MA may be used in conjunction with MB to specify a date range. The MA option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.
MB=yyymmdd	Instructs RECLAIM to process only those files modified before a specified date. MB may be used in conjunction with MA to specify a date range. The MB option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.
MD=yyymmdd	Instructs RECLAIM to process only those files modified on a specified date. The MD option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.
MT	Specifies the use of a 7-track tape drive. The MT option is used with DUMP.
NF=num	Specifies the maximum number of files to be processed. If the NF option is omitted, there is no limit on the number of files processed, except for the DUMP directive. The limit for DUMP is 4095 files. The NF option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.
NN=pfname	Specifies a new file name for the file named by the PF option. The NN option is used with COPY, DUMP and LOAD.

<u>Option</u>	<u>Description</u>
NT	Specifies the use of a 9-track tape drive. The NT option is used with DUMP.
OV	Specifies that a compacted dump is to be written over the old dump tape. The OV option is used with COMPACT.
PF=pfn	<p>Specifies the name of the permanent file or files to be processed by RECLAIM. If only one file is to be processed, pfn is any valid NOS file name. If PF and FN are both omitted, the files to be processed are determined by the other options which have been specified. If multiple files are to be selected by name, specify an asterisk (*) for pfn.</p> <p>When PF=* is specified, a file name list containing up to 999 file names may be entered. A line of file names ending with a comma indicates that the file name list is continued on the next line. For interactive requests, RECLAIM prompts you to enter the file names with the message:</p> <p style="text-align: center;">ENTER FILE NAMES.</p> <p>The file name list contains one entry for each file to be processed. Each entry consists of the name of the file to be processed, unless the file is to be renamed. If the file is to be renamed, the entry is in the format, newname=oldname. Entries are separated by commas. An entry not followed by a separator indicates the end of the list. A comma at end of line indicates that another line follows.</p> <p>Following is an example of a file name list:</p> <p style="text-align: center;">RECLAIM,Z./DUMP,TN=MYVSN,PF=*/ALPHA,GAMMA=BETA</p> <p>This command will cause RECLAIM to dump files ALPHA and BETA. The files will be identified in the dump and in the database as ALPHA and GAMMA. The PF option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD and RESET.</p>
PO=plp2...pn	Specifies general tape processing options for a dump tape. Refer to Processing Options in section 12 for a description of dump tape processing options. Options I, P, and S are not accepted by RECLAIM. The PO option is used with COMPACT, COPY, LIST, LOAD, DUMP, and SET. The PO option is seldom required, as the RECLAIM defaults cover most situations adequately. PO=UR is the default for the COPY, LIST, and LOAD directives. PO=UW is the default for the COMPACT and DUMP directives.
PW=password	Password. Refer to the NOS Version 2 Tape Management System (TMS) User Reference manual for a detailed description.
RC=nnnn	Specifies the dump record number (1 to 4095) within a dump that indicates where the dump file is to be positioned. The RC option is used in conjunction with the FI option to specify the location of a file within a dump. (The RC value associated with a particular file is listed in the REC column of the RECLAIM listable output.) The RC option is used with COPY and LOAD.

<u>Option</u>	<u>Description</u>
RP=option	<p>Specifies that a file is to be loaded or copied from a dump file, even if a local or permanent file by the same name already exists. Possible values for option are Y, N, or C.</p> <p>When used with the LOAD directive, specifying RP or RP=Y indicates that an existing permanent file will be replaced by a file loaded from a dump. Specifying RP=N or omitting the RP option indicates that an existing permanent file will not be replaced by a loaded file.</p> <p>When used with the COPY directive, specifying RP or RP=Y indicates that a local file will be rewound and overwritten by a file copied from a dump. Specifying RP=C indicates that a file from a dump will be copied to an existing local file at its current location. If RP=N is specified, or if the RP option is omitted, RECLAIM returns the existing local file before copying the file from DUMP.</p> <p>The RP option is used with COPY and LOAD.</p>
TN=vsfn	<p>Specifies the VSN of the tape to or from which files are to be processed (VSN of the first reel for a multi-reel dump). Only files on this tape which also meet any other specified selection criteria are processed. This option is logically equivalent to and interchangeable with the DN option except if a DT, MT, or NT option is not specified. In this case, DN implies a mass storage dump file while TN implies a tape dump file (therefore, TN causes RECLAIM to issue an internal REQUEST command, whereas DN does not). The TN option is used with COMPACT, COPY, DELETE, DUMP, LIST, LOAD, RESET and SET.</p>
TO=username	<p>User name. Refer to the NOS Version 2 Tape Management System (TMS) User Reference manual for a detailed description.</p>
TY=type	<p>Specifies the permanent file type to be processed. Values that can be specified for the TY option are: D (direct access) and I (indirect access). If this option is omitted, both file types are eligible for processing. The TY option is used with COMPACT, COPY, DELETE, LIST, LOAD, and RESET.</p>
UI=user index	<p>Instructs RECLAIM to process only files for the specified user index when the COPY, LIST, or LOAD directive is used to process a dump tape or file not on the database, or when no database is being used (DB=0). If this option is either omitted or specified as UI=0, the user index of files on the dump tape or file being processed will be ignored.</p>
UN=username	<p>Instructs RECLAIM to process files only for the specified user name on the RECLAIM database, when files are shared among multiple users. If this option is omitted, processing is performed only on the calling user's dumped files. Files for all users will be processed if UN=0 is specified. The UN option is used with COMPACT, COPY, DELETE, LIST, LOAD and RESET. If the UN option specifies a user name not originally in the RECLAIM database, RECLAIM loads all files (LOAD directive) or lists information about all dumped files (LIST directive) on a file set specified by the TN or DN options. Loaded files are recorded in the database under the user name specified by the UN option.</p>

RECLAIM OUTPUT

All RECLAIM directives except END, QUIT and SET generate output listings in the following format:

```
RECLAIM Vvvv  OP=dddddd  UN=username yy/mm/dd. hh.mm.ss.  PAGE nn
PFN  TYPE LAST MOD   DUMP DATE  LENGTH USERNAME   TAPE  FI REC
```

```
pfname ty yy/mm/dd  yy/mm/dd      len  dmpuser  dumpid fi rc
pfname ty yy/mm/dd  yy/mm/dd      len  dmpuser  dumpid fi rc
```

n FILES PROCESSED

where:

vvv Current version of RECLAIM.

dddddd Name of the RECLAIM directive being processed.

username User name of the calling user.

nn Current page number (not displayed for interactive users).

pfname Name of the permanent file being processed.

ty File type (I for indirect access or D for direct access).

len Length in PRUs of file pfname when dumped.

dmpuser The user name under which file pfname was originally dumped.

dumpid VSN of the tape or name of dump file containing file pfname (VSN of the first reel for a multi-reel dump).

fi The dump number on dumpid containing file pfname.

rc The record number on dump fi containing file pfname.

Entering a RECLAIM command with the NH parameter specified suppresses printing of the heading and the n FILES PROCESSED line; only the file information is printed.

Example 1

The following RECLAIM command lists a portion of dump file database MYDB, and terminates the RECLAIM session.

```
/reclaim,db=mydb,z./list,tn=01/end
RECLAIM V4.2  OP=LIST      UN=OMEN    83/12/14. 13.03.34.
PFN  TYPE LAST MOD   DUMP DATE  LENGTH USERNAME   TAPE  FI REC

DEMA  I 83/11/18  83/12/13    187  OMEN    01      1  2
DEMB  I 83/11/18  83/12/13    27   OMEN    01      1  1
```

2 FILES PROCESSED.

RECLAIM COMPLETE.

/

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, M=m,  
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.
M=m	Specifies the file access mode if the file does not already exist, or if you want to change the access mode. If M=m is omitted for a newly created file, M=READ is assumed. If the parameter is omitted for a previously existing file, the current access mode is retained.
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 a removable 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. When you save a file using this command, the charge number and project number currently in effect for your job are associated with the file.

Format:

```
SAVE, lfn1=pfn1, lfn2=pfn2, ..., lfnn=pfnn / PW=password, CT=ct, M=m, SS=subsystem,
BR=br, PR=pr, PN=packname, R=r, NA, WB, AL=level, XD=expiredate, XT=expireterm, AC=ac.
```

<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 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 permit 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 border="0"> <tr> <td style="padding-right: 20px;">If m is:</td> <td>Other users can:</td> </tr> <tr> <td>R or READ</td> <td>Retrieve a copy of the file. This copy can</td> </tr> <tr> <td>RU or READUP</td> <td>be read or executed.</td> </tr> <tr> <td>RA or READAP</td> <td></td> </tr> <tr> <td>RM or READMD</td> <td></td> </tr> <tr> <td>U or UPDATE</td> <td></td> </tr> <tr> <td>E or EXECUTE</td> <td>Retrieve a copy of the file. This copy can only</td> </tr> <tr> <td></td> <td>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</td> </tr> <tr> <td></td> <td>the file. The user can enter GET, OLD, NEW, and</td> </tr> <tr> <td></td> <td>APPEND commands, but not a REPLACE command, for</td> </tr> <tr> <td></td> <td>the file.</td> </tr> <tr> <td>W or WRITE</td> <td>Retrieve a copy of the file, append data to it,</td> </tr> <tr> <td></td> <td>replace it, or purge it.</td> </tr> <tr> <td>N or NULL</td> <td>No access is allowed.</td> </tr> </table>	If m is:	Other users can:	R or READ	Retrieve a copy of the file. This copy can	RU or READUP	be read or executed.	RA or READAP		RM or READMD		U or UPDATE		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.
If m is:	Other users can:																																
R or READ	Retrieve a copy of the file. This copy can																																
RU or READUP	be read or executed.																																
RA or READAP																																	
RM or READMD																																	
U or UPDATE																																	
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; possible values are MD (one backup copy required), N (no backup), or Y (two backup copies required). If BR=br is omitted, BR=Y is assumed.
PR=pr	Specifies the preferred file residence; possible values are D (disk residence preferred), L (locked to disk), M (alternate storage residence preferred), N (no preference), or T (tape alternate storage residence preferred). If PR=pr is omitted, PR=N is assumed.
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.
AL=level	Specifies the security access level you want assigned to the file. Unless changed by your site, level can be one of the following names: LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7 The default is the access level of the local file being saved.
XD=expiredate	Specifies the expiration date for the file password. The value expiredate must be a 6-digit string of the form yymmdd; where yy, mm, and dd are 2-digit year, month, and day designators, respectively. You can specify this parameter only if you also specify the PW=password parameter.
XT=expireterm	Specifies the life of the file password in days. If you specify 0, the file password expires immediately. If you specify 4095 or *, the file password does not expire. You can specify this parameter only if you also specify the PW=password parameter.
AC=ac	Specifies whether alternate users may obtain information about the file using CATLIST. Entries are Y (yes) and N (no). The default is N.

The local files, lfn_i, are rewound before and after the save operation.

SETPFAC COMMAND

The SETPFAC command changes the set of security access categories associated with a permanent file. You can either specify an entirely new category set or add to or delete from the current category set. By default on a secured system, the system associates all of the access categories for which you are authorized to any permanent file you create.

Format:

```
SETPFAC,pfn,AC=0,category1,category2,...,categoryn/PN=packname,R=r,NA,WB.
```

or

```
SETPFAC,pfn,AC=category1,category2,...,categoryn/PN=packname,R=r,NA,WB.
```

<u>Parameter</u>	<u>Description</u>
pfn	Specifies the permanent file whose access category set you want to change.
AC=0,... category _i ...	Specifies a new category set. Unless changed by your site, category _i can be any of the following names: CAT00,CAT01,...,CAT31
AC=... category _i ...	Specifies categories to add or delete from the current category set. To add a category, prefix the category name with a plus sign or use no prefix at all. To delete a category, prefix the category name with a minus sign.
PN=packname	Specifies the auxiliary device name. You must specify this parameter if pfn resides on an auxiliary device.
R=r	Specifies the device type. You must specify this parameter if a removable 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 first format creates a new access category set. The second format adds to or deletes from the current category set.

On a secured system, the categories you specify must be valid for your job (ENQUIRE,B returns this information). On an unsecured system, the system does not use the categories to monitor access to the file but does maintain the categories as a file attribute.

SETPFAL COMMAND

The SETPFAL command changes the security access level of a permanent file. By default, a direct access file inherits the access level of the creating job. An indirect access permanent file inherits the access level of the local file from which it was created. You can also explicitly set the access level when you create a permanent file by specifying the AL=level parameter on the command you use to create the file.

Format:

SETPFAL,pfn,AL=level/PN=packname,R=r,NA,WB.

The first two parameters are required and are order-dependent.

<u>Parameter</u>	<u>Description</u>
pfn	Specifies the permanent file whose access level you want to change.
AL=level	Specifies the security access level you want assigned to the file. Unless changed by your site, level can be one of the following names: LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7
PN=packname	Specifies the auxiliary device name. You must specify this parameter if pfn resides on an auxiliary device.
R=r	Specifies the device type. You must specify this parameter if a removable 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.

On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information). In addition, to lower the access level of a file, you must be authorized to do so (refer to the LIMITS command).

On an unsecured system, the system does not use this access level assignment to monitor access to the file but does maintain this assignment as a file attribute.

LOAD/DUMP MEMORY COMMANDS

11

The load/dump memory commands transfer information that resides in your job field length to a local file or 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 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 volume 2, Guide to System Usage). 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. Refer to Express Deadstart Dump Interpreter in the NOS 2 Analysis Handbook for a description of how file ZZZZDMB can be formatted into a report.

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. If you omit this parameter or specify a null entry, the record name becomes D000000.
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	47	35	17	0	
0	7700	0016	0	0		} prefix table
1	D dump record name			0		
2	yy/mm/dd.					
3	hh.mm.ss.					
4	pgnr					
5	0					
6	0					
7	0					
10	pgnr					
11	ccdr					
.						
.						
.						
16						

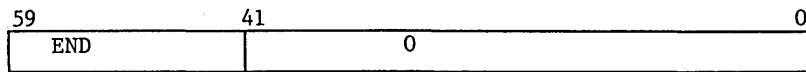
59	53	17	0
D	dump record name	0	

	59	53	47	17	0
0	ID	0	5		
1	version				
2	version				
3	△	yy/mm/dd.			
4	△	hh.mm.ss.			
5	machine characteristics				

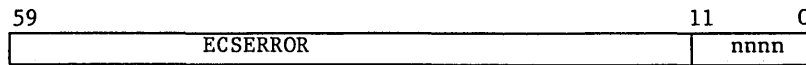
	59	47	17	0
0	XP	0	32g	
1	exchange package			
:	(A0) register			
:	:			
20	(A7) register			
:	RA			
30	RA+1			
31				
32				

	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				



If an extended memory error occurs, the block in error will be filled with 400g words of the following format:



The variable items in the preceding display have the following descriptions:

<u>item</u>	<u>Description</u>
n	Octal number of central memory words assigned to the job.
m	Octal number of extended memory words assigned to the job.
△	Space (display code value 55g).
nnnn	Word number of the block in error (starting with 1).
pgnr	Name of the system program called by DMB. Words 4 and 10 are blank-filled.
ccdr	An image of RA+70 to RA+75 of the field length dumped. This is usually a copy of the command being executed prior to the dump.

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.

NOTE

The DMDECS command can dump only up to two million words of extended memory. If you want to dump more than two million words, use the DMB command (described in this section).

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 10_8 . If fwa is greater than or equal to 400000_8 , the first dump address is fwa minus 400000_8 , 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 40_8 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 40_8 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 10_8 words of the job's field length are dumped. If fwa equals lwa, the 10_8 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 rewound 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.

NOTE

The DMPECS command can dump only up to two million words of extended memory. If you want to dump more than two million words, use the DMB command (described in this section).

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 10g 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 7700g, 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

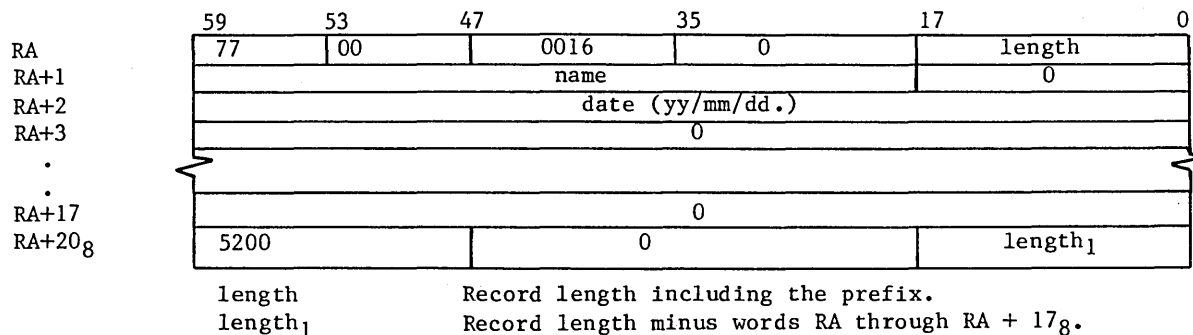
The RBR command loads one binary record from a specified file.

Format:

RBR,n,name.

<u>Parameter</u>	<u>Description</u>
n	Specifies that 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, 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.
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, 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.
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.

To write on unlabeled tapes, you must have special authorization (refer to the LIMITS command).

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, or 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>Pi</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 7-track I or SI format tapes. Refer to the P and S options.
L	Does not issue tape detailed status error messages to the user dayfile. The default is installation-defined (refer to the M option).
M	Issues only the first and last tape detailed status error messages to the user dayfile 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 L option).
N	Specifies that the 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 an unlabeled tape, it directs the system to stop at the first tape mark after the EOT is sensed. For a labeled tape, it directs the system to stop at the tape mark plus EOF1 or the tape mark plus EOF1 when the EOT is encountered.

Pi

Description

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 and the RECLAIM command, 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.

Specifying QN=9999 on the LABEL command has the same effect for multifile tapes.

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 and 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

To assign a tape unit using the ASSIGN command, your job must be of system origin or you must be authorized to use magnetic tapes and to assign devices by EST ordinals.

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=vs_{n1}/vs_{n2}=...=vs_{n-1}/vs_n, {MT}, {D=den}, F=format, LB=lb,
 {FC=fcount}, {CV=cv}, NS=ns, PO=p₁p₂...p_n, {CK}, AL=level.
 {C=ccount}, {N=cv}, {CB}

Required parameters:

<u>Parameter</u>	<u>Description</u>
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 _{n1} /vs _{n2} =...=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 cpi	HD	800 cpi
HI	556 cpi	PE	1600 cpi
HY	800 cpi	GE	6250 cpi
200	200 cpi	800	800 cpi
556	556 cpi	1600	1600 cpi
800	800 cpi	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.

<u>Format</u>	<u>Description</u>
I	Internal.
SI	System internal.†
L	Long block stranger.
S	Stranger.
F	Foreign (designed for tapes of unknown data format).

LB=lb	Labeled or unlabeled tape. Default is KU if VSN is omitted or KL if VSN is specified.
-------	---

<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.
---------------	---

<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. Any block containing fewer than ns frames is considered as noise and is discarded. The NS parameter is not valid for I or SI format tapes or for other tape formats when either PE (9-track, 1600 cpi) or GE (9-track, 6250 cpi) density is specified. For PE and GE densities, the noise block size is zero. Default is 18 frames for other formats. Maximum value is 31 frames. If NS=0 is specified, the default is used.
PO=P1P2...Pn	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). <p style="margin-left: 40px;">CK Each dump is written at the previous EOI of lfn.</p> <p style="margin-left: 40px;">CB Each dump is written at the BOI of lfn.</p>
AL=level	Specifies the security access level to assign to the file. Unless changed by your site, level can be one of the following names: <p style="margin-left: 40px;">LVL0, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7</p> <p>The default is the current access level of the job.</p> <p>On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information) and for the device to which you are assigning the file.</p> <p>On an unsecured system, the parameter has no effect. The system does, however, check for valid parameter specifications.</p> <p>When using this command to assign a file for checkpoint dumps on a secured system, use the AL=level parameter to assign the file at the highest access level valid for the job (ENQUIRE,B returns this information). The CKP command aborts the job if the checkpoint file is not at this access level.</p>

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=vs_n, { MT } , { D=den } , { CV=cv } , FA=fa, OFA=ofa, VA=va,
 { NT } , { den } , { N=cv }

OWNER=username/familyname, LSL=ls_l, U.

<u>Parameter</u>	<u>Description</u>
VSN=vs _n	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.

NOTE

Avoid specifying a volume serial number that matches a local file name attached to your job or the local file will be lost.

Omitting the VSN parameter discards any local file named BLKTAPE from your job.

<u>Parameter</u>	<u>Description</u>																																
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 border="0" style="margin-left: 40px;"> <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 cpi</td> <td>HD</td> <td>800 cpi</td> </tr> <tr> <td>HI</td> <td>556 cpi</td> <td>PE</td> <td>1600 cpi</td> </tr> <tr> <td>HY</td> <td>800 cpi</td> <td>GE</td> <td>6250 cpi</td> </tr> <tr> <td>200</td> <td>200 cpi</td> <td>800</td> <td>800 cpi</td> </tr> <tr> <td>556</td> <td>556 cpi</td> <td>1600</td> <td>1600 cpi</td> </tr> <tr> <td>800</td> <td>800 cpi</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 cpi	HD	800 cpi	HI	556 cpi	PE	1600 cpi	HY	800 cpi	GE	6250 cpi	200	200 cpi	800	800 cpi	556	556 cpi	1600	1600 cpi	800	800 cpi	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 cpi	HD	800 cpi																														
HI	556 cpi	PE	1600 cpi																														
HY	800 cpi	GE	6250 cpi																														
200	200 cpi	800	800 cpi																														
556	556 cpi	1600	1600 cpi																														
800	800 cpi	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 border="0" style="margin-left: 40px;"> <thead> <tr> <th><u>cv</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>AS</td> <td>ASCII/6-bit display code conversion.</td> </tr> <tr> <td>US</td> <td>Same as AS.</td> </tr> <tr> <td>EB</td> <td>EBCDIC/6-bit display code conversion.</td> </tr> </tbody> </table>	<u>cv</u>	<u>Meaning</u>	AS	ASCII/6-bit display code conversion.	US	Same as AS.	EB	EBCDIC/6-bit display code conversion.																								
<u>cv</u>	<u>Meaning</u>																																
AS	ASCII/6-bit display code conversion.																																
US	Same as AS.																																
EB	EBCDIC/6-bit display code conversion.																																
FA=fa	File accessibility character indicating who has access to the labeled tape. Value entered in HDR1 and EOF1 labels.																																
	<table border="0" style="margin-left: 40px;"> <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 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>	<u>fa</u>	<u>Description</u>	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.																								
<u>fa</u>	<u>Description</u>																																
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.																																
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.																																

[†]Refer to Magnetic Tape Users in appendix A.

<u>Parameter</u>	<u>Description</u>
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.

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). Although the SI=setid parameter must be specified when appending a file to a multifile set, the appended file will be given the same setid as the preceding file regardless of the setid value specified in the LABEL command. This is done to ensure that all files in a file set have the same set identifier.

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, $\left\{ \begin{array}{l} \text{MT} \\ \text{NT} \end{array} \right\}$, D=den, F=format, LB=lb,
 $\left\{ \begin{array}{l} \text{FC=fcount} \\ \text{C=ccount} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{CV=cv} \\ \text{N=cv} \end{array} \right\}$, NS=ns, PO=p₁p₂...p_n, $\left\{ \begin{array}{l} \text{CK} \\ \text{CB} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{SI=setid} \\ \text{M=setid} \end{array} \right\}$,
 $\left\{ \begin{array}{l} \text{SN=secno} \\ \text{V=secno} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{QN=seqno} \\ \text{P=seqno} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{FI=fileid} \\ \text{L=fileid} \end{array} \right\}$, FA=fa, G=genno, E=gvn,
 $\left\{ \begin{array}{l} \text{CR=cdate} \\ \text{C=cdate} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{RT=yyddd} \\ \text{T=ddd} \end{array} \right\}$, $\left\{ \begin{array}{l} \text{W} \\ \text{R} \end{array} \right\}$, AL=level, AC=ac, CT=ct, MD=md, PW=password, TO=to, UN=username.

Required parameter:

<u>Parameter</u>	<u>Description</u>
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.

<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	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 cpi	HD	800 cpi
HI	556 cpi	PE	1600 cpi
HY	800 cpi	GE	6250 cpi
200	200 cpi	800	800 cpi
556	556 cpi	1600	1600 cpi
800	800 cpi	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 (designed for tapes of unknown data format).

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). Not applicable 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"> <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 is discarded. The NS parameter is not valid for I or SI format tapes or for other tape formats when either PE (9-track, 1600 cpi) or GE (9-track, 6250 cpi) density is specified. For PE and GE densities, the noise block size is zero. Default is 18 frames for other formats. Maximum value is 31 frames. If NS=0 is specified, the default is used.								
PO=p1p2...pn	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"> <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>The system leaves the set identifier field in the HDR1 label blank if you omit the SI=setid parameter or if you specify blanks for setid.</p> <p>You must specify a nonblank file set identifier to build a multifile set. Hence, if you omit this parameter when you write the first file on the tape, the tape cannot then be a multifile set tape. When you append a file to a multifile set tape, you must specify this parameter to position the tape. Also, the specified set identifier must be the same as that of the first file in the multifile set.</p>								
SN=secno or V=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 or P=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 or C=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>
AL=level	<p>Specifies the security access level to assign to the file. Unless changed by your site, level can be one of the following names:</p> <p style="text-align: center;">LVLO, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7</p> <p>The default is the current access level of the job.</p> <p>On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information) and for the device to which you are assigning the file.</p> <p>On an unsecured system, the parameter has no effect. The system does, however, check for valid parameter specifications.</p> <p>When using this command to assign a file for checkpoint dumps on a secured system, use the AL=level parameter to assign the file at the highest access level valid for the job (ENQUIRE,B returns this information). The CKP command aborts the job if the checkpoint file is not at this access level.</p>

<u>Parameter</u>	<u>Description</u>
AC=ac	Alternate auditability. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.
CT=ct	File category. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.
MD=md	File Mode. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.
PW=password	Password. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.
TO=to	TMS tape option. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.
UN=username	User name. Refer to the NOS 2 Tape Management System (TMS) User Reference Manual for a detailed description.

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.
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.
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=QSORT,B=STAPE.
LABEL,STAPE,VSN=TP03,D=PE,SI=BINSET,FI=LSORT,QN=9999.
FTN5,I=LSORT,B=STAPE.
LABEL,STAPE,VSN=TP03,D=PE,SI=BINSET,QN=2.
COPYEI,STAPE,QSORTB.
QSORTB.
```


The GET command retrieves three indirect access files - BSORT, QSORT, and LSORT - containing FORTRAN 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.  
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. For output listing files not connected to your terminal, the LISTLB command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

Format:

```
LISTLB, lfn, {SI=setid } , {QN=seqno } , 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 HDR1 label contains this value are listed. You must specify this parameter to list the labels of a multifile set.
QN=seqno or P=seqno	One- to four-digit file sequence number. If seqno is specified, only the label group whose HDR1 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. 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 HDRI/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.† You can direct the operator to make the tape assignment by including a comment following the command terminator. This comment appears at the system console. 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. TMS does not support the REQUEST command.

Format:

```
REQUEST, lfn, VSN=vsn1/vsn2=...=vsn-1/vsn, { MT } , { D=den } , F=format,
      LB=lb, { FC=fcount } , { CV=cv } , NS=ns, PO=p1p2...pn, { CK } , AL=level.comment
      { C=ccount } , { N=cv }
```

Required parameter:

<u>Parameter</u>	<u>Description</u>
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=vs _{n1} /vs _{n2} =...=vs _{n-1} /vs _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 REQUEST 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>	
den	Density	den	Density
LO	200 cpi	HD	800 cpi
HI	556 cpi	PE	1600 cpi
HY	800 cpi	GE	6250 cpi
200	200 cpi	800	800 cpi
556	556 cpi	1600	1600 cpi
800	800 cpi	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>
F	Foreign (designed for tapes of unknown data format).
I	Internal.
L	Long block stranger.
SI	System internal.†
S	Stranger.

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"> <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.								
NS=ns	Noise size. Any block containing fewer than ns frames is considered as noise and is discarded. The NS parameter is not valid for I or SI format tapes or for other tape formats when either PE (9-track, 1600 cpi) or GE (9-track, 6250 cpi) density is specified. For PE and GE densities, the noise block size is zero. 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.								
AL=level	Specifies the security access level to assign to the file. Unless changed by your site, level can be one of the following names: <p style="margin-left: 40px;">LVLO, LVL1, LVL2, LVL3, LVL4, LVL5, LVL6, or LVL7</p> The default is the current access level of the job.								

[†]Refer to Magnetic Tape Users in appendix A.

Parameter

Description

On a secured system, the access level you specify must be valid for the job (ENQUIRE,B returns this information) and for the device to which you are assigning the file.

On an unsecured system, the parameter has no effect. The system does, however, check for valid parameter specifications.

When using this command to assign a file for checkpoint dumps on a secured system, use the AL=level parameter to assign the file at the highest access level valid for the job (ENQUIRE,B returns this information). The CKP command aborts the job if the checkpoint file is not at this access level.

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 NT062.
```

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 combination 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.

<u>Parameter</u>	<u>Description</u>
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 position and/or 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 ZZZZCO, ZZZZC1, and ZZZZC2 are also included if present. Data files using the default names ZZCLAA, ZZCLAB, ..., and ZZCLZZ are also included if present.
- 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 mass storage 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 on a mass storage file 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.

When procedure scratch files are copied, they are copied from BOI to EOI.

For checkpointed magnetic tape files, only the relative position of the tape is captured so that the tape can be properly positioned during restart. Mixed mode 7-track tapes are not supported when using CHECKPOINT or RESTART.

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,x₁.

<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.

Parameter

Description

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.

The procedure scratch files ZZZZC0, ZZZZC1, and ZZZZC2 are not restored from the checkpoint file. The values of the symbolic names EFG, EF, R1G, R1, R2, R3, and DSC of this restart job are used, not the values from the checkpoint file CPA.

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.

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 mass storage files included in the specified checkpoint. This option does not restore ZZZZC0, ZZZZC1, or ZZZZC2. 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.

Before you enter the RESTART command, do the following:

- Assign the checkpoint file to the job. You must include a REQUEST, ASSIGN, or LABEL command if the checkpoint file resides on magnetic tape. You can use a GET or ATTACH command if the checkpoint file resides on mass storage.
- Assign to the job all necessary magnetic tape files to be used by the restarted job. You can use the LABEL command.

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.†

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,nnnn,x;.

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 attempt to retrieve the file unless one of the following is true:

- You specified the FC parameter and the file is already local to the restart job.
- The file resides on magnetic tape.

†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.

NOS provides the following utilities for file maintenance and screen formatting.

- EDIT Edits a text file.
- FSE Edits a file in line or screen mode.
- 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.
- PDU Compiles a panel definition file and loads it in a user library.
- 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.

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: 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.

FSE COMMAND

The FSE command calls the full screen editor FSE. For a complete description of the features of FSE and its parameters and directives, refer to the NOS Version 2 Full Screen Editor User's Guide.

Format:

FSE, FN=filename, OP=access, I=input, L=output, IP=procedure, WF=workfile.directives

or

FSE, filename, access, input, output, procedure, workfile.directives

The first format is the order-independent format for parameters. The second is the order-dependent format.

<u>Parameter</u>	<u>Description</u>
FN=filename	Specifies the file you want to edit. The file can be a local file or a permanent file. The default is the file most recently edited during your job.

OP=access Specifies the character set code and the location of the file to be edited. The following entries are valid:

<u>Access</u>	<u>Abbreviation</u>	<u>Description</u>
DISPLAY or NORMAL	D N	Selects 6-bit display code. This is the default if your terminal is NORMAL mode.
ASCII	A	Selects 6/12-bit display code. This is the default if your terminal is in ASCII mode.
ASCII8	8	Selects 7-bit ASCII code right-justified in a 12-bit byte.
GET	G	Specifies that the file is a direct access or an indirect access permanent file.

To specify both a character set and the location of the file, you must specify two values. For example, OP=8G and OP=AG are valid.

To specify the OP=access parameter, you must also specify the FN=filename parameter.

I=input Specifies an input file for directives. The default is file INPUT.

L=output Specifies an output file. The default is OUTPUT.

<u>Parameter</u>	<u>Description</u>
IP=procedure	Specifies a file to be the FSE procedure library. The default procedure library is FSEPROC. If file FSEPROC is not a local file, FSE automatically retrieves it. If FSEPROC is not one of your indirect access permanent files, FSE retrieves a copy from user name LIBRARY.
WF=workfile	Specifies a work file for FSE. The default is ZZZWORK.
directives	Specifies FSE directives to be executed before FSE executes those in the input file. Use semicolons to separate directives.

If you specify both an input directive file (using the I=input parameter) and an output listing file (using the L=output parameter), FSE is said to be in batch mode. When operating in batch mode, the input directive file must be coded in 6/12-bit display code.

NOTE

When FSE is executing in batch mode, or when the FSE command is included in a batch job, you can significantly increase processing efficiency by disabling UNDO processing (using the SET UNDO directive). This action inhibits FSE from keeping a record of each operation performed.

KRONREF COMMAND

The KRONREF command generates a cross-reference listing of system symbols used by decks on a Modify OPL. For output listing files not connected to your terminal, the KRONREF command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

NOTE

Before entering a KRONREF command specifying both OPL and NOSTEXT, you must enter an RFL command to reserve sufficient memory to contain systems text and OPL contents.

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.
- Mass storage table words.
- Global or local file name table words and ordinals.
- 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}.
- Executing job table, equipment status table, or queue file table.
- Negative field length symbols and dayfile message options.
- File types and mass storage constants.
- Job origin types and queue types.
- Error flags referenced.
- Common deck calls.
- PP packages called.
- Special entry points.

MODIFY COMMAND

The MODIFY command edits a Modify-formatted program library file. For output listing files not connected to your terminal, the MODIFY command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

Format:

MODIFY,P1,P2,...,Pn.

The following optional parameters can be specified in any order.

<u>Pi</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 _{n1} is COMPILE. 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. This parameter is meaningful only if you specify the Q=processor or X=processor parameter.
CL=lf _{n4}	Write full listing on lf _{n4} . CL=lf _{n4} is equivalent to the L=lf _n 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 CV=63 or CV=64 is specified, Modify prevents compile file generation (C=0).

<u>Pi</u>	<u>Description</u>
D	Do not abort the job on a directive error. (This parameter is referred to as debug mode.) 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. (This parameter is referred to as full edit mode.) By default, the U parameter or EDIT directives determine the decks to be edited.
I=lfng	Take directives from next record on lfng. INPUT is the default. The system reads no directives if you specify I=0.
L=lfng	Write list output on lfng. 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=lfng Write new program library on lfng. 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, the system performs an evict operation on the file specified by the N=lfng 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	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. 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=lfng ₂ , CL=lfng ₄ , CS=lfng ₅ , and CG=lfng ₃ 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).
S=lfng ₁₀	Write source output on file lfng ₁₀ . 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. Do not specify this parameter if you specify the A, Q=processor, or X=processor parameter.
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 80 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. For output listing files not connected to your terminal, the OPLEDIT command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

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.
D	Deck status.
S	Statistics.

If this parameter is omitted or just LO 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 LO is specified, list option E is selected.

<u>Pi</u>	<u>Description</u>
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.

PDU COMMAND

The PDU command compiles a panel definition file and stores the compiled panel in a user library. Refer to the NOS Version 2 Screen Formatting Reference manual for information on panels.

Format:

PDU,I=panel,L=listing,C=capsule,LIB=library,REC=recname,RW=option.

or

PDU,panel,listing,capsule,library,recname,option.

The first format is the order-independent format for the parameters. The second is the order-dependent format. Only the first parameter is required.

<u>Parameter</u>	<u>Description</u>
I=panel	Specifies the name of the panel definition file. The file must be local. You must specify this parameter.
L=listing	Specifies the name of the listing file. The default is OUTPUT. If you specify L=0, no listings are generated.
C=capsule	Specifies the name of the capsule file, the file on which the system writes the compiled panel. The default is CAPSULE. If you specify C=0, no capsule is generated.
LIB=library	Specifies the name of the local file to receive the capsule. The default user library is PANELIB. If you specify LIB=0, the system does not store the compiled panel in a user library.
REC=recname	Specifies the name of the panel definition record to be selected for processing from the panel definition file. If an asterisk (*) is entered, all records are processed. If REC is omitted, the current record is processed with no regard to the record name.
RW=option	Specifies if the panel definition file and listing file are to be rewound before and after processing.

<u>option</u>	<u>Description</u>
NO or N	Files are not rewound.
YES or Y	Files are rewound.

If RW is omitted, the files are rewound.

The PDU command is an interactive procedure. Hence, you can get help information for the command and its parameters and be prompted for parameters. If you enter

PDU?

the system gives you information about the command itself and prompts you for parameters. If you enter

PDU,parameter?

the system gives you information about the specified parameter and prompts you for parameters.

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 Version 2 Administration Handbook. For output listing files not connected to your terminal, the PROFILE command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

Format:

PROFILE, p_1, p_2, \dots, 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 PROFILC is assumed.
CN=cn	Charge number inquiry. All project numbers valid for charge number cn are written to output file. Valid only if the OP=I option is 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.
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=op 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 the CN or PN=pn parameter 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.

$dir_1, dir_2, \dots, dir_n$

Each directive is separated by a delimiter which can be any special character (display code greater than 44g) except an asterisk (*).

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>
APN=pn	Adds or activates project number pn.
AUN=un	Adds user names un (must be preceded by PN directive).
/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 (/).
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).
EFN=fn	Specifies the indirect access permanent file that contains the epilogue to be associated with the specified project. The epilogue can be a procedure or a binary program. A null entry, which is the default, clears the project epilogue.

<u>dir_i</u>	<u>Description</u>								
EPW=pw	Specifies the file password associated with the project epilogue file. A null entry, which is the default, indicates that no password is associated with the file.								
EUN=un	Specifies the user who owns the project epilogue file. A null entry, which is the default, indicates that the owner of the project epilogue file is the user on whose behalf the epilogue is to run.								
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.								
PCR=option	Defines the conditions under which the system sets a charge-required state after the project prologue executes. You can specify one of the following values: <table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>Option</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>U</td> <td>Set a charge-required state unconditionally.</td> </tr> <tr> <td>A</td> <td>Sets a charge-required state only if the project prologue aborts.</td> </tr> <tr> <td>null</td> <td>Does not set a charge-required state. This is the default.</td> </tr> </tbody> </table>	<u>Option</u>	<u>Description</u>	U	Set a charge-required state unconditionally.	A	Sets a charge-required state only if the project prologue aborts.	null	Does not set a charge-required state. This is the default.
<u>Option</u>	<u>Description</u>								
U	Set a charge-required state unconditionally.								
A	Sets a charge-required state only if the project prologue aborts.								
null	Does not set a charge-required state. This is the default.								
PEX=yyymmdd.	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.								
PFN=fn	Specifies the indirect access permanent file that contains the prologue to be associated with the specified project. The prologue can be a procedure or a binary program. A null entry, which is the default, clears the project prologue.								
PN=pn	Project number for which the following directives (until the next PN directive) apply.								
PPW=pw	Specifies the file password associated with the project prologue file. A null entry, which is the default, indicates that no password is associated with the file.								
PUN=un	Specifies the user who owns the project prologue file. A null entry, which is the default, indicates that the owner of the project prologue file is the user on whose behalf the prologue is to run.								
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.								

<u>dir_i</u>	<u>Description</u>
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,...,Pn.

The parameters are optional and can be specified in any order.

For file-specifying parameters (those associated with the keywords C, G, I, K, N, O, 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). The digit 6 or 8 overrides this default. The digit 6 directs the system to represent the file in the 6-bit display code set. The digit 8 directs the system to represent the file in the 7-bit 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 7-bit ASCII code set. For example, C8=FILE directs the system to write the decks on the compile file FILE in the ASCII code set.

<u>P_i</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, COMPILER becomes the compile file. If you specify the K=lf _{n1} parameter, the C=lf _{n1} parameter is ignored.

†The digit 8 directs the system to represent the file in the 6/12-bit display code set if the job is interactive and the file is an input or output file.

<u>P1</u>	<u>Description</u>								
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=lf _{n8} parameter).								
H=n	Specifies whether a 63- or 64-character character set is to be used in the Update run.								
	<table border="0" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: left;"><u>n</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>3</td> <td>A 63-character character set.</td> </tr> <tr> <td>4</td> <td>A 64-character character set.</td> </tr> <tr> <td>omitted</td> <td>Character set that was used in the old program library.</td> </tr> </tbody> </table>	<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.
<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 6/12-bit ASCII code if INPUT is associated with the terminal. Otherwise, it is display code.								
K=lf _{n1}	Specifies the compile file and specifies that the decks are written to that file in their order of appearance in the COMPILE directives. The default is file COMPILE. The default character code set is 6-bit display code.								

<u>Pi</u>	<u>Description</u>
L=c1c2...cn	Specifies one or more of the following list options.
	<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 7-bit ASCII code and the old program library is in 6-bit display code. In this case, the default is the 7-bit ASCII code set.

<u>P_i</u>	<u>Description</u>										
O=lfng	Specifies the output file. The default is OUTPUT. The default character code is the same as that of the old program library.										
P=lfng/s ₁ /s ₂ /.../s ₇	Specifies file lfng 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"> <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=lfng	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=lfng parameter. The default character code set of the source file is the same as that of the old program library.										
T=lfng	Functions the same as the S=lfng parameter, except that it excludes common decks from the source file. This parameter takes precedence over the S=lfng 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 6-bit 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.

NOTE

When XEDIT encounters lines longer than 150 characters it truncates those lines without warning; therefore, you should use caution when calling XEDIT to edit a file that may contain lines longer than 150 characters or when editing an ASCII file in NORMAL mode.

Format:

XEDIT, lfn₁, p₁, p₂, ..., p_n. directives

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>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 .

Parameter

Description

Pi

Description

I=lf_{n2}

Take editing directives from lf_{n2}. 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 lf_{n1}.)

L=lf_{n3}

Put all XEDIT output on the specified file. OUTPUT is the default. If you specify L=0, the system suppresses all output.

LK

XEDIT processes the edit file in look-only mode. The edit file can be examined, but not modified.

NH

Suppress printing of the XEDIT header.

P

Retrieve and edit permanent file lf_{n1}. Direct access files are attached in write mode. If P is omitted, the file lf_{n1} is assumed to be a local file.

directives

Process the directives that follow the command terminator. The first character that follows the terminator delimits the directives. The value directives has the following form (the semicolon serves as the delimiter):

;directive₁;directive₂;...;directiven

Do not place a terminator after the last directive.

A library is a file containing records that are accessed individually. This section describes the creation and maintenance of user libraries. User libraries are not to be confused with program libraries. User libraries are created using the LIBGEN, ULIB, PDU, or TDU commands. A user library can be named in an LDSET or a LIBRARY command to establish a local or global library set. Program libraries are source code libraries, created and edited using either the Modify utility or the Update utility.

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.
LIBGEN	Generates a user library; that is, a library of relocatable, capsule, absolute, CP overlay, or procedure file records that can be accessed by the CYBER Loader.
LIBRARY	Specifies a global library set for the job.
ULIB	Creates or modifies a local copy of a user library.
VFYLIB	Compares the records in two files and lists their differences.

We recommend that, except for files consisting solely of S-type and/or U-type records, only the CATALOG and ITEMIZE commands be used on files structured by the CYBER Record Manager (CRM). Because of certain incompatibilities with NOS file structures, the output from commands other than CATALOG and ITEMIZE may not reflect the CRM file structure accurately.

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.)

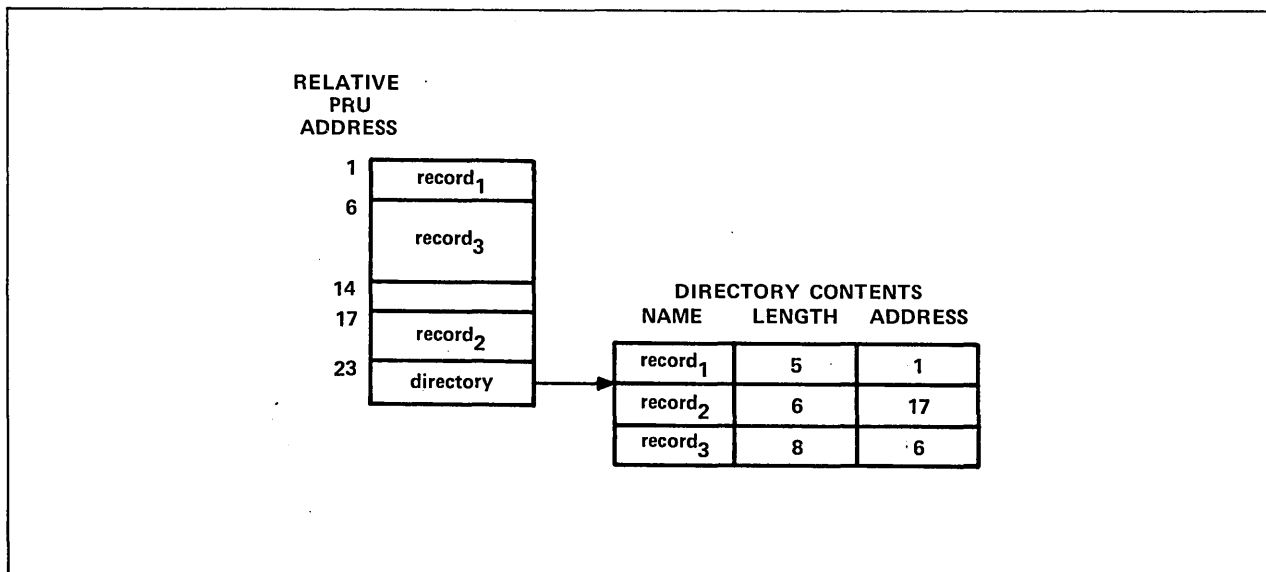


Figure 15-1. Random Access File Structure

To access a record randomly, the file must be on mass storage and must have a directory containing the address of each record (figure 15-1). 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.

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 type and name (if any) of the record are determined from the first and second words 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 the 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 example), they are considered binary records.

The following record types can be specified on the COPYX, 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>Record Type</u>	<u>Meaning</u>
ABS	Central processor overlay capsule containing multiple entry points.
CAP	CYBER Loader capsule.
OPL	Old program library deck created by Modify.
OPLC	Old program library common deck created by Modify or LIBEDIT.
OPLD	Old program library directory created by Modify or LIBEDIT.
OVL	Central processor overlay.
PP	Peripheral processor program.
PPU	Peripheral processor unit (first-level peripheral processor) program.
PROC	Procedure file.
REL	Relocatable central processor program.
TEXT	Data file (none of the other record types).
ULIB	User library directory.

The following record types are recognized by COPYL, COPYLM, and ITEMIZE commands.

<u>Record Type</u>	<u>Meaning</u>
ABS	Central processor overlay capsule containing multiple entry points.
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>Record Type</u>	<u>Meaning</u>
OPLD	Old program library directory created by Modify or LIBEDIT.
OVCAP	Central processor overlay capsule.
OVL	Central processor overlay.
PROC	Procedure file.
REL	Relocatable central processor program.
TEXT	Data file (none of the other record types)
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 library with master control character x. 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.
UPLR	Random program library deck created by Update.
UPLRC	Random program library common deck created by Update.
6PP	6000 series peripheral processor program.
7PP	7000 series peripheral processor program.
8PP	CYBER 180 peripheral processor program.

Appendix G in Volume 4, Program Interface, contains the formats of PP (6PP), OPL, OPLD, ULIB, TEXT, and PROC records. (The OPLC record format is identical to that of the OPL record.) The Modify Version 1 Reference Manual describes how to create ACF, OPL, OPLC, and OPLD records. The CYBER Loader Version 1 Reference Manual contains the formats of ABS, CAP, OVCAP, OVL, PP, and REL records. The CYBER Loader Instant contains the PPU (7PP) record format. The Update Version 1 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.

<u>Parameter</u>	<u>Description</u>																										
lfn	Specifies the file to be cataloged. The default is FILE.																										
Pi	Pi can be any of the following:																										
	<table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;"><u>Pi</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>N=n</td> <td>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.</td> </tr> <tr> <td>L=fname</td> <td>Specifies the name of the output file; if L=fname is omitted, CATALOG assumes L=OUTPUT.</td> </tr> <tr> <td>T</td> <td>Lists the entire text record if the record name begins with any of the following characters:</td> </tr> <tr> <td></td> <td style="text-align: center;"> <table border="0" style="width: 50%;"> <tr> <td>APRD</td> <td>IPRD</td> </tr> <tr> <td>CMRD</td> <td>LIBD</td> </tr> <tr> <td>EQPD</td> <td></td> </tr> </table> </td> </tr> <tr> <td></td> <td>If you omit T, the system does not list the contents of the text record. If the text record name begins with OVERLAY, the system always lists the first line of the record.</td> </tr> <tr> <td>U</td> <td>Lists all records within a user library. If file lfn is a user library and you omit U, CATALOG lists only the ULIB record.</td> </tr> <tr> <td>D</td> <td>Suppresses the comments field; suppresses all page headings after the initial page heading for each file.</td> </tr> <tr> <td>CS</td> <td>Suppresses character set indicator (63 or 64) for OPL and OPLC records.</td> </tr> <tr> <td>R</td> <td>Rewinds lfn before and after cataloging; if R is omitted, lfn is not rewound.</td> </tr> </tbody> </table>	<u>Pi</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:		<table border="0" style="width: 50%;"> <tr> <td>APRD</td> <td>IPRD</td> </tr> <tr> <td>CMRD</td> <td>LIBD</td> </tr> <tr> <td>EQPD</td> <td></td> </tr> </table>	APRD	IPRD	CMRD	LIBD	EQPD			If you omit T, the system does not list the contents of the text record. If the text record name begins with OVERLAY, the system always lists the first line of the record.	U	Lists all records within a user library. If file lfn is a user library and you omit U, CATALOG lists only the ULIB record.	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.
<u>Pi</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:																										
	<table border="0" style="width: 50%;"> <tr> <td>APRD</td> <td>IPRD</td> </tr> <tr> <td>CMRD</td> <td>LIBD</td> </tr> <tr> <td>EQPD</td> <td></td> </tr> </table>	APRD	IPRD	CMRD	LIBD	EQPD																					
APRD	IPRD																										
CMRD	LIBD																										
EQPD																											
	If you omit T, the system does not list the contents of the text record. If the text record name begins with OVERLAY, the system always lists the first line of the record.																										
U	Lists all records within a user library. If file lfn is a user library and you omit U, CATALOG lists only the ULIB record.																										
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. For CATALOG listing files not connected to a terminal, you can control the print density and page length using the SET command and the PL and PD symbolic names. 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.
TYPE	Record type (refer to Library Record Types in this section).
LENGTH	Record length in words, printed as an octal number. For TEXT records, this includes the prefix table length. For PROC records, this contains the actual record length. For other record types, the prefix table length is not included.
CKSUM	A checksum (that is, a value used to verify that the contents of a record were copied correctly). Except for TEXT records, the checksum does not verify the contents of the prefix table.
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.

NOTE

When printing output for text files, CATALOG uses the first seven characters of the first word of the record as the record name. CATALOG assumes that the record is in 6-bit display code. Therefore, if you attempt to display CATALOG output for a record in 6/12 display code while your terminal is in ASCII mode, the output display may be garbled. To avoid this problem, you should set your terminal to NORMAL mode before listing CATALOG output.

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 A	REL	41	7424	yy/mm/dd.
3	B B	REL	41	0465	yy/mm/dd.
4	C 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. The COPYL and COPYLM commands are commonly used to maintain files of procedures or relocatable records.

The COPYL and COPYLM commands differ only in the handling of multiple occurrences of a record on the old file. The COPYL command uses each record on the replacement file only once, replacing the first matching record from the old file. The COPYLM command uses the first matching record encountered on the replacement file to replace each matching record from the old file. The COPYL command 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 the COPYL or COPYLM command 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 except when you use the COPYL command to replace multiple occurrences of the same record (with multiple occurrences of that record on the replacement file). The system copies the records to the new file in the same order as they are on the old file.

The COPYL and COPYLM commands 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.

The COPYL and COPYLM commands replace only the record types 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>
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.
E	Process oldlfn until the end-of-information.
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.)
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.

These options can be specified by combining one or more letters in any order, such as TRA, AR, ERTA, or TR.

The COPYL and COPYLM commands 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 REPPFILE and writes them on NEWFILE. The A option appends to the end of NEWFILE any records in REPPFILE that do not match a record on OLDFILE.

```
COPYL,OLDFILE,REPPFILE,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,REPPFILE,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, REPPFILE, and NEWFILE.

```
/itemize,oldfile
```

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
```

REC	ITEMIZE OF REPPFILE NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
1	B	PROC	4		
2	C	PROC	4		
	* EOI *	SUM =	10		

```
ITEMIZE COMPLETE.
```

```
/itemize,newfile
```

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 extracts records selected from one file and writes them on 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.

Positional format:

GTR, lfn₁, lfn₂, d, NR, S, NA, T, directive₁, directive₂, ..., directive_n

Keyword format:

GTR, lfn₁, lfn₂/p₁, p₂, ..., p_n, directive₁, directive₂, ..., directive_n

NOTE

In positional format, the parameters must be in the order shown. Parameters are identified by their position, not by keywords. An omitted parameter is denoted by consecutive commas.

In keyword format, parameters preceding the / are still processed in positional format, but those which follow it (p₁, p₂, ..., p_n) are recognized by keyword, and may occur in any order. The keywords are the same as the suggested values for the positional format parameters: U, D, NR, S, NA, and T. Keywords U and D are covered under positional parameter "d" in the explanation of parameters which follows.

In either format, do not place blanks between the terminator (right parenthesis or period) and directive₁. Also, do not place a terminator after directive_n.

<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. For disk files, GTR always positions lfn ₂ at EOF before copying the selected records. GTR only appends records to tape files when you specify the NR parameter. Otherwise, GTR starts writing on tape files at their BOI.

Parameter

Description

d Random access directory parameter. If d is specified, lfn₂ 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 second and subsequent 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.
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 remainder of the library, including the 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 remainder of the library, including the 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. lfn₁ is not rewound after the operation; lfn₂ is not rewound before or after the operation. If lfn₁ has a directory, the directory is copied to lfn₂.

If NR is omitted, both files are rewound after the operation, but only when lfn₂ is a tape file is lfn₂ rewound before the operation.

<u>Parameter</u>	<u>Description</u>
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.
T	TEXT record name removal option. If specified, the line containing the record name (the first line of the record) is removed for each selected TEXT type record when the record is written to lfn ₂ .
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 is 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>Description</u>
<u>Directive</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, U.

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 a user library; default is list only the user library directory (ULIB type record).

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 (except TEXT records for which the prefix table is included).
CKSUM	Logical checksum (octal), excluding the prefix table (except TEXT records for which the prefix table is included).
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.

NOTE

When printing output for text files, ITEMIZE uses the first seven characters of the first word of the record as the record name. ITEMIZE assumes that the record is in 6-bit display code. Therefore, if you attempt to display ITEMIZE output for a record in 6/12-bit display code while your terminal is in ASCII mode, the output display may be garbled. To avoid this problem, you should set your terminal to NORMAL mode before listing ITEMIZE output.

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 OVERLAY APRDxxx, CMRDxxx, EQPDxxx, IPRDxxx, LIBDxxx 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 - REQUESTED 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
```

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 15-2). 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 debug mode was not specified (refer to the description of the D parameter under Command Format), LIBEDIT lists its interpretation of the conflicting directives, issues an error message, and aborts the job step. If debug mode was specified, LIBEDIT continues processing the directives.

For output listing files not connected to your terminal, the LIBEDIT command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

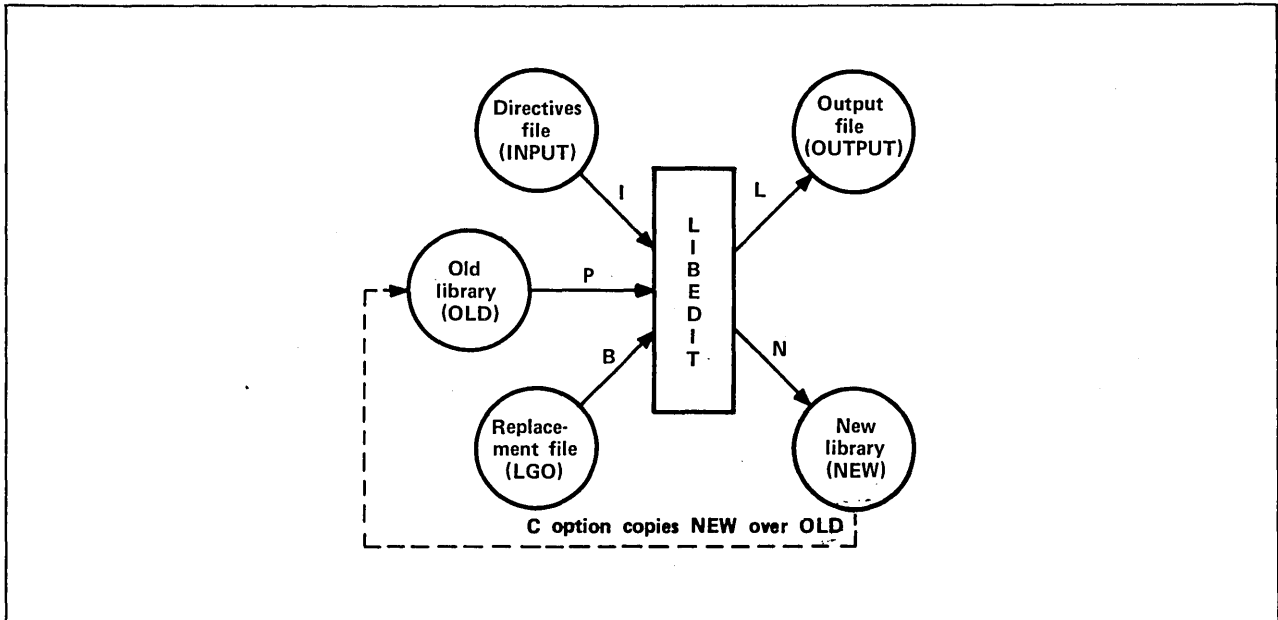


Figure 15-2. LIBEDIT Input and Output

COMMAND FORMAT

The following command calls LIBEDIT. Its parameters specify options and files used for the call as illustrated in figure 15-2.

Format:

LIBEDIT,p₁,p₂,...,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>
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.
C	Copy the new file over the old file after LIBEDIT processing.
D	Do not abort LIBEDIT processing because of errors. (This parameter is referred to as debug mode.)
I=lf _{n3}	Take the new directives from the next record of file lf _{n3} . The default is INPUT (INPUT is connected to the terminal in interactive jobs). I=0 indicates that there are no directives.
L=lf _{n5}	List output on file lf _{n5} . The default is OUTPUT. L=0 specifies no output.

<u>Pi</u>	<u>Description</u>												
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>F</td> <td>Give a full listing (enable all list options).</td> </tr> <tr> <td>M</td> <td>List modifications.</td> </tr> <tr> <td>N</td> <td>List records written to new file.</td> </tr> </tbody> </table>	<u>listopt</u>	<u>Meaning</u>	C	List directives.	E	List errors.	F	Give a full listing (enable all list options).	M	List modifications.	N	List records written to new file.
<u>listopt</u>	<u>Meaning</u>												
C	List directives.												
E	List errors.												
F	Give a full listing (enable all list options).												
M	List modifications.												
N	List records written to new file.												
	<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>												
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.												
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.												
N=lf _{n2}	Write the new file on lf _{n2} . The default is NEW.												
P=lf _{n1}	Edit old file lf _{n1} . The default is OLD. P=0 directs the system to create the new file from replacement files.												

NOTE

Although LIBEDIT can write the new file (lf_{n2}) directly to tape, the directory that LIBEDIT writes on such a tape contains tape PRU random addresses rather than disk PRU random addresses. If this tape is subsequently copied to disk, the directory will not be processed correctly by programs such as GTR and Modify. Therefore, if you intend to copy the file back to disk at a later time, you should specify a temporary disk file for the LIBEDIT new file (lf_{n2}) and then use the COPY utility to copy the new file to tape.

U=record	Require the old file to be a user library, add any new records 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. The value record becomes the name of the user library directory record. If you specify only the keyword U or omit the parameter entirely, the default record name is ULIB.
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.

<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=lfm3.

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=lfm3 parameter of the command, or can follow the terminator of the command when the Z parameter is specified. Directives are not required.

Unless specified otherwise by parameters or directives, 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 unless you specified the *NOREP 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.

Records on a replacement file which do not appear on the old file with the same name and type are considered unreplaceable. Such records are inserted at EOF of the new file, unless the NI parameter or *NOINS directive is specified. If a record with the same name or type appears on more than one replacement file or appears more than one on a replacement file, the extra copies are treated as unreplaceable records.

LIBEDIT processing of user libraries beginning with a ULIB record differs depending on whether or not the LIBGEN directive is specified. If no LIBGEN directive is specified, the user library is processed as one record (from ULIB header to OPLD). In this case, the only part of the user library that can be manipulated by LIBEDIT is the ULIB header. The records within the user library are transparent to LIBEDIT and cannot be referenced by directives. If the LIBGEN directive is specified, LIBEDIT deletes the ULIB header and OPLD records from all user libraries encountered. In this case, LIBEDIT directives can reference any records within the user library but cannot reference the ULIB header.

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.
*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).
*/	Specifies a directive file comment.

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 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/
P3
```

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 .

<u>Parameter</u>	<u>Description</u>
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,...,gid_n

<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	COMMENT	
	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,...,gidn
```

or

```
*B rid,gid1,gid2,...,gidn
```

<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
```

<u>Parameter</u>	<u>Description</u>
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
```

<u>Parameter</u>	<u>Description</u>
rid	Names the old file record to be written on the new file.
gid	Identifies records or groups of records on the replacement file that are to be copied.

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 overrides the VFYLIB directive.

Format:

*LIBGEN recordname

<u>Parameter</u>	<u>Description</u>
recordname	Specifies the name of the new user library directory record. The default is ULIB.

LIST Directive

The LIST directive changes the list file and the list option (refer to the L=lf_{n5} 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_{n2} parameter of the LIBEDIT command performs the same function.

Format:

*NEW newfile

<u>Parameter</u>	<u>Description</u>
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.

Format:

*NOINS

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
```

<u>Parameter</u>	<u>Description</u>
lfn ₁	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.

Format:

```
*NOREW
```

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
```

<u>Parameter</u>	<u>Description</u>
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 or TEXT type 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
```

<u>Parameter</u>	<u>Description</u>
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.

<u>Sequence 1</u>	<u>Sequence 2</u>
*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
```

<u>Parameter</u>	<u>Description</u>
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
```

<u>Parameter</u>	<u>Description</u>
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 U parameter overrides the VFYLIB directive.

Format:

*VFYLIB

*/Directive (Comment Directive)

The */ directive allows comments to be inserted into the LIBEDIT directive file.

Format:

*/ text

<u>Parameter</u>	<u>Description</u>
text	Specifies the comments (up to 80 characters in length) that are copied to the list file.

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 or the job is interactive). 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.  hh.mm.ss.          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.  hh.mm.ss.          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. hh.mm.ss.  PAGE 1
    *FILER F2
    *B *,REC1
RECORDS WRITTEN ON FILE NEW                          yy/mm/dd. hh.mm.ss.  PAGE 2
  RECORD  TYPE      FILE      DATE      COMMENT
INSERTED REC 1    TEXT      RF2
    **EOF**
    OLD

ERROR DIRECTORY - RECORDS NOT REPLACED.              yy/mm/dd. hh.mm.ss.  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 the CYBER Loader.

Format:

LIBGEN,p1,p2,...,pn.

Any of the following parameters may be specified in any order (only one instance of each).

<u>P_i</u>	<u>Description</u>
F=lf _{n1}	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. If the file F does not contain any of the record types specified, no library is generated. The default for this parameter is LGO.
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=lf _{n2} 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 latter case is the default.
P=lf _{n2}	Specifies the file on which the user library is to be written. The default is ULIB.

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 15-3 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.

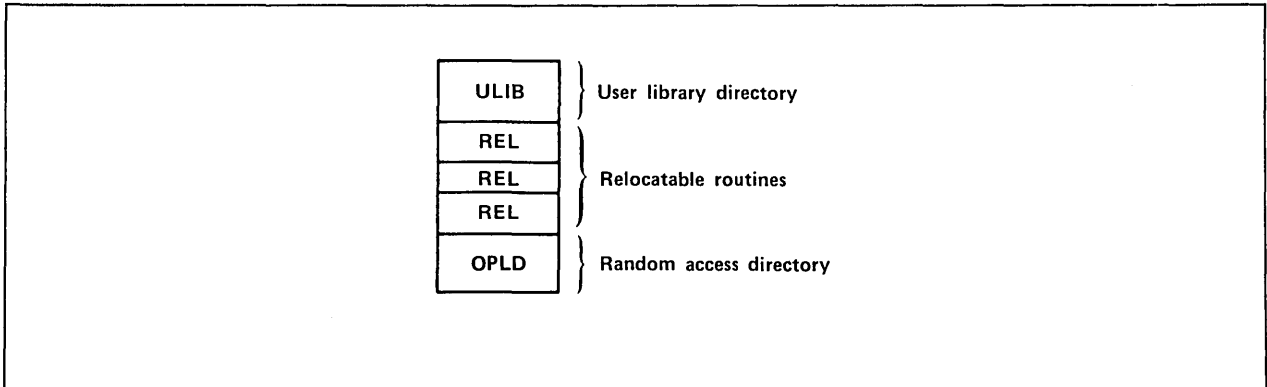


Figure 15-3. User Library Structure

Unless the NX=n parameter specifies otherwise, the ULIB directory contains the external reference/entry point linkage between routines in the user library. When the 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=$BLKDAT.$.
```

ULIB COMMAND

The ULIB command creates a user library or adds a record to, deletes a record from, or replaces a record in an existing user library. ULIB changes affect only the local copy of a user library. You can use the SAVE or REPLACE command, or copy the new library file to a direct access permanent file, to make the user library file permanent.

Format:

```
ULIB,OP=operation,REC=record,LIB=library.
```

or

```
ULIB,operation,record,library.
```

All parameters are required. The first format is the order-independent format for the parameters. The second is the order-dependent format.

<u>Parameter</u>	<u>Description</u>												
OP=operation	Specifies the library operation to be done. You can specify one of the following values:												
	<table><thead><tr><th><u>Operation</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>A</td><td>Add a record to the user library (same as R).</td></tr><tr><td>C</td><td>Create a new user library.</td></tr><tr><td>D</td><td>Delete a record from the user library.</td></tr><tr><td>F</td><td>Copy a record from the user library and make the copy a local file with the same name as the record.</td></tr><tr><td>R</td><td>Replace a record in the user library (same as A).</td></tr></tbody></table>	<u>Operation</u>	<u>Description</u>	A	Add a record to the user library (same as R).	C	Create a new user library.	D	Delete a record from the user library.	F	Copy a record from the user library and make the copy a local file with the same name as the record.	R	Replace a record in the user library (same as A).
<u>Operation</u>	<u>Description</u>												
A	Add a record to the user library (same as R).												
C	Create a new user library.												
D	Delete a record from the user library.												
F	Copy a record from the user library and make the copy a local file with the same name as the record.												
R	Replace a record in the user library (same as A).												
REC=record	Specifies the name of the record to be added, deleted, replaced, or copied. The record name must be the name of a local file.												
LIB=library	Specifies the name of the local library file to be created or modified. Except in the case of OP=C or OP=F, ULIB returns the original library file and creates a new local file. Hence, ULIB cannot be used to directly modify a library on a direct access file.												

The ULIB command is an interactive procedure. You can get help information for the command and its parameters and be prompted for parameters. If you enter

ULIB?

the system gives you information about the command itself and prompts you for parameters. If you enter

ULIB,parameter?

the system gives you information about the specified parameter and prompts you for parameters.

VFYLIB COMMAND

The VFYLIB command rewinds two files, compares their records, and lists the differences. VFYLIB does not perform a bit-by-bit comparison of the two files. It only compares such entries as name, type, length, and checksum for the records of the two files. For a binary comparison of two files, use the VERIFY command. For output listing files not connected to your terminal, the VFYLIB command honors the page length and print density set for your job using the SET command and the PL and PD symbolic names.

Format:

VFYLIB, lfn₁, lfn₂, lfn₃, NR.

<u>Parameter</u>	<u>Description</u>
lfn ₁	Name of the first file; if this parameter is omitted, VFYLIB assumes file OLD.
lfn ₂	Name of the second file; if this parameter is omitted, VFYLIB assumes file NEW.
lfn ₃	Name of the output file; if this parameter is omitted, VFYLIB assumes file OUTPUT.
NR	If specified, lfn ₁ and lfn ₂ are not rewound after verification.

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.

NOTE

When printing output for text files, VFYLIB uses the first seven characters of the first word of the record as the record name. VFYLIB assumes that the record is in 6-bit display code. Therefore, if you attempt to display VFYLIB output for a record in 6/12-bit display code while your terminal is in ASCII mode, the output display may be garbled. To avoid this problem, you should set your terminal to NORMAL mode before listing VFYLIB output.

Example:

The following are CATALOG listings of file OLD and file NEW.

REC	CATALOG OF OLD NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
1	A	REL	41	7424	yy/mm/dd.
	A				
2	(00)	SUM =	41	LIBRARY =	1
3	B	REL	41	0465	yy/mm/dd.
	B				
4	C	REL	41	5525	yy/mm/dd.
	C				
5	* EOF *	SUM =	143		

CATALOG COMPLETE.

REC	CATALOG OF NEW NAME	TYPE	FILE LENGTH	1 CKSUM	DATE
1	A	REL	41	7424	yy/mm/dd.
	A				
2	B	REL	41	0465	yy/mm/dd.
	B				
3	(00)	SUM =	102	LIBRARY =	1
4	D	TEXT	1	1000	
5	* EOF *	SUM =	103		

CATALOG COMPLETE.

VFYLIB.	OLD FILE = OLD	NEW FILE = NEW	yy/mm/dd.	hh.mm.ss.	PAGE
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, LIBEDIT, and LIBGEN commands. To duplicate the examples, you should execute the jobs in sequence.

Example 1:

The following submit job builds a program library from a replacement file of relocatable binary (REL) records.

```

/JOB
LIBTES1.
/USER
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:

REC	CATALOG OF LGO NAME	TYPE	FILE LENGTH	1 CKSUM	yy/mm/dd. DATE	hh.mm.ss. COMMENTS	PAGE	1
1	A	REL	41	7424	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	A							
2	D	REL	41	6566	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	D							
3	C	REL	41	5525	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	C							
4	B	REL	41	0465	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	B							
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. hh.mm.ss. PAGE 1

```
*BUILD LIBRARY
*B,*_REL/A,B,C,D
```

RECORDS WRITTEN ON FILE TESTLIB yy/mm/dd. hh.mm.ss. 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.

CATALOG OF TESTLIB			FILE 1	yy/mm/dd. hh.mm.ss.	PAGE 1
REC	NAME	TYPE	LENGTH	CKSUM DATE	COMMENTS
1	A	REL	41	7424 yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
2	B	REL	41	0465 yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
3	C	REL	41	5525 yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
4	D	REL	41	6566 yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
5	LIBRARY	OPLD	13	2073 yy/mm/dd.	
6	* EOF *	SUM =	217		

Example 2:

This submit 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
FTNS,L=0.
ATTACH,OLD=TESTLIB.
DEFINE,NEW=TES2LIB.
LIBEDIT.
CATALOG,NEW,R.
/EOR
    SUBROUTINE BONE
    STOP
    END
    SUBROUTINE NEWC
    STOP
    END
    SUBROUTINE D
    STOP
    END
/EOR
*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. hh.mm.ss. PAGE 1

```

*TYPE REL
*I,B,BONE
*D,C

```

RECORDS WRITTEN ON FILE NEW yy/mm/dd. hh.mm.ss. PAGE 2

RECORD	TYPE	FILE	DATE	COMMENT
A	REL	OLD	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
B	REL	OLD	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
INSERTED BONE	REL	LGO	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
DELETED-(C)	REL	OLD		
REPLACED D	REL	LGO	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
ADDED NEWC	REL	LGO	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
ADDED LIBRARY	OPLD	*****	yy/mm/dd.	
EOF		OLD		

The CATALOG output shows the following contents of the new program library.

REC	CATALOG OF NEW NAME	FILE TYPE	1 LENGTH	yy/mm/dd. hh.mm.ss. CKSUM	DATE	PAGE COMMENTS	1
1	A	REL	41	7424	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M	
2	A B	REL	41	0465	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M	
3	BONE	REL	41	4312	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M	
4	BONE D	REL	41	6566	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M	
5	NEWC	REL	41	4432	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M	
6	NEWC LIBRARY	OPLD	15	7700	yy/mm/dd.		
7	* EOF *	SUM =	262				

Example 3:

This submit job uses LIBGEN to generate a user library file from the program library file TES2LIB created in example 2.

```

/JOB
LIBTES3.
/USER
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.

CATALOG OF LIBFILE			FILE	1	yy/mm/dd.	hh.mm.ss.	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,ROUND=	A/	S/ M
	A							
3	B	REL	41	0465	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	B							
4	BONE	REL	41	4312	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	BONE							
5	D	REL	41	6566	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	D							
6	NEWC	REL	41	4432	yy/mm/dd.	SUBROUTINEOPT=0,ROUND=	A/	S/ M
	NEWC							
7	LOADLIB	OPLD	15	0515	yy/mm/dd.			
8	* EOF *	SUM =	275					

Example 4:

This submit job illustrates a method for deleting records from a user library.

```
/JOB  
LIBTES4.  
/USER  
ATTACH,LIBFILE/M=W.  
LIBEDIT,P=LIBFILE,U,Z.#*D,REL/NEWC  
CATALOG,NEW,R,U.  
/EOR
```

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. hh.mm.ss. PAGE 1

*D,REL/NEWC

RECORDS WRITTEN ON FILE NEW yy/mm/dd. hh.mm.ss. PAGE 2

RECORD	TYPE	FILE	DATE	COMMENT
A	REL	LIBFILE	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
B	REL	LIBFILE	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
BONE	REL	LIBFILE	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
D	REL	LIBFILE	yy/mm/dd.	SUBROUTINEOPT=0,ROUND= A/ S/ M
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.

LIBRARY COMMAND

The LIBRARY command specifies a global library set for your current job. Once you specify a global library set, you can execute any program or procedure on a global library file by entering the entry point name for the program or procedure name. Special entry points ARG=, NPC=, RFL=, and SDM= are also honored. (Refer to Volume 4, Program Interface for more information.) A program can load overlays randomly from this global library set.

Format:

LIBRARY,file₁,file₂,...,file_n/directive

<u>Parameter</u>	<u>Description</u>
file _i	Specifies a file containing a user library. The file must be a local file or a system library.
directive	Specifies if file ₁ ,file ₂ ,...file _n are to be added to, to be deleted from or to replace the global library set.
<u>directive</u>	<u>Description</u>
A	Add the specified files to the global library set.
D	Delete the specified files from the global library set.
R	Replace the existing global library set with the specified files. R is the default.

To clear your global library set, specify no parameters:

LIBRARY.

NOTE

A no-auto-drop status is set for any file inserted in the global library set. The no-drop-status is cleared when a library is removed from the global library set. Refer to SETFS command for an explanation of no-auto-drop status.

MAXIMUM SIZE OF GLOBAL LIBRARY SET

The maximum number of libraries allowed in your global library set varies with the number of nonsystem libraries you specify. Figure 15-4 shows the various possibilities:

User libraries	Number of system libraries allowed	Maximum number of libraries
0	24	24
1	13	14
2	2	4

Figure 15-4. Maximum Size of Global Library Set

Nominally, you can never specify more than two user libraries (nonsystem libraries). However, you can circumvent this restriction by making the local file name of a user library the same as that of a system library. The system libraries USRLIB1, USRLIB2, ..., and USRLIB8 are available for this purpose.

LIBRARY SEARCH ORDER

The system searches the libraries in the global library set in the order in which you specified them in the LIBRARY command. If an entry point name appears in multiple libraries, the system uses its first occurrence. If you want to further control the library search order, use the LIBLOAD and SATISFY commands, described in the CYBER Loader Reference Manual.

EXAMPLES

Suppose you want to write a command to yield the date and another to yield your social security number. This example uses a procedure to produce the date and a FORTRAN 5 program to display your social security number.

File DATPROC has the following contents:

```
.PROC,DATE*I.
DISP,$TODAY'S DATE IS:$,DATE+.
REVERT,NOLIST.
---eor---
.PROC,DISP*I,STRING1,STRING2.
NOTE.+STRING1 STRING2
REVERT,NOLIST.
```

File SSPROG has the following contents:

```
PROGRAM SSNUM
PRINT*,'MY SOCIAL SECURITY NUMBER IS 530-82-7161.'
END
```

The following command entries establish your global library set:

```
/get,datproc,ssprog.
/ftn5,i=ssprog,l=0.
0.009 CP SECONDS COMPILATION TIME.
/libedit,n=mylib,u,z.**file datproc
RECORDS WRITTEN ON FILE MYLIB
RECORD      TYPE      FILE      DATE      COMMENT
ADDED      SSNUM      REL      LGO      yy/mm/dd.
ADDED      DATE      PROC      DATPROC
ADDED      DISP      PROC      DATPROC
LIBRARY GENERATION COMPLETE.
/library,mylib
LIBRARY,MYLIB.
```

The FTN5 command creates file LGO and LIBEDIT creates user library MYLIB, containing the procedure and compiled program. The LIBRARY command specifies your global library set.

The following sequence tests your commands:

```
/date
TODAY'S DATE IS: yymmdd
/ssnum
MY SOCIAL SECURITY NUMBER IS 530-82-7161.
0.003 CP SECONDS EXECUTION TIME.
```

Suppose you wish to include three of your own user libraries in the global library set for your current job. You can accomplish this with the following commands:

```
/get, mylib1, mylib2, mylib3.
/library, mylib1, mylib2, mylib3.
TOO MANY USER LIBRARIES SPECIFIED.

/replace, usrlib1 = mylib3.
/library, mylib1, mylib2, usrlib1.
LIBRARY, MYLIB1, MYLIB2, USRLIB1.
```

Note that it was necessary to use the name of a system library USRLIB1 for the third library MYLIB3 in order to use more than two libraries at once.

The following command produces a display of your global library set:

```
/enquire,L

LOADER INFORMATION.
MAP OPTIONS = DEFAULT
DEBUG = OFF
GLOBAL LIBRARY SET IS -
MYLIB1, MYLIB2, USRLIB1
```


This section distinguishes NAM/CCP 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 or TAF, for example). If you are connected to a CDCNET network, refer to the CDCNET Terminal Interfaces Usage manual for a description of terminal input and output. 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 J). 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 connection exists between the terminal and the communications network, including prior to the completion of login.

TERMINALS

IAF and the network provide support for 18 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 teletypes
2	CDC 713-10, 722, 751-1, 752, 756
3	CDC 721, 722-30
4	IBM 2741
5	M40
6	Hazeltine 2000
7	Terminals conforming to the ANSI X3.64 standard (DEC VT100, TeleVideo 924, 950, 955)
8	Tektronix 4010, 4014
9	HASP protocol post-print format control
10	200 User Terminal
11	CDC 714-30
12	CDC 711-10
13	CDC 714-10/20
14	HASP protocol pre-print format control

[†]Terminals of all classes can be connected to NOS. Some terminals in terminal classes 1, 2, 3, and 5 through 8 can be connected to a packet switching network (PSN) with a connection established to NOS.

<u>Class</u>	<u>Terminal</u>
15	CDC 731-12, 732-12, 734
16	IBM 2780
17	IBM 3780
18	IBM 3270

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 J. 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 J). Important characteristics which may vary from terminal to terminal include:

- Abort output block character (AB).
- Backspace character (BS).
- Cancel character (CN).
- Control character (CT).
- End-of-line character (EL).
- End-of-block character (EB).
- Interruption character (BI).
- 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. The information in this section pertains to line mode operations only.

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 .

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 FSE. 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.

LENGTH OF OUTPUT/INPUT LINES

A logical input line can consist of a maximum of 2500 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 2500 characters, the system discards the entire logical line and issues the *OVL* message.

TERMINATING INPUT LINE

The line feed key, which terminates physical lines, and the carriage return key, which terminates logical lines, vary with terminal class. Appendix K lists the carriage return, line feed, and message transmission keys for the various terminal classes. Line feed and carriage return are generic terms which represent characters that terminate a line of text. The logical end-of-line character can be changed by you (refer to the EL command in appendix J), your application, or by your local site.

You must terminate each logical line of input by entering a carriage return. This tells the system that the current logical line is complete. The system may respond by positioning the carriage (or cursor) at the beginning of the next line. You can then enter additional input information on the new line.^{†††}

You continue a logical line on a new physical line by entering a line feed. The system responds by positioning the carriage (or cursor) at the beginning of the next line.

CORRECTING INPUT LINE

You can correct entry errors in the input line before pressing[Ⓢ] 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).

The backspace character can be changed by you (refer to the BS command in Appendix J), your application, or by your site.

[†]This applies only to IAF and can be changed by your site.

^{††}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 J).

^{†††}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 in order to enter input before the system prompts for it).

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 J.

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          You enter the cancel line character.  
*DEL*  
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;

```
WAIT..
```

You should then reenter the unprocessed commands after the system issues its REPEAT.. 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 (interruption and termination sequences).

Terminal definition commands change the terminal's characteristics. Appendix J gives a detailed description of terminal definition commands.

Interactive status commands get an immediate status response from the system. The interactive status commands, ctD, ctE, and ctS, use the network control character defined for your terminal (refer to table J-1). Detailed descriptions of ctD, ctE, and ctS 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 J).

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 enter a carriage return.^{††} 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 **CR**.

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 block, 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 controls described in the following sections, the system resumes output.

ABORTING OUTPUT BLOCK

To abort the current output block, first suspend output by pressing the BREAK key and then enter the abort block 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 J).

^{††}For mode 4 terminals, enter the network 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 block 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 J 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 J 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 J).

CONTINUING A SUSPENDED JOB

After you have suspended your job during program execution by entering the interruption sequence, 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 DO
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 network control character, abort block character, interruption character, or termination character or after entering the cancel character. For further information about these characters, refer to appendix J.

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 64-character set 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	Disconnect terminal from IAF and the host.
0005	Initiate ASCII input mode.
0006	Initiate transparent input mode.
0007	Initiate transparent output mode.
0010	NAM/CDCNET terminal redefinition.
0011	Initiate ASCII output mode.
0013	End-of-string.
0016	NAM/CCP terminal redefinition.

The use of these control bytes is described in detail in section 12 of Volume 4, Program Interface. For additional information, refer to the EFFECT macro description in Volume 4.

CHARACTER SETS

A

A character set is composed of graphic and 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.

- 6-bit display code.
- 6/12-bit display code.
- 7-bit ASCII code.

Display code is a set of 6-bit codes from 00_g to 77_g.

The 6/12-bit display code is a combination of 6-bit codes and 12-bit codes. The 6-bit codes are 00_g through 77_g, excluding 74_g and 76_g. (Refer to Character Set Anomalies for the interpretation of the 00_g and 63_g codes.) The 12-bit codes begin with either 74_g or 76_g and are followed by a 6-bit code. Thus, 74_g and 76_g are considered escape codes and are never used as 6-bit codes within the 6/12-bit display code set. The 12-bit codes are 7401_g, 7402_g, 7404_g, 7407_g, and 7601_g through 7677_g. All other 12-bit codes (74xx_g and 7600_g) are undefined.

The 7-bit ASCII code (as defined by ANSI Standard X3.4-1977) is 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 0000₈ code from the end-of-line byte. The 7-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, and the 00₈ code is undefined. However, if the installation uses the 64-character set, output of 6/12-bit display codes 7404₈ or 6-bit display code 00₈ produces a colon. In ASCII mode for interactive jobs, a colon can be input only as a 7404₈ 6/12-bit display code.

When using either the 63- or 64-character set, the use of undefined 6/12-bit 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).

Translation of 7-bit ASCII to 6-bit display code causes character set folding from the 128-character ASCII set to the 63- or 64-character ASCII subset. The following special character substitutions occur:

7-Bit ASCII		6-Bit Display Code		7-Bit ASCII	
Code	Character	Code	Character	Code	Character
0140	`	74	@	0100	@
0173	{	61	[0133	[
0174		75	\	0134	\
0175	}	62]	0135]
0176	~	76	^	0136	^

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 7-bit ASCII codes and 6/12-bit 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, look up the code in the column listing the appropriate code set and then find 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 6-bit 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-bit 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 7-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 jobs using Line Printers). Only 7-bit ASCII code files can be printed using the ASCII graphic 95-character set. To print a 6/12-bit display code file (usually created by an interactive job in ASCII mode), you must convert the file to 7-bit ASCII code. To do this, you enter the FCOPY command (section 9). The 95-character set is represented by 7-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. In addition, you may choose from one of several subsets of the characters on the print train by specifying the print train image (for information on specifying print train images, refer to the ROUTE command in section 9).

The following table lists the print trains according to the associated ROUTE command EC parameter values.

Print Train	ROUTE EC Values That Will Print	ROUTE EC Values That Will Not Print	Suggested Print Band
0†	NONE, B4, B6	A4, A6, A9	595-1/596-1
1	NONE, B4, B6	A4, A6, A9	595-1/596-1
2†	NONE, B4, B6	A4, A6, A9	595-1/596-1
3†	NONE, B4, B6	A4, A6, A9	595-1/596-1
4	NONE, A4, A6, A9	B4, B6	595-6/596-6
5	NONE, A4, A6	B4, B6, A9	595-4/596-4 or 595-5/596-5
6	A9	NONE, B4, B6, A4, A6	595-6/596-6
7	NONE, B4, B6, A4, A6, A9		595-6/596-6

†These print trains are reserved for future use. If specified, they have the same effect as if print train 1 had been specified.

For PSU printers, the batch character sets and associated print bands are as follows:

<u>Character Set</u>	<u>533/536 Print Band</u>
ASCII graphic 64-character set	530-1
ASCII graphic 95-character set	530-2

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 the repeated listing of the line. The ASCII print trains print an underline (__) instead of the concatenation symbol.

If an unprintable character exists in a line (that is, a 7-bit ASCII code outside the range 0040g through 0176g), 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-bit display code file on a line printer with the ASCII graphic 95-character set, you must convert the 6/12-bit display code file to a 7-bit ASCII code file with the FCOPY command (refer to section 9). The resulting 7-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 Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12- Bit Display Code	7-Bit ASCII Code	ASCII Graphic (64 Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12- Bit Display Code	7-Bit ASCII Code
: colon		00†			# number sign	# number sign	60	60	0043
Display code 00 is undefined at sites using the 63-character set.					[opening bracket	[opening bracket	61	61	0133
A	A	01	01	0101] closing bracket] closing bracket	62	62	0135
B	B	02	02	0102	% percent sign	% percent sign	63	63†	0045
C	C	03	03	0103	: colon	: colon	63	63	0072
D	D	04	04	0104	" quote	" quote	64	64	0042
E	E	05	05	0105	_ underline	_ underline	65	65	0137
F	F	06	06	0106	! exclamation point	! exclamation point	66	66	0041
G	G	07	07	0107	& ampersand	& ampersand	67	67	0046
H	H	10	10	0110	' apostrophe	' apostrophe	70	70	0047
I	I	11	11	0111	? question mark	? question mark	71	71	0077
J	J	12	12	0112	< less than	< less than	72	72	0074
K	K	13	13	0113	> greater than	> greater than	73	73	0076
L	L	14	14	0114	@ commercial at	@ commercial at	74†	7401	0100
M	M	15	15	0115	\ reverse slant	\ reverse slant	75	75	0134
N	N	16	16	0116	^ circumflex	^ circumflex	76†	7402	0136
O	O	17	17	0117	; semicolon	; semicolon	77	77	0073
P	P	20	20	0120	: colon			7404	0072
Q	Q	21	21	0121	% percent			7404	0045
R	R	22	22	0122	` grave accent		74†	7407	0140
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
X	X	30	30	0130	f			7606	0146
Y	Y	31	31	0131	g			7607	0147
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
5	5	40	40	0065	n			7616	0156
6	6	41	41	0066	o			7617	0157
7	7	42	42	0067	p			7620	0160
8	8	43	43	0070	q			7621	0161
9	9	44	44	0071	r			7622	0162
+ plus	+ plus	45	45	0053	s			7623	0163
- dash	- dash	46	46	0055	t			7624	0164
* asterisk	* asterisk	47	47	0052	u			7625	0165
/ slant	/ slant	50	50	0057	v			7626	0166
(opening parenthesis	(opening parenthesis	51	51	0050	w			7627	0167
) closing parenthesis) closing parenthesis	52	52	0051	x			7630	0170
\$ dollar sign	\$ dollar sign	53	53	0044	y			7631	0171
= equal	= equal	54	54	0075	z			7632	0172
space	space	55	55	0040	{ opening brace		61†	7633	0173
, comma	, comma	56	56	0054	vertical line		75†	7634	0174
. period	. period	57	57	0056	} closing brace		62†	7635	0175
					~ tilde		76†	7636	0176
					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 Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12- Bit Display Code	7-Bit ASCII Code	ASCII Graphic (64 Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12- Bit Display Code	7-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 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-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
+ plus	+ plus	+ plus	45	45	0053	12	12-8-6
- dash	- dash	- dash	46	46	0055	11	11
* asterisk	* asterisk	* asterisk	47	47	0052	11-8-4	11-8-4

†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 2 of 3)

CDC Graphic (64 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code	Punch Code	
						026	029
/ slant	/ slant	/ slant	50	50	0057	0-1	0-1
(opening parenthesis	(opening parenthesis	(opening parenthesis	51	51	0050	0-8-4	12-8-5
) closing parenthesis) closing parenthesis) closing parenthesis	52	52	0051	12-8-4	11-8-5
\$ dollar sign	\$ dollar sign	\$ dollar sign	53	53	0044	11-8-3	11-8-3
= equal	= equal	= equal	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
≡ equivalence	# number sign	# number sign	60	60	0043	0-8-6	8-3
[opening bracket	[opening bracket	[opening bracket	61	61	0133	8-7	12-8-2†
] closing bracket] closing bracket] closing bracket	62	62	0135	0-8-2	11-8-2†
% percent sign†	% percent sign†	% percent sign†	63†	63†	0045	8-6	0-8-4
: colon	: colon	: colon	63	63	0072	8-2	8-2
≠ not equal	" quote	" quote	64	64	0042	8-4	8-7
⤵ concatenation	underline	underline	65	65	0137	0-8-5	0-8-5
∨ logical OR	! exclamation point	! exclamation point	66	66	0041	11-0	12-8-7
∧ logical AND	& ampersand	& ampersand	67	67	0046	0-8-7	12
↑ superscript	' apostrophe	' apostrophe	70	70	0047	11-8-5	8-5
↓ subscript	? question mark	? question mark	71	71	0077	11-8-6	0-8-7
< less than	< less than	< less than	72	72	0074	12-0	12-8-4
> greater than	> greater than	> greater than	73	73	0076	11-8-7	0-8-6
≤ less or equal	@ commercial at	@ commercial at	74			8-5	8-4
≥ greater or equal	\ reverse slant	\ reverse slant	75	75	0134	12-8-5	0-8-2
- logical NOT	^ circumflex	^ circumflex	76			12-8-6	11-8-7
; semicolon	; semicolon	; semicolon	77	77	0073	12-8-7	11-8-6
		@ commercial at	74†	7401	0100		
		^ circumflex	76†	7402	0136		
		: colon†		7404†	0072		
		% percent sign		7404	0045		
		` grave accent	74†	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 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-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		
		{ opening brace	61†	7633	0173		
		vertical line	75†	7634	0174		
		} closing brace	62†	7635	0175		
		~ tilde	76†	7636	0176		

†The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

Table A-3. ASCII to 6/12-Bit Display Code Conversion (Sheet 1 of 2)

ASCII Character (128 Character)	7-Bit ASCII Code		6/12-Bit Display Code	ASCII Character (128 Character)	7-Bit ASCII Code		6/12-Bit Display Code
	Octal	Hexadecimal			Octal	Hexadecimal	
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	< less than	0074	3C	72
SO	0016	0E	7656	= equal	0075	3D	54
SI	0017	0F	7657	> greater than	0076	3E	73
				? question mark	0077	3F	71
DLE	0020	10	7660	@ commercial at	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
! exclamation point	0041	21	66	Q	0121	51	21
" quote	0042	22	64	R	0122	52	22
# number sign	0043	23	60	S	0123	53	23
\$ dollar sign	0044	24	53	T	0124	54	24
% percent sign††	0045	25	63††	U	0125	55	25
% percent sign	0045	25	7404	V	0126	56	26
& ampersand	0046	26	67	W	0127	57	27
' apostrophe	0047	27	70	X	0130	58	30
(opening parenthesis	0050	28	51	Y	0131	59	31
) closing parenthesis	0051	29	52	Z	0132	5A	32
* asterisk	0052	2A	47	[opening bracket	0133	5B	61
+ plus	0053	2B	45	\ reverse slant	0134	5C	75
, comma	0054	2C	56] closing bracket	0135	5D	62
- dash	0055	2D	46	^ circumflex	0136	5E	7402
. period	0056	2E	57	_ underline	0137	5F	65
/ slant	0057	2F	50				

†Reserved for network use. Refer to Character Set Tables in this appendix.
 ††The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies in this appendix.

Table A-3. ASCII to 6/12-Bit Display Code Conversion (Sheet 2 of 2)

ASCII Character (128 Character)	7-Bit ASCII Code		6/12-Bit Display Code	ASCII Character (128 Character)	7-Bit ASCII Code		6/12-Bit Display Code
	Octal	Hexadecimal			Octal	Hexadecimal	
` 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	{ opening brace	0173	7B	7633
l	0154	6C	7614	vertical line	0174	7C	7634
m	0155	6D	7615	} closing 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 6-bit 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 6-bit display code.

<u>63-Character Set</u>		
<u>6-Bit Display Code</u>	<u>External BCD</u>	<u>6-Bit Display Code</u>
00	16 (%)	00
33 (0)	12 (0)	33 (0)
63 (:)	12 (0)	33 (0)

<u>64-Character Set</u>		
<u>6-Bit Display Code</u>	<u>External BCD</u>	<u>6-Bit Display Code</u>
00 (:)	12 (0)	33 (0)
33 (0)	12 (0)	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 or write lowercase ASCII or EBCDIC characters, you must assign the tape in binary mode and use FCOPY to read or write the tape.

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 7-bit ASCII character code and 6-bit display code. Table A-5 lists the conversions between the EBCDIC character code and the 6-bit display code. Table A-6 shows the character set conversions between external BCD and 6-bit display code for seven-track tapes.

Table A-4. Nine-Track ASCII Coded Tape Conversion

7-Bit ASCII				6-Bit Display Code		7-bit ASCII				6-Bit Display Code	
Code (Hex)	Char†	Code (Hex)	Char††	Char	Code (Octal)	Code (Hex)	Char†	Code (Hex)	Char††	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	1	11	DC1	1	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
6-bit display code 00 is undefined at sites using the 63-character set.											
3A	:	1A	SUB	:	63	5A	Z	7A	z	Z	32
3B	;	1B	ESC	;	77	5B	[1C	FS	[61
3C	<	7B	{	<	72	5C	\	7C		\	75
3D	=	1D	GS	=	54	5D]	01	SOH]	62
						5E	^	7E	~	^	76
						5F	_	7F	DEL	_	65

†When these characters are copied from/or to a tape, the characters remain the same but the codes change from one code set to the other.

††These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each 7-bit ASCII character is changed to an alternate 6-bit 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, 01g.

†††A 6-bit display code space always translates to a 7-bit ASCII space.

Table A-5. Nine-Track EBCDIC Coded Tape Conversion

EBCDIC				6-Bit Display Code		EBCDIC				6-Bit Display Code	
Code (Hex)	Char†	Code (Hex)	Char††	Char††	Code (Octal)	Code (Hex)	Char†	Code (Hex)	Char††	Char	Code (Octal)
40	space	00	NUL	space	55	C4	D	84	d	D	04
4A	¢	1C	LFS	[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		\	75
6B	,	0C	FF	,	56	E2	S	A2	s	S	23
6C	%	2D	ENQ	%	63	E3	T	A3	t	T	24
6C	z	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
6-bit 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/or to a tape, the characters remain the same (except EBCDIC codes 4A, 4F, 5A, and 5F) but the codes change from one code set to the other.

††These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each EBCDIC character is changed to an alternate 6-bit 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, 01₈.

†††All EBCDIC codes not listed translate to 6-bit display code 55g (space). A 6-bit display code space always translates to an EBCDIC space.

Table A-6. Seven-Track Coded Tape Conversions

External BCD	ASCII Character	6-Bit Display Code (Octal)	External BCD	ASCII Character	6-Bit Display Code (Octal)
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

This 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 Version 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.

All pages are at revision level L (without change bars).

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ABSC, sc.	An informative message indicating the initial service class of your job. sc Service class	None.	CLASS
cmd ACCEPTED..	The network has accepted your terminal definition command cmd.	None.	CCP
ACCESS CATEGORIES NOT VALID FOR JOB.	You attempted to use the SETPFAC command or macro to set one or more permanent file access categories for which you are not validated.	Retry using a set of access categories for which you are validated and that is valid for your job.	PFM
ACCESS DENIED - SECURITY CONFLICT.	You do not have access to this account.	Contact your site administrator.	IAFEX
ACCESS LEVEL NOT VALID FOR FILE.	One of the following occurred: - You attempted to save a local file using an access level for the permanent file that is not valid for the device on which the local file resides. - You attempted to define an existing local file as a direct access permanent file using an access level for the permanent file that is not valid for the device on which the local file resides.	Retry using an access level for the permanent file that is not lower than the access level of the local file.	PFM
ACCESS LEVEL NOT VALID FOR FILE filename AT nnn.	The access level of the job is not valid for the device on which the file filename resides.	None. You may not access this file while your job is at the current access level.	IMS
ACCESS LEVEL NOT VALID FOR JOB.	One of the following occurred: - You attempted to save a local file at an access level that is not valid for your job. - You attempted to define an existing local file as a direct access permanent file at an access level that is not valid for your job. - You attempted to set an access level for a permanent file that is not valid for your job using the SETPFAL command or macro.	Retry using a valid access PFM level.	PFM
ACCESS LEVEL NOT VALID FOR JOB, filename AT nnn.	The access level specified for file filename is not within the job validations.	None. You may not access this file while your job is at the current access level.	IMS
ACCESS LEVEL NOT VALID ON PF DEVICE.	On a SAVE, DEFINE, or SETPFAL command or macro, you specified an access level that is outside the range of access levels allowed on your permanent file device.	Inform your site security administrator that your permanent file device is not valid for all the access levels for which you are validated.	PFM
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
ADDING FILE lfn TO DATABASE.	File pfn is being read; RECLAIM is adding information for this file to your database.	None.	RECLAIM
ADDING TAPE vsn TO DATABASE.	Tape with VSN vsn is being read; RECLAIM is adding information for this tape to your database.	None.	RECLAIM
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 address.	LDC read an address on a correction statement that is greater than or equal to the job's field length. The correction statement is ignored and LDC continues.	Correct the correction statement and retry.	CPMEM
ADDRESS OUT OF RANGE.	You tried to set a breakpoint address which was out of range.	Correct address and retry.	DEBUG
AESR, 123456.789UNTS.	Informative message issued to job dayfile at job termination, listing the number of SRUS used by the job since the last ACSO, ACSR, or AUSR message.	None.	1AJ
AFD - INPUT FILE NOT FOUND.	The file specified by the I-filename parameter is not a local file.	Retry using a local file.	DAYFILE
AFD - 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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ALL EQ-S CHECKPOINTED. RECOVERY ABORTED.	In the process of aborting a CM recovery, 1CK was called to process all outstanding checkpoint requests.	Level 0 deadstart is required.	1CK
filename 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
ALTERNATE STORAGE ERROR.	It is not possible to drop the disk space associated with the specified permanent file, since an error exists on the alternate storage copy of the file.	Inform site analyst.	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
APPLICATION NAME IS REQUIRED.	You entered an APPSW command that did not specify a secondary application.	Reenter the command with the appl parameter specified.	TLX
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 was disconnected.	Check the accuracy of the entry. If problems persist, contact installation personnel.	NETVAL
ARG. ERROR.	Your program specified incorrect arguments when it called LDR to load an overlay.	Correct and retry.	1AJ
ARGUMENT ERROR.	The command syntax is incorrect. Specification of duplicate parameters (more than one occurrence of a keyword) or mutually exclusive parameters (such as MT/NT, CB/CK, FI/L, R/W, PO=R/PO=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.	Reenter command using the correct parameters and syntax.	COPYB CPMEM FOTD LDI LO72 RECOVER RESEX SUBSYST
ARGUMENT ERRORS.	The ENQUIRE command's syntax is incorrect.	Check parameters on command and retry.	ENQUIRE
ARITHMETIC ERROR.	An arithmetic hardware error occurred.	Inform site analyst.	RECLAIM
ARITHMETIC INDEFINITE.	The CPU floating-point arithmetic unit attempted to use an indefinite operand.	Analyze the job output and dumps to determine the cause of the error.	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 of the error.	1AJ
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 of the error.	1AJ
ASL = nnnnnn, JSL = nnnnnn.	Informative message issued after a SETASL or SETJSL command indicating the current value of the account block SRU limit (ASL) and job step SRU limit (JSL).	None.	CONTROL
ASL = nnnnnn, JSL = nnnnnn.	Informative message issued after a SETASL or SETJSL command indicating the current value of the account block SRU limit (ASL) and the job step SRU limit (JSL).	None.	CONTROL
est, ASSIGNED TO filename.	Informative message specifying EST ordinal est was assigned to file filename.	None.	LFM
ttest, ASSIGNED TO filename.	Equipment type tt with EST ordinal est was assigned to file filename.	None.	LFM
jsn ASSIGNS EXCEED DEMANDS.	System error. The resources actually assigned to the job with the specified job sequence name exceed the resources demanded on a RESDURC command.	Inform site analyst.	RESEX
ATTRIBUTE COMMANDS MUST BE OF ONE NETWORK TYPE.	Terminal attributes unique to NAM/CCP and to NAM/CDCNET were specified on the same TRMDEF command.	On the TRMDEF command, check to ensure that all terminal attributes are supported by your network type.	TRMDEF
BAD CATALOG/PERMIT SECTOR.	PFM has encountered a catalog or permit sector which does not have a valid sector length. This indicates corruption of system areas on disk.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLDAD on the device.	PFM
BAD DATA IN OLDPL HEADER.	The file is a program library but is not of a recognizable format.	Try rebuilding the file if ITEMIZE possible. Inform site analyst.	

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
BAD FCOPY CONVERSION TABLE.	FCOPY tables are in error.	Write a PSR and include support materials to allow CDC to duplicate the problem.	FCOPY
nnnnnnnnnn BAD FORMAT BLOCKS.	Blocks not meeting X or SI-coded conversion requirements have been encountered on input.	Check tape format for errors.	COPYB
BINARY SEQ. ERROR, RECxxxx CDyyyy.	A binary command was found to be out of sequence and the job was terminated without EXIT processing. Error is on command yyyy of record xxxx.	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, filename AT address.	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 NUMBER nnnnnn, filename AT address.	A tape error has occurred at block nnnnnn. A second message generally accompanies this message, giving more detail as to the nature of the error.	None.	1MT
BLOCK SEQUENCE ERRDR, filename AT address.	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 1 format tapes only).	Ensure accuracy of format parameter (F) and label parameter (LB) on command or macro. If problem still exists, clean tape drive or use a new tape.	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 COPYB command without the E or B conversion parameter specified.	Refer to the COPY or COPYB command description, correct the command, and retry.	COPYB
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 COPYB exceeds maximum for that format (5120 characters).	Reduce the block size or character counts.	COPYB
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, 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.	COPYB
BLOCK TOO LARGE, filename AT address.	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) and label parameter (LB) command or macro. If the problem still exists, clean the tape drive or user a new tape.	1MT
BOT/EOT ENCOUNTERED, filename AT address.	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.	Contact CYBER Software Support.	1AJ
BREAKPOINT WITHIN DEBUGGER.	You specified a breakpoint address within the breakpoint itself.	Correct address and reenter command.	DEBUG
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 the operation does not reference an address beyond end of buffer or job's field length.	1AJ
BUFFER ARGUMENT ERROR, filename AT address.	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 - PQSMF issued and no HDR1 label found in FET or extended label buffer Refer to Volume 4 of the NOS Version 2 Reference Set. If the message was caused by the FCOPY command, it indicates that the product of the fixed line length (FL parameter) and lines per block (LB parameter) was greater than 3840.	Examine program to determine error.	1MT FCOPY
BUFFER ARGUMENT ERROR ON filename AT address.	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 of the NOS Version 2 Reference Set.	Examine program to determine error in buffer pointers.	1MS

<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 1008.	Take a deadstart dump, write a PSR, and send the dump and PSR to CDC along with any support materials which will allow CDC to duplicate the problem.	SLL
BUFFER CONTROL WORD ERROR, filename AT address.	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 OVERFLOW - ALL FILES LISTED BUT NOT SORTED.	Informative output file message indicating that you have more permanent files than can fit in the sort buffer. The system listed all of the files but not in alphabetical order.	None.	CATLIST
BUFFER SPACE ERROR (TOV).	Internal error.	Submit a PSR with information on how to duplicate the problem.	IEDIT
pfn BUSY.	The specified direct access file is currently attached in an incompatible mode or is already being accessed by 4095 users in your specified mode (R, RA, RU, or RM).	Reissue ATTACH until file becomes available or issue ATTACH specifying WB or NA option.	PFM
CALLING JOB MAY NOT DROP ITSELF.	You attempted to drop your current job.	Do not attempt to drop your current job.	DROP
CANNOT ATTACH/GET FILE fffffff-FILE SKIPPED.	An error occurred when attempting to get or attach file fffffff. The file may have been attached or purged by another job.	Check your dayfile for a specific error message. If you do not find one, inform the site analyst. It's normal for this message to appear, referencing the RECLAIM database, during a DUMP operation.	RECLAIM
CANNOT CHANGE CLASS OF ON-LINE JOB.	You extended a CLASS command with a JSN parameter that attempted to change the service class of an online job. You cannot use the JSN parameter to change the service class of an online job.	None.	CLASS
CANNOT CHANGE CLASS OF SUBSYSTEM.	The service class of a job executing at the subsystem service class cannot be changed.	None.	CLASS
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 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	Separator following verb (directive name) 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	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 command or line, make last nonblank character a valid separator.	CCL
CCL102- EQUAL SIGN MUST FOLLOW FIRST SYMBOLIC NAME	The first parameter in a SET command is a symbolic name to be equated. 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- COMMAND INCOMPLETE	A terminator was detected immediately following the command name.	Check command format (refer to section 6) and rewrite command, using a comma or left parenthesis after the command name.	CCL
CCL104- EXPRESSION AND COMMAND TERMINATED BY terminator	An expression or a command name was followed by a command terminator (period or right parenthesis) instead of a comma. Any of the following conditions may produce this error. - 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 the separator following a command name was a left parenthesis, an unmatched right parenthesis may appear matched. For example, WHILE(R1+R2)+3, LABEL is	Implement the action that applies to the condition that produced the error. - Place a comma after the expression or the command name and add a label parameter. - Delimit operator with periods on both sides. - If error was caused by a misleading left parenthesis following the command name, add	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
	terminated by right parenthesis. This message is informative and is issued in conjunction with message CCL123.	a comma immediately after the command name. Otherwise, recheck all parentheses and match unmatched right parenthesis.	
CCL120- ALL COMMANDS SKIPPED - xxxxxx	In process of skipping, end of command record was reached. Any of the 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.	Implement the action that applies to the condition that produced the error. - 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	The 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	The commands ELSE, ENDIF, ENDW, SKIP, and WHILE require a 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	Dayfile message which precedes CCL124 is an error message which was caused by the command printed after CCL124.	Refer to the error message which precedes this in the dayfile for cause and suggested action.	CCL
CCL125- NUMBER OF COMMANDS SUPPRESSED=x	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- COMMAND TERMINATOR MUST FOLLOW LABEL	Either of the following conditions may produce this error. - A label string is last parameter on an ELSE, ENDIF, ENDW, IF, SKIP, or WHILE command. It must be followed by a period or a right parenthesis. - An expression contained a comma, and the character string that followed was assumed to be a label string.	Implement the action that matches the error. - 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	The label string parameter on an ELSE, ENDIF, ENDW, IFE, SKIP, or WHILE command is greater than xx characters. xx is the 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 INCORRECT OPERATOR x	x is not a valid operator in the 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)	The format of the expression is incorrect. The 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(filename,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 - (DT, LEN, NUM, SS, STR, STRB, STRD)	You made one of the following errors: - 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.	Implement the action that matches the error. - Correct format of function. - \$-delimit special characters.	CCL
CCL153- INCORRECT FUNCTION CALL (WITHIN FILE FUNCTION)	The LEN, FILE, NUM, STR, STRB OR STRD function, or SS symbolic name appears within a FILE function.	Restructure the expression so that the function or symbolic value in question is not within a FILE function.	CCL
CCL154- INCORRECT EXPONENT	A negative exponent in an expression is incorrect.	Remove minus sign preceding exponent.	CCL

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
CCL155- OPERATOR OR OPERAND SEQUENCE ERROR	You made one of the following errors: - Sequence of adjacent operators is incorrect. 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 incorrect.	Implement the action that matches the error. - Use parentheses to separate adjacent operators. - Use an operator to separate operand and parenthesis.	CCL
CCL156- STRING TOO LONG -string	The listed character string (string) is too long. A character string may not be longer than 10 characters, excluding \$ delimiters for literals, unless it appears within the STR function.	Shorten character string to 10 characters or less, or enclose it within the STR function.	CCL
CCL157- UNKNOWN NAME -string	The alphanumeric character string string is not a symbolic name recognized by the system.	Replace string 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, LEN, NUM, STR, STRB, STRD)	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 INCORRECT IN SET COMMAND	The first parameter in the SET command was not one of the following symbolic names: DSC, EF, EFG, R1, R2, R3, RIG, or SS.	Replace first parameter with a valid symbolic name.	CCL
CCL160- NUMERIC OR LITERAL ILLEGAL (IN FILE FUNCTION)	The expression of a FILE function may contain only symbolic names defined for the function.	Remove numeric or literal character string. Refer to section 6 for a list of FILE symbolic names.	CCL
CCL161- STACK OVERFLOW	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. (that is, > instead of .GT.).	Use period bracketed operators.	CCL
CCL163- SUBSYSTEM REFERENCE ERROR	The system does not recognize the name associated with the symbolic name SS.	Refer to section 6 for a list of valid names.	CCL
CCL164- PAGE PARAMETER ERROR	An error was detected when setting the job default page size symbols PD, PS, PW.	Check to see if the value in the expression is within the range of the symbol.	CCL
CCL165- CONCATENATION MUST HAVE STRING OPERANDS	One or both of the operands for a string concatenation (//) were not within one of the string functions.	Enclose the operand in question within an STR, STRB, or STRD function.	CCL
CCL200- PROCEDURE NESTING LEVEL xx EXCEEDED	The current procedure call forced procedure nesting to exceed the 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	A procedure name cannot be greater than 7 characters.	Rename procedure.	CCL
CCL202- PROCEDURE FILE IS EXECUTE ONLY MODE	The procedure file was attached prior to the call in execute-only mode. The procedure file may have been saved prior to the call in execute-only mode.	Return file and attach with default mode.	CCL
CCL203- PROCEDURE FILE NAME NOT SPECIFIED OR INCORRECT	You made one of the following errors: - File name specified on procedure call was greater than 7 characters. - pfile parameter on procedure call was null. Your site does not have a default for pfile.	Implement the action that matches the error. - Specify a file name with seven or fewer characters. - Specify a file name.	CCL
CCL204- MULTIPLE EQUIVALENCE SPECIFICATIONS FOR xx	The keyword xx 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	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
CCL206- SYMBOLIC SPECIFICATION INCORRECT xxx	A parameter, xxx, on the procedure call is followed by a plus sign. A plus sign indicates that the system is to convert the 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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL207- PROCEDURE NAMED BEGIN IS INCORRECT	A procedure must not be named BEGIN.	Select another procedure name.	CCL
CCL211- SPECIFICATION EXCEEDS XX CHARACTERS	The 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 INCORRECT string s	s is an incorrect separator and string is the character string that precedes s. Any of the following conditions produces this error. <ul style="list-style-type: none"> - 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 incorrect separator s, the string is null. - #DATA, #FILE, or #PRIMARY has been specified on the procedure call. 	Implement the action that applies to the condition that produced the error. <ul style="list-style-type: none"> - Change separator s to a comma. - \$-delimit character string. - Remove #DATA, #FILE, or #PRIMARY from procedure call and specify it on procedure header directive. 	CCL
CCL213- END OF INFORMATION ON PROMPTED INPUT	The user assigned INPUT to a file that is at the end of information and CCL needs to prompt for a parameter. The INPUT file probably contains faulty data.	Correct the file contents or supply the CCL parameters interactively.	CCL
CCL214- NOT ENOUGH FL - CHECK *CCFL* IN *PPCOM*	There is insufficient field length to execute CCL when called by VALEX during prologue processing. This is caused by the symbol CCFL in PPCOM being set too low.	Modify the value of CCFL to be greater than the FL required for the CCLBRWE overlay of CCL, reassemble NOSTEXT and VALEX, and rebuild the NOS system.	CCL
CCL230- PROCEDURE FILE NOT FOUND	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	The local or permanent file indicated on the procedure call was found, but the system could not find the procedure on the file.	Check procedure name for errors and correct. If name is correct, check file for procedure.	CCL
CCL232- PROCEDURE FILE IS BUSY	The CCL ATTACH found a file busy status on the procedure file.	Clear file busy condition and resubmit 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	The system could not find procedure xxx, specified on the call-by-name command, on the 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	The number of characters in a keyword on the procedure header directive exceeds xx, as defined by installation.	Define a keyword with xx or fewer characters.	CCL
CCL236- SPECIAL DEFAULT SPECIFICATION UNKNOWN	The number sign [#], which specifies a special default option on the procedure header directive, must be followed by FILE, PRIMARY, or DATA.	If number sign is part of a character string, enclose the character string with dollar signs. If a special default option is desired, follow the number sign with either DATA, FILE, or PRIMARY.	CCL
CCL237- SEPARATOR FOLLOWING SECOND DEFAULT IS */*	In form keyword=default1/default2/ on the procedure header directive, the second / is incorrect.	If / is part of default2, enclose the character string with dollar signs. If not, either remove it or replace it with a comma or period.	CCL
CCL238- FORMAL PARAMETER LIMIT XX EXCEEDED	The number of parameters on the 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 MUST BE TERMINATED BY A PERIOD	The procedure header directive was not terminated by a period, and no commands were found after the header directive.	Terminate procedure header with a period and add at least one command to procedure.	CCL
CCL240- *MAY NOT APPEAR IN FORMAL PARAMETER NAME	A keyword containing an asterisk was found on the procedure header.	Redefine the keyword without any asterisks.	CCL
CCL249- DATA FILE LFN EXCEEDS 7 CHARACTERS - filename	The file name specified on a .DATA directive is longer than seven characters.	Specify a shorter file name.	CCL
CCL250- CONCATENATED STRING EXCEEDS XX	Use of a right arrow (or the underline character in ASCII) produced a linked string of more than xx characters or a string of xx characters with a following separator. xx is equal to IP.SCL. (Default is 150).	Reduce number of linked characters.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL261- DATA DIRECTIVE SPECIFIED RESERVED FILE - filename	The file name that you specified on a .DATA directive is a working file.	Select another file name.	CCL
CCL262- PROCEDURE CONTAINS NO COMMANDS	A procedure should contain at least one command.	None.	CCL
CCL263- PROC BODY STRING EXPANDS .GT. xx	After parameter substitution, a single contiguous string in the procedure body is longer than the maximum line length. xx is equal to IP.SCL. (Default is 150).	Rewrite the string using fewer characters or fewer concatenated strings.	CCL
CCL261- ENDW WITH MATCHING LABEL FOUND	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 the WHILE expression is true, normal processing continues with the command following the WHILE command. If the 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, PRINT SKIPPED COMMANDS	To assist in isolation of WHILE command errors such as CCL261, all skipped commands are printed in the job dayfile until the 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 OCCURRED DURING ENDW	An external abort occurred during the 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	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	A REVERT command cannot appear in commands of the job file.	Remove REVERT command.	CCL
CCL272- INCORRECT REVERT PARAMETER - param	The parameter param was used on the REVERT command. The REVERT command allows only ABORT, EX, or NOLIST as parameters.	Replace parameter param with a valid specification.	CCL
CCL273- EXPECTING (OR) AFTER REVERT, EX	REVERT, EX, does 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	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 CONFLICTS WITH POST RADIX OF B	An 8 or 9 is incorrect in an octal number.	Remove any 8 or 9 if the number is octal, or remove post radix B if number is decimal.	CCL
CCL303- LITERAL NOT TERMINATED	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
CCL320- CORRECT CCL320- ENTER	This message precedes a system prompt for a parameter when a batch mode procedure call contains a parameter entry in error.	None.	CCL
CCL321- EXPECTING (AFTER PARAMETER: ON HEADER	In a parameter-prompting procedure, the equal sign after a parameter name or description must be followed by a left parenthesis to indicate the start of a checklist.	Correct procedure header and resubmit.	CCL
CCL322- END HEADER WITH . OR USE , / \ BEFORE PARAM	A parameter definition or the #I after the procedure name 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, patterns, and description text 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
CCL325- *S RANGE MUST BE .GE.1 AND .LE.40	The range for the *S checklist pattern was zero or greater than 40.	Correct the *S range.	CCL
CCL326- I OR M MUST FOLLOW PNAME= ON HEADER.	The procedure name must be followed by #I for interactive procedures or by #M for menu procedures.	Insert I or M after the * following the procedure name.	CCL

	<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL327-	EXPECTING VALUE AFTER PATTERN: ON HEADER	The syntax immediately following the equal sign in the checklist is incorrect. The system expects an alphanumeric value, a literal, or a special name as a value.	Correct header and resubmit.	CCL
CCL328-	HEADER PARAMETER DESCRIPTION TOO LARGE.	A procedure header parameter description exceeds 40 6/12-bit display code or 80 8-bit display code characters.	Reduce the size of the description string.	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-	EXPECTING , , 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-	EXPECTING , 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 *N(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 PATTERN IN CHECKLIST	A single checklist may not contain more than one *K, *N/*D, or *R.	Correct procedure header and resubmit.	CCL
CCL333-	DUPLICATE *N/*D PATTERN IN CHECKLIST			
CCL333-	DUPLICATE *R PATTERN IN CHECKLIST			
CCL334-	MISSING QUOTE - PROMPT/TITLE/ DESCRIPTION	A prompt, title, or description string on the procedure header must begin and end with a quote and cannot extend onto another line.	Insert quotes.	CCL
CCL335-	DESCRIPTION MISSING] BRACKET.	The terminating right bracket (]) on a description string was not found.	Start the string with a left bracket ([) and terminate it with a right bracket (]).	CCL
CCL336-	PATTERN OR VALUE ON PROC HEADER .GT. 40	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
CCL337-	EXPECTING , OR) AFTER CHECKLIST PATTERN	An *R checklist pattern can only be followed by another pattern preceded by a comma or a right parenthesis indicating the end of the checklist.	Correct procedure header and resubmit.	CCL
CCL352-	HELP PARAMETER NAME NOT FOUND	If a comma follows the .HELP directive, the system expects to find a parameter name or the parameter NOLIST.	Correct directive and resubmit.	CCL
CCL353-	ONLY COMMA ALLOWED AFTER .HELP	The system found something other than a comma after .HELP directive.	Correct directive and resubmit.	CCL
CCL354-	.HELP DIRECTIVE INCORRECTLY WRITTEN	The .HELP directive can have one of four formats: .HELP. .HELP,keyword. .HELP,keyword,NOLIST. .HELP,,NOLIST.	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 DIRECTIVES 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
CCL373-	EXPECTING (OR , AFTER .IF .ELSE .ENDIF.	A .IF, .ELSE, or .ENDIF directive in the procedure body must have a left parenthesis or a comma immediately following the directive name.	Insert a left parenthesis or a comma.	CCL
CCL374-	EXPECTING PERIOD AFTER .EX	A period must follow the .EX directive name in the procedure body.	Insert a period after the directive name.	CCL
CCL375-	COMMAND IS NULL ON .EX DIRECTIVE	You have not specified a command on the .EX directive.	Specify a command on the directive.	CCL
CCL376-	.IF DIRECTIVE HAS INCOMPLETE EXPRESSION	A .IF directive in the procedure body is incomplete.	Complete the expression and add a label or command to the directive.	CCL
CCL377-	EXPECTING ,) OR . AFTER .IF EXPRESSION	A comma, period, or right parenthesis must follow the expression on a .IF directive in the procedure body.	Insert a comma, period, or right parenthesis after the expression.	CCL
CCL378-	, OR (MUST FOLLOW .DIRECTIVE NAME	A procedure directive is missing the comma or left parenthesis separator.	Place separator after the directive name.	CCL
CCL379-	NO .ENDIF FOR .IF/.ELSE LABEL ***** - LAST LINES SKIPPED.	A .ENDIF directive with a label matching the .IF/.ELSE directive label ***** was not found.	Add a .ENDIF directive or correct label on existing .ENDIF.	CCL

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
CCL380- EXPECTING TITLE OR , AFTER *M ON HEADER	A title, if there is one, or a comma was not found immediately following the *M on a menu procedure.	Place a title surrounded by quotes after the *M or else place a comma after *M.	CCL
CCL381- KEYWORD MISSING ON MENU HEADER	You must specify a formal parameter keyword after the procedure name in the procedure header of a menu-generating procedure.	Insert a parameter keyword.	CCL
CCL382- PROMPT ON PROC HEADER IS TOO LARGE	A prompt/description string on your procedure header is greater than 40 8/12-bit display code or 80 6-bit display code characters.	Reduce the size of the prompt string.	CCL
CCL383- MISSING SELECTION ON MENU HEADER	Your procedure header checklist has a null specification.	Specify a checklist with integer selections.	CCL
CCL384- TITLE ON PROC HEADER IS TOO LARGE	A title string on a menu procedure header is greater than 40 8/12-bit display code or 80 6-bit display code characters.	Reduce the size of the title string.	CCL
CCL385- EXPECTING = AFTER KEYWORD ON HEADER	An equal sign must follow the parameter keyword on your procedure header.	Insert an equal sign after the parameter keyword.	CCL
CCL386 - EXPECTING COMMA AFTER HEADER TITLE	A character other than a comma was found following the title on the procedure header.	Place a comma after the title.	CCL
CCL387- BEGINNING OF HEADER CHECKLIST NOT FOUND	Either your procedure header does not contain a checklist or the checklist begins with a special character other than a left parenthesis.	Specify a checklist beginning with a left parenthesis.	CCL
CCL388- EXPECTING INTEGER HEADER SELECTION OPTION	You specified a noninteger selection in your procedure header checklist. A selection must be a 1- to 10-digit integer.	Make all menu selections integers.	CCL
CCL390- EXPECTING , OR) AFTER CHECKLIST SELECTION	You specified a special character other than comma or right parenthesis after a menu selection on your procedure header.	Replace the special character with a comma or terminate the checklist with a right parenthesis.	CCL
CCL392- EXPECTING , OR) AFTER HEADER PROMPT	You specified a special character other than comma or right parenthesis after a prompt/description string on your procedure header.	Replace the special character with a comma or terminate the checklist with a right parenthesis.	CCL
CCL393- EXPECTING PERIOD AFTER HEADER CHECKLIST	You specified either no period or a special character other than a period following a procedure header checklist.	Insert a period after the checklist.	CCL
CCL394- TOO MANY HEADER SELECTIONS SPECIFIED	You specified more than 50 menu selections in your procedure header checklist.	Reduce the number of selections.	CCL
CCL395- HEADER SELECTION EXCEEDS 10 CHARACTERS	In your procedure header, a menu selection number contains more than 10 digits.	Use 10 or fewer digits for your selection.	CCL
CCL421- EXPECTING , OR (AFTER .DIRECTIVE	A character other than a comma [,] or right parenthesis [)] was specified between the directive name and its parameters.	Place a comma [,] or right parenthesis [)] after the directive name.	CCL
CCL440- EXPECTING INHIBIT CHARACTER - .IC DIRECTIVE	The .IC directive is missing a special character which would replace the current inhibition character.	Specify a special character as a replacement.	CCL
CCL441- EXPECTING CONCATENATION CHARACTER - .CC	The .CC directive is missing a special character which would replace the current concatenation character.	Specify a special character as a replacement.	CCL
CCL442- EXPECTING TERMINATOR ON DIRECTIVE	A CCL directive was not terminated by a period or right parenthesis.	Terminate the directive with a period or a right parenthesis.	CCL
CCL443- EXPECTING ON/OFF .EXPAND DIRECTIVE	The only arguments allowed for .EXPAND are ON and OFF. The argument must be preceded by a comma or left parenthesis and followed by a period or right parenthesis. The argument may be omitted.	Correct the argument or syntax and resubmit.	CCL
CCL480- USE NUMERIC MIN..MAX VALUES IN RANGE	A non-numeric minimum or maximum range was found within a checklist on the procedure header.	Use a numeric minimum or maximum range.	CCL
CCL481- USE CONSECUTIVE PERIODS MIN..MAX IN RANGE	A range was specified on the procedure header without 2 consecutive periods separating the minimum and maximum.	Use consecutive periods to specify a range.	CCL
CCL482- MINIMUM EXCEEDS MAXIMUM IN RANGE	A checklist range on the procedure header has a minimum larger than the maximum.	Change either the minimum or maximum and resubmit the procedure.	CCL
CCL483- *F OR *P RANGE MUST BE .GE. 1 AND .LE. 7	The *F and *P checklist patterns specify a file name, which cannot exceed 7 characters.	Correct range maximum to 7 or less.	CCL
CCL484- *A RANGE MUST BE .GE. 1 AND .LE. 40	The range for the *A checklist pattern was zero or greater than 40.	Correct *A checklist range.	CCL
CCL485- USE *SN(SET) OR *SN/M FOR SET ATTRIBUTES	The *S checklist pattern has two forms. The first is *Sn(SET) and the second is *Sn/m (described as *Sm..n/k in section 4). The *S specification on the procedure header does not conform to either.	Correct the *S checklist pattern.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CCL486- USE M=A,D,B,AD,AB for *SN/M TYPE SET	When the *Sn/m checklist pattern [described as *Sm..n/k in section 4] is used, only A, D, B, AD, or AB may be substituted for m. For example: S2..10/AD	Correct the *S checklist pattern.	CCL
CCL487-RANGE MAXIMUM MUST BE.GT.ZERO	A checklist range on the procedure header has a maximum of zero.	Change the maximum to a value greater than zero.	CCL
CF0 FROM JOB jsn .	The CF0 command was issued to the job by the job jsn.	None.	IRI
CF0 FROM OPERATOR	The CF0 command was issued to the job by the console operator.	None.	IRI
CHANNEL MALFUNCTION, filename AT address.	A hardware malfunction occurred.	Inform site analyst.	1MT
CHARGE COMMAND REQUIRED.	Dayfile message indicating that a CHARGE command must be executed before the requested command can be executed.	Enter a valid CHARGE command and then reenter the original command.	VALEX
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.	VALEX
CHARGE NOT VALID.	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 and project exist but are inactive.	Check correctness of charge and project number. Use another charge and project number.	VALEX
CHARGE NOT VALID 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.	VALEX
CHARGE NUMBER EXPIRED.	Dayfile and output file message indicating the charge number expiration date has occurred.	None.	VALEX
CHARGE = chargenumber, PROJECT = projectnumber.	Message issued to the job dayfile any time a CHARGE command is processed. The message is also issued when charge and project numbers are set automatically (at job initiation).	None.	VALEX
CHARGE RESTRICTED TO DEFAULT.	You may not specify charge and project numbers other than the default ones for this user.	Specify the default charge and project numbers.	VALEX
CHECKPOINT nnnn COMPLETE.	Indicates that checkpoint nnnn has completed. Issued if only one checkpoint file is present.	None.	CHKPT
CHECKPOINT nnnn COMPLETED TO filename.	Indicates that checkpoint nnnn has been completed to file filename. Issued if alternate CB checkpoint files are used.	None.	CHKPT
CHECKPOINT FILE ACCESS LEVEL ERROR.	In a secure system, the checkpoint file access level is not the same as the job access level limit. This is to insure that the checkpoint file is secure enough to contain all local files. The access level of the checkpoint file is not changed to this level since this is not possible for magnetic tape files or direct access permanent file, and these are the usual file types used for checkpoint files.	Insure that the access level of the checkpoint file is the same as the job access level upper limit.	CHKPT
CHECKPOINT FILE ERROR.	For RESTART, this message indicates that the structure of the file declared to be the checkpoint file does not correspond to that of a checkpoint file.	Correct specification of checkpoint file.	RESTART
CHECKPOINT FILE ERROR.	During a checkpoint operation, more than two checkpoint files were present or two checkpoint files of type CK were present.	Correct specification of checkpoint file.	CHKPT
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
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
CID ERROR.	Updating of resource file returned error status other than end-of-device.	Rerun the job. If the problem reoccurs, inform a site analyst, write a PSR and include support material so that CDC can duplicate the problem.	RESEX
CID ERROR nn ON filename.	A CID error with octal error code nn occurred while RESEX was trying to open tape file filename.	Use a different tape.	RESEX
CKP REQUEST.	A checkpoint has been initiated.	None.	CHKPT
CLASS ARGUMENT ERROR.	One of your arguments is not correct.	Check command description, then retry.	CLASS

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
CLASS COMPLETE.	An informative message indicating that your attempt to change a job's service class was successful.	None.	CLASS
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 of the error.	1AJ
CM OR EC REQUEST EXCEEDS MAXIMUM.	Field length of more than 377700 octal (CM) or more than 77777 octal (EM) was specified in an RFL command.	Retry with lower values.	CONTROL
CM OUT OF RANGE.	The program referenced an address outside the job CM field length.	Analyze the job output and dumps to determine the cause of the error.	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
COMMAND ARGUMENT ERROR.	An incorrect parameter was entered for the command.	Correct parameter and retry.	IEDIT
COMMAND ERROR.	An incorrect parameter was specified on the command.	Correct command and retry.	EDIT LISTLB
COMMAND FILE ERROR.	An unrecovered disk error occurred while the system was processing the command.	Rerun the job.	LFM
COMMAND LIMIT.	The system has processed the maximum number of commands allowed in one of your jobs (refer to the LIMITS command).	If you must rerun the job, restructure it such that it requires fewer commands.	1AJ
COMMAND NOT ALLOWED IN PROLOGUE/EPILOGUE.	A CHARGE command was placed in a system prologue or in a project prologue or epilogue.	Remove the CHARGE command from the system prologue or the project prologue or epilogue.	VALEX
COMMAND NOT PROCESSED DUE TO OVERLAP.	You have specified overlapping line numbers on the edit command.	Correct line numbers and retry.	IEDIT
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.	You entered a command string longer than 80 characters.	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.	1RI
COMMENT FROM OPERATOR	A COMMENT command was issued to the job by the console operator.	None.	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, filename AT address.	The system was unable to connect a peripheral device.	Inform site analyst.	1MT
CONNECTION REJECTED.	The requested application is already servicing the maximum number of terminals.	Request a different application or log off and try later.	NETVAL
COPY COMPLETE.	Copy termination condition was satisfied before EOI was encountered.	None.	COPYB COPYC
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.	COPYB
COPY INDETERMINATE.	Your copying operation with S, L, or F tape is unpredictable. Only the COPY command supports these formats.	Retry copy using the COPY command.	COPYB
COPY NOT ALLOWED OF RESERVED FILE NAME filename.	An attempt was made to make a local copy of a file with the same name as a RECLAIM scratch file, dump file, INPUT, or OUTPUT.	Retry using a different local file name.	RECLAIM
COPYING -- type / name	The record with the type and name on the old file was copied to the new file.	None.	COPYL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
COPYL COMPLETE.	All records on the old file have been processed.	None.	COPYL
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
CORRECTION IDENTIS LISTED IN ORDER OF INSERTION.	Informative message indicting an expanded listing.	None.	ITEMIZE
CPM - ACCESS LEVEL NOT VALID FOR JOB.	You attempted to change the job access level to a value not valid for the job.	Retry with a valid level.	CPM
CPM - ARGUMENT ERRDR.	An incorrect argument was specified.	Retry the command using valid arguments.	CPM
CPM - INCORRECT PACK NAME.	An incorrect pack name was specified.	Ensure that a valid pack name is used.	CPM
CPM - INCORRECT PAGE VALUE.	There was an error in your job page size information.	Retry with correct values.	CPM
CPM - INCORRECT REQUEST.	A job has an incorrect CPM function or subfunction code, or extended reprieve is set (for EREXIT).	Specify auto recall on monitor request call to CPM.	CPM
CPM - INCORRECT SHELL FILE.	You entered a SHELL command with one of the following errors: <ul style="list-style-type: none"> - The specified shell program was not found on mass storage. - You specified the local load option for a file that was not found in the local file name table (FNT). 	Correct the SHELL command or create the proper shell program file.	CPM
CPM - LIBRARY NOT FOUND = library.	The system could not find the specified library.	Ensure that the library name is correct on the command or macro.	CPM
CPM - MASS STORAGE ERROR.	One of the following occurred: <ul style="list-style-type: none"> - A irrecoverable error occurred when your job attempted to access a mass storage device. - Your job was aborted after being rolled out due to a recoverable mass storage device error. 	Inform site analyst.	CPM
CPM - MISSING SHELL LOAD OPTION.	You attempted to set a shell control without specifying a load option.	Specify a load option in the SHELL command or macro and try again.	CPM
CPM - SYSTEM ERROR.	For function 42 (Special Charge) or 106 (Set Job Characteristics), CPM was unable to read the input file.	Write a PSR and include support materials to allow CDC to duplicate the problem.	CPM
CPM - USER ACCESS NOT VALID.	One of the following occurred: <ul style="list-style-type: none"> - You tried to perform a CPM operation for which you are not authorized. - 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 L display utility was initiated at the system console with the appropriate commands.	CPM
CPU ABORT.	RECLAIM system error.	Check previous messages and the dayfile for the reason for the abort. If you find no other messages, inform the site analyst.	RECLAIM
CPU ERROR EXIT.	Self-explanatory.	None.	RECLAIM
CPU ERROR EXIT AT address.	The errors listed after this message occurred at address, causing job step termination.	Refer to the descriptions of the error messages issued with this message.	1AJ
CRC ERROR.	An error was detected in the cyclic redundancy character. The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Retry or inform site analyst.	1MT
CT, CN, OR OV KEYWORD NOT PRESENT	The COMPACT directive has been specified on the RECLAIM command; COMPACT requires either the CT, CN, or OV option, but none of these was specified.	Retry with the CT, CN, or OV option.	RECLAIM
CUMULATIVE LIMIT EXCEEDED.	Dayfile and output file message indicating that one of the installation-defined resource usage accumulators for this project exceeded the maximum allowed.	None.	VALEX
CUMULATIVE SRU LIMIT EXCEEDED.	Dayfile and output file message indicating that accumulated SRUs have exceeded the maximum allowed.	None.	VALEX
DATA BASE ERROR.	The system has detected an error in its validation file.	Contact installation personnel.	VALEX
DATA/PERMIT ERRORS.	When the specified file was loaded from tape, errors were encountered in both data and permit information.	Enter the CHANGE command or macro with CE parameter to allow access to the file. Examine the file and check if data is	PFM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
DATA TRANSFER ERROR.	An error occurred in a read operation during a file transfer.	accurate. Enter a CATLIST command to see if the permits are accurate.	PFM
DATABASE CORRUPTED.	RECLAIM encountered a problem with the database; usually indicates that the database is empty or has been overwritten with something other than a database.	Check to ensure that your database is in correct RECLAIM format and that it is not empty.	RECLAIM
DATABASE NOT FOUND-DEFINING NEW ONE.	RECLAIM did not find the specified database in your catalog and is attempting to define a new one.	None.	RECLAIM
DAYFILE - INCORRECT JOB TERMINATION OPTION.	An incorrect value was specified for the JT option.	Correct option and retry.	DAYFILE
DAYFILE - INPUT FILE CANNOT BE EXECUTE ONLY.	You specified an execute-only file in the I=infile parameter of the DAYFILE command. The input file cannot be execute-only.	Specify an input file that is not execute-only.	DAYFILE
DAYFILE - INTERACTIVE 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 - 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 - PAGE SIZE FORMAT.	The page size option is nonnumeric.	Retry with a valid option.	DAYFILE
DAYFILE - PRINT DENSITY.	The print density option is not 3,4,6, or 8.	Retry with a valid option.	DAYFILE
DAYFILE - PRINT DENSITY FORMAT.	The print density option is nonnumeric. Print density must be 3,4,6, or 8.	Retry with a valid option.	DAYFILE
DC REQUIRED IF JSH/UJN NOT SPECIFIED.	If the JSH or the UJN parameters are omitted, the DC parameter must be entered.	Add the DC parameter and retry.	QGET DROP
-DEBUG-GARBAGE IN ZZZZZDS FILE.	File ZZZZZDS contains information other than the suspended job of the current session. Retrieval of debug information from session is impossible.	Return file ZZZZZDS and restart debug session.	DEBUG
DEBUG-ILLEGAL PARAMETER-p.	You specified an incorrect parameter on the DEBUG command. The three parameters allowed are ON, OFF, or RESUME.	Reenter DEBUG command with correct parameter.	DEBUG
DEBUG-ONLY ONE PARAMETER ALLOWED.	You entered more than one parameter on the DEBUG command.	Reenter DEBUG command with correct parameter.	DEBUG
-DEBUG-ZZZZZDS 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
DECK LIST.	Informative message indicating an expanded listing.	None.	ITEMIZE
DEFAULT CHARGE NOT VALID.	Dayfile and output file message indicating one of the following: - The default charge or project number associated with the current user name does not exist. - The default charge or project number associated with the current user name is not available to that user name. - The default charge and project numbers associated with the current user name exist but are inactive.	Use another charge and project number.	VALEX
DEFAULT CHARGE NULL.	No default charge and project numbers are defined for this user.	Specify valid charge and project numbers.	VALEX
DEFAULT FAMILY USED.	You entered a FAMILY command with no parameter. The default family was used.	None.	CONTROL
DEL	The system has deleted the last line of input. This occurred when the terminal operator entered the cancel character (CN) followed by the end of line/end of block terminator.	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.	Rerun the job. If the problem reoccurs, inform a site analyst, write a PSR and include support material so that CDC can duplicate the problem.	RESEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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 ENQUIRE, JSN to identify existing detached jobs and DROP, jsn to drop selected jobs.	IAFEX
DEVICE ERROR ON FILE filename AT address.	An irrecoverable error occurred on the mass storage device containing the file filename.	Inform customer engineer.	IMS
DEVICE INACCESSIBLE - DETACH FAILED.	Your job could not be detached due to a mass storage device error.	Inform site analyst.	IAFEX
DEVICE INACCESSIBLE - LOGIN FAILED.	Login failed because of a mass storage device error.	Inform site analyst.	IAFEX
DEVICE INACCESSIBLE - RECOVERY FAILED.	Your job could not be recovered due to a mass storage error.	Inform site analyst.	IAFEX
DEVICE UNAVAILABLE.	Message resulted from one of the following: - A packname was specified for a pack that is not currently mounted. - For a DIS job, no SUI or USER command has been entered. - No permanent file device could be found for your user name. - On a secure system, no permanent file device with the proper access level could be found for your user name.	If a pack name was specified, try again with the WB or NA parameter to request that the pack be mounted. For other errors, inform site operator.	PFM
DIRECT ACCESS DEVICE ERROR.	Message resulted from one of the following: - An incorrect device type was specified. - The device on which the local file resides may not contain direct access files.	Specify correct device type or return the local file and retry.	PFM
DIRECT ACCESS FILE ERROR.	The system sector for the file contains incorrect data or cannot be read.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLDAD on the device.	PFM
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
DISK FULL.	Job was terminated without exit processing.	Resubmit job.	IAJ
DM* TOO SHORT.	System file DM* cannot accommodate the dump.	Inform site analyst.	CPMEM
DROP COMPLETE.	Informative only.	None.	DROP
DROPPED BY USER JOB jsn.	The job was dropped by the job jsn.	None.	DROP
DSP - ALTERNATE FAMILY NOT ALLOWED.	You attempted to route a file or a job to a different family, and you are not validated to do this.	If you need to be able to route to a different family, contact site personnel.	DSP
DSP - CANNOT ROUTE JOB INPUT FILE.	An attempt was made to route the job input file.	Copy the 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 the complete bit before calling DSP.	DSP
DSP - DEFERRED ROUTING NOT ALLOWED.	A file cannot be deferred routed if data declaration or implicit/explicit remote text is specified.	Correct and retry.	DSP
DSP - DEVICE FULL.	There is no space on the device for current use.	Retry ROUTE command or macro at a later time.	DSP
DSP - FILE NAME ERROR.	An attempt was made to create a file with an incorrect 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.	The forms code specification [FC=fc parameter] 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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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 the job for processing.	DSP
DSP - INCORRECT DATA DECLARATION.	The specified data declaration is not recognized.	Correct and retry.	DSP
DSP - INCORRECT DISPOSITION CODE.	The specified disposition code is not recognized.	Correct disposition code and retry.	DSP
DSP - INCORRECT EXTERNAL CHARACTERISTICS.	You specified an undefined external characteristic code.	Correct and retry.	DSP
DSP - INCORRECT FILE MODE.	You attempted to route an execute-only file.	Ensure that the file to be routed is not an execute-only file.	DSP
DSP - INCORRECT FILE TYPE.	You entered a ROUTE command or macro that attempts to route a primary file or a direct access file. These files cannot be routed.	Copy the file to a temporary local file and retry the ROUTE for the local file.	DSP
DSP - INCORRECT INTERNAL CHARACTERISTICS.	You specified an undefined internal characteristics code.	Correct internal characteristics code and retry.	DSP
DSP - INCORRECT LID.	You specified an incorrect logical identifier (LID) for a remote mainframe on the ROUTE command.	Ensure that your LID specification is correct. If the problem persists, inform site analyst.	DSP
DSP - INCORRECT ORIGIN TYPE.	The system cannot queue the file for input with the specified origin type.	Specify valid origin type.	DSP
DSP - INCORRECT REQUEST.	Message resulted from one of the following: - Parameter block is not within field length. - Request was not made with auto recall.	Change parameter block addresses or specify auto recall.	DSP
DSP - INCORRECT SERVICE CLASS.	The two-character service class is not valid for the user or the origin type of the file.	Correct and retry. If you are unable to specify a service class that you ought to be able to use, contact site personnel.	DSP
DSP - INCORRECT TID.	Message resulted from 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 - INCORRECT USER COMMAND.	For an operation on an input file, either a USER command was not included or the user name or password on the USER command is not valid.	Verify that a USER command is present and that the user name and password are correct.	DSP
DSP - JSN ALREADY IN SYSTEM.	You attempted to specify a forced JSN, but the specified JSN was already in use.	Retry, specifying a different JSN.	DSP
DSP - MASS STORAGE ERROR.	A mass storage error was encountered on the file.	Recreate the file and retry.	DSP
DSP - QFT FULL.	The system cannot process your file because it is already processing its maximum number of queued files.	Wait and retry.	DSP
DSP - QUEUED FILE READ ERROR.	The system encountered an equipment malfunction and, consequently, cannot process your deferred routed file.	Recreate the output file.	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
DSP - USER ACCESS NOT VALID.	Caller is not validated to specify one of the following: DLID SLID Data declaration	Contact site personnel if you need to be able to do this.	DSP
DSP - USER SECURITY COUNT EXHAUSTED.	You attempted to route an input file to a user name which has exhausted its security count.	Contact site support personnel to get your security count reset.	DSP
DUMP FILE CONTAINS 63 DUMPS. FILE IS FULL.	In response to a DUMP or COMPACT operation, RECLAIM has determined that the dump file already contains the maximum allowable number of dumps; this directive is ignored.	Compress the current dump file using the COMPACT directive or use a new tape.	RECLAIM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
DUMP FILE MALFUNCTION-EOI ENCOUNTERED.	An EOI was encountered before the specified dump file or record was found; the tape has probably been overwritten.	Check the contents of the dump file tape.	RECLAIM
DUMP FILE MALFUNCTION-FILE NAME MISMATCH.	The specified file name does not match the file name found at the specified position on the dump tape; the tape has probably been overwritten.	Check the contents of the dump tape.	RECLAIM
DUMP FILE MALFUNCTION-POSITION LOST.	RECLAIM system error.	Inform site analyst.	RECLAIM
DUMP FILE MALFUNCTION-UNRECOGNIZABLE PFC.	The PFC for the dumped file is incorrect; either the tape is bad or has been overwritten.	Check the contents of the tape.	RECLAIM
DUMP FILE MUST BE IN WRITE MODE	A DUMP or COMPACT was attempted using a mass storage dump file which was attached in some mode other than WRITE mode.	Attach the dump file in WRITE mode and retry.	RECLAIM
DUMP FILE NOT FOUND	A LOAD or COPY was attempted and the mass storage dump file indicated by the data base could not be found. The file may have been purged or may never have been made permanent.	Check your options. If they are correct, you may have to load your files from an alternate source. Check with your site analyst.	RECLAIM
DUMP NOT ALLOWED OF RESERVED FILE NAME filename.	An attempt was made to dump a local file with the same name as a RECLAIM scratch file, dump file, INPUT, or OUTPUT.	Retry using a different local file name.	RECLAIM
cmd DUPLICATE CHARACTER	The value you specified in the terminal definition command conflicts with another terminal definition.	Try another value.	CCP
DUPLICATE FILE NAME.	The file specified already exists in the system.	Use different name in request.	LFM
DUPLICATE FILE NAME.	A duplicate file name was encountered on a single directive.	Correct directive.	IEDIT
DUPLICATE UJN - MUST SPECIFY JSN.	Two of your jobs have the same UJN.	Use the ENQUIRE command to determine the appropriate JSN to use.	QGET
DUPLICATE UJN - MUST SPECIFY JSN.	More than one job exists with the specified UJN.	Specify a JSN instead of (or in addition to) the UJN.	DRDP
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
EDITING COMPLETE.	Library editing is complete.	None.	LIBEDIT
filename EMPTY.	The file specified on a SAVE, REPLACE or APPEND contains no data.	Verify that file contains data and retry.	PFM
EMPTY CATALOG.	There are no permanent files in your catalog.	None.	PURGALL
EMPTY CATALOG.	If the UN parameter was not specified, no files exist in your catalog. If the UN parameter was specified, either no files exist which belong to that user, or you are not permitted to access any of the files belonging to that user.	None.	CATLIST
EMPTY FILE (SSF).	Internal error.	Submit a PSR with information on how to duplicate the problem.	IEDIT
EMPTY SORT INPUT FILE.	Dayfile message indicating that the file specified on the SORT command contains no data.	Correct and rerun.	SORT
END est, nn.	The operator ended batch equipment est for nn copies.	None.	QAP
END OF SET INCORRECT REQUEST, filename AT address.	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 incorrect.	Return the tape via a RETURN, UNLOAD, or EVICT command, or reposition the tape with the LABEL command.	1MT
END OF TAPE, filename AT address.	The end-of-tape was encountered.	Ensure that the 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.	Write a PSR and include support materials to allow CDC to duplicate the problem.	ENQUIRE

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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
ENTER TEXT MODE.	Informative message indicating you are in text mode.	None.	IAFEX
ENTERING RECOVERY MODE ... WAIT	SCOPE 2 is entering recovery mode. There is a delay until the recovery cycle completes.	Wait for the user prompt.	HELLO7
ENTERING 7000 MODE ... WAIT	The user is entering 7000 mode.	Wait for prompt.	HELLO7
ENTRY POINT NOT FOUND.	The specified entry point could not be found.	Verify that the entry point is valid.	1AJ
ENTRY POINT TABLE OVERFLOW.	The number of entry points per program unit has exceeded the table space allowed.	Contact CYBER Software Support.	LIBRARY
EOF ENCOUNTERED.	End-of-file was encountered before copy termination condition was satisfied.	None.	COPYB
EDI CHANGED BY RECOVERY.	The file was truncated during deadstart recovery.	Use the 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
EDI 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.	COPYB COPYC
EPILOGUE AND SHELL CONFLICT.	A shell program with the no-abort option set was active at the time you attempted to specify charge and project numbers which had an epilogue defined.	Specify charge and project numbers which do not have an epilogue defined.	VALEX
EQest,DNdn,message.	A form of PFM error message (issued to the System dayfile, error log, and sometimes the user dayfile) identifying the mass storage equipment on which the error occurred. est EST ordinal of device dn Device number message PFM error message	Refer to the significance and action of the message as given in this list of error messages.	PFM
EQUIPMENT NOT AVAILABLE.	The requested equipment is either in use or does not exist, or no requested equipment with the proper access level can be found.	Ensure accuracy of macro or command or retry at a later time.	LFM RESEX
ERASE LIMIT, filename AT address.	For 1600 cpi or 6250 cpi densities, the block could not be rewritten properly after forty erasures (approximately 15 feet of tape) or; for 800 cpi, the current block could not be rewritten properly due to parity errors after forty attempts to erase a single area of tape (approximately forty feet of tape). The tape is unusable and the job will terminate.	Clean tape drive. Have the tape cleaned and certified or use a new tape. Run diagnostic T7X to insure proper tape drive operation.	1MT
ERRDR CODE ec, filename AT address.	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 an error while getting a list of the files at the control point. This could occur if there are many local files.	Contact CYBER Software Support.	CHKPT
ERROR FLAG TERMINATION, filename AT address.	The job was aborted with a tape operation in progress. The operation/request is not complete. For example, the operator killed the job while tape error recovery was in progress.	None.	1MT
ERROR FLAG TERMINATION, FILE filename AT address	The job was aborted with an input/output request in progress. The operation/request is not complete.	None.	1MS
ERROR IN ALTERNATE CATLIST PERMISSION.	The option specified on the AC parameter is incorrect.	Check the AC parameter description and correct the command.	PFILES
ERROR IN ARGUMENT.	One of the following occurred: - You specified no arguments or a blank argument. - You specified too many files. - You entered an incorrect parameter or combination of parameters. - You specified both XD and XT parameters in a command. - You specified an XD or XT parameter in a command other than CHANGE, DEFINE, or SAVE.	Reenter the command with correct parameters.	PFILES

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
	- You specified an XD or XT parameter in a DEFINE or SAVE command without specifying a PW parameter.		
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 incorrect or could not be recognized. - The TM option was selected but no data was specified. - Both the device number parameter and the pack name or auxiliary device parameter were selected; auxiliary devices do not have device numbers. - Both the NA and WB parameters were specified; select one or the other.	Correct command and retry.	BLANK ITEMIZE PURGALL
ERROR IN ARGUMENTS.	RESTART detected incorrect arguments on the command call.	Correct the command.	RESTART
ERROR IN ARGUMENTS.	A parameter specified on the Job, USER, or CHARGE command was not correct.	Correct the parameter in error and retry.	VALEX
ERROR IN ATTACHING USER DATABASE.	RECLAIM system error.	Check the dayfile for a more specific error message. If there is no message in the dayfile, inform the site analyst.	RECLAIM
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 CHVAL ARGUMENTS.	An incorrect keyword was specified on the command or too many characters were specified for the UC parameter.	Correct and retry.	MODVAL
ERROR IN CONTROL ARGUMENTS.	Either the number of arguments was incorrect, an incorrect 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 DEFAULT CHARGE.	A default charge number is defined for this user, but no default project number is defined.	Specify valid charge and project numbers.	VALEX
ERROR IN DEVICE NUMBER.	The file residency as specified by the device number parameter was incorrect.	Correct the device number parameter.	PURGALL
ERROR IN DOCUMENT ARGUMENTS.	An error was found among the numbers that were entered as parameters.	Check parameters and retry.	DOCUMENT
ERROR IN EXPIRATION DATE.	One of the following occurred: - You specified an expiration date that is before the current date. - You specified a date with the month not in the range 1-12 or the day not in the range 1-31. - You specified an XT parameter greater than the maximum allowed.	Reenter the command with a correct XD or XT parameter.	PFILES
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 CATEGDRY.	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 the specified file was loaded from tape, an error was encountered in the data.	Enter the CHANGE command or macro with CE parameter to allow access to the file. Examine the file and ensure the accuracy of the data.	PFM
ERROR IN FILE TYPE.	You specified an incorrect file type.	Check the PURGALL command description and correct the statement.	PURGALL
ERROR IN FILENAME LIST.	A syntax error was found in the list of file names entered using the PF=* option.	Correct the file name list and retry.	RECLAIM
ERROR IN KRONREF ARGUMENTS.	The command entered contained an incorrect parameter.	Correct and rerun.	KRONREF
ERROR IN LIBRARY ARGUMENTS.	The LIBRARY command has an incorrect 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:lfm were included on the LIMITS command.	Correct and rerun.	MODVAL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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.	An incorrect parameter or an incorrect number of parameters were entered on the PASSWOR command or in response to an interactive prompt from the PASSWOR command.	Correct the PASSWOR command and resubmit.	MODVAL
ERROR IN PERMIT DATA.	When the specified file was loaded from tape, an error was encountered in the permit entries.	Enter the CHANGE command or PFM macro with CE parameter to allow access to the file. Do a CATLIST to see if the permits are accurate.	PFILES
ERROR IN PREFERRED RESIDENCE.	The option specified on the PR parameter on the SAVE, DEFINE or CHANGE command is incorrect.	Check the PR parameter description and correct the command.	PFILES
ERROR IN RESIDENCE.	The R:r parameter you specified contains an incorrect device type.	Correct the R:r parameter.	PFILES
ERROR IN ROUTE FUNCTION, LFN+filename.	Informative message issued to the system dayfile indicating that an error occurred while routing filename.	Examine the job's dayfile for a more specific error message.	DSP
ERROR IN SECHDR PARAMETERS.	You specified an unknown parameter on a SECHDR command.	Reenter the command with the correct parameters.	SECHDR
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
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 the parameters are in correct format and new values are valid.	TRMDEF
ERROR IN UPROC ARGUMENTS.	Dayfile message indicating that incorrect command arguments were entered.	Specify the correct arguments and rerun.	MODVAL
ERROR IN 5TH COPYL PARAMETER	Fatal error; only A, R, E, and T are recognized parameters.	Correct processing option.	COPYL
ERROR LIMIT. integer ADDITIONAL ERRORS FOUND.	More than 20 errors were found during conversion.	Verify deck and retry.	FCOPY
ERROR LIMIT EXCEEDED.	integer Errors in excess of 20 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.	COPYB
ERROR ON DETACH.	The system was not able to detach the job upon your request.	Retry command.	IAFEX
ERROR PROCESSING LIMIT.	Following an error, a job has tried to restart itself more than 512 times.	None.	1AJ
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.	CPMEM
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
EXPECTING A SELECTION OR A CALL FOR HELP	In response to a line mode prompt by a menu procedure, something other than a menu selection or a question mark [requesting help] was entered.	Make a menu selection or request help by entering a question mark (?).	CCL
EXPECTING ?, \ BETWEEN PARAMETERS	An incorrect separator was used in a line mode interactive procedure call.	Enter parameter values in response to the prompts, or reissue the command in the correct format.	CCL
EXPECTING ?, / \ BETWEEN PARAMETERS	An incorrect separator was used in a line mode interactive procedure call.	Enter parameter values in response to the prompts, or reissue the command in the correct format.	CCL

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
EXPECTING ? OR TERMINATOR AFTER SELECTION	On a line mode menu procedure call, only a period, a right parenthesis, or a question mark (requesting help) may follow the menu selection.	Make the appropriate menu selection in response to the prompt, or reissue the command in the correct format.	CCL
EXPECTING ? OR TERMINATOR AFTER NAME ON CALL	On a line mode menu procedure call, only a right parenthesis, a period, a question mark (requesting help) or a comma followed by a menu selection may follow the procedure/file name.	Make the appropriate menu selection in response to the prompt, or reissue the command in the correct format.	CCL
EXPECTING SELECTION AFTER COMMA ON CALL	On a line mode menu procedure call, a comma followed the procedure/file name, but it was not followed by a valid menu selection.	Make the appropriate menu selection in response to the prompt, or reissue the command with a valid selection.	CCL
EXPECTING ? , \ SEPARATOR ON RESPONSE	An incorrect separator was used in a response to a prompt from a line mode interactive procedure.	Enter parameter values in response to the prompts, or use the appropriate separators as indicated for multiple parameter value responses.	CCL
EXPECTING ? , / \ SEPARATOR ON RESPONSE	An incorrect separator was used in a response to a prompt from a line mode interactive procedure.	Enter parameter values in response to the prompts, or use the appropriate separators as indicated for multiple parameter value responses.	CCL
EXTENDED MEMORY 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.	IAJ
EXTENDED MEMORY FLAG REGISTER PARITY.	Parity error detected on extended memory flag register operation.	Inform customer engineer.	IAJ
EXTENDED MEMORY MINIMUM CM NOT VALIDATED.	You are not validated for the minimum amount of central memory required when using extended memory.	See site personnel concerning validations.	VALEX
EXTENDED MEMORY 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.	IAJ
EXTENDED MEMORY PARITY ERROR.	ECS transfer on a model 176 failed.	Inform customer engineer.	IAJ
EXTENDED MEMORY PARITY ERROR.	Hardware error.	Inform site analyst.	RECLAIM
EXTENDED MEMORY READ ERROR.	An irrecoverable extended memory read error occurred.	None.	CPMEM
FAMILY NOT FOUND.	No family exists with the name you specified.	Retry with an existing family name.	CONTROL
FCOPY COMPLETE.	The FCOPY conversion is finished.	None.	FCOPY
FET TOO SHORT.	FET is not long enough for processing by requested function.	Increase FET size.	LFM
FIELD BEGINNING AT nn NO COMPARE.	Verification of label field at character position nn does not compare with that specified by the user on an OPEN statement.	None.	1MT
FILE BOI/EDI/UI MISMATCH.	Data in the system sector for the file does not match information from the EDI sector and/or catalog information. This indicates that the file has been destroyed.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLDAD on the device.	PFM
FILE COUNT PARAM NOT NUMERIC.	The file count specified with the N command option is not numeric.	Correct the parameter and retry.	ITEMIZE
FILE ERROR filename.	An incorrect 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 IS CONNECTED	The file the user tried to submit is connected.	Check the file name and retry.	HELL07
FILE/JOB NOT FOUND.	The specified file or job was not found in the system.	Use the ENQUIRE command to ensure the job is still in the system.	CONTROL
FILE LENGTH ERROR.	The length of a file does not equal the length specified in the catalog. This indicates that the file has been destroyed.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLDAD on the device.	PFM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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
FILE NAME CONFLICT.	The old file, replacement file, and the new file were not all specified by unique file names.	Specify unique file names.	COPYL COPYLM
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.	COPYB FCOPY
FILE NAME CONFLICT.	The same file name was specified for both the file to be itemized and for the list file.	Specify the correct file name and retry.	ITEMIZE
FILE NAME ERROR.	File name contains incorrect characters or contains more than seven characters.	Ensure that a valid file name is specified.	LFM LISTLB PFM
FILE NAME MISSING.	The L0=P, LP=FP or L0=X option was specified in a CATLIST command, but no file was specified with the FN parameter.	Correct and retry.	CATLIST
FILE NAME TOO LONG.	A file name was specified that has more than seven characters.	Use a file name that has seven or less characters.	COPYL
FILE NOT DISK RESIDENT.	The disk space associated with the specified permanent file has already been dropped.	None.	PFM
FILE NOT FOUND - filename.	The designated input file did not exist prior to the copy or verify.	If the file should have existed, reaccess the file and retry.	COPYB VERIFY
FILE NOT FOUND OR FAILED CRITERIA - filename	RECLAIM did not process the named file because it could not find the file on the dump tape or file (for a LOAD or COPY operation); it could not find a local or permanent file by that name (for a DUMP operation); or it found the file, but failed to meet other criteria. the file filename A file name you specified in a RECLAIM PF or FN option	Check to see that a file exists with the specified file name.	RECLAIM
FILE NOT ON MASS STORAGE.	The specified file does not reside on mass storage.	Copy file to mass storage and retry.	1AJ
FILE STATUS NOT SPECIFIED.	Your SETFS command does not specify a file status.	Retry command with a file status parameter.	MFILES
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, UPDATE, 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 filename AT address.	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.	IMS
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
FIP - ACN acn NOT WITHIN RANGE.	A data transfer error has occurred. The system has halted the file transfer. acn Application connection number (octal)	Inform site analyst.	FIP
FIP - CID ERROR xxB ON filename, reason	NETXFR has terminated file transfer because of a fault while reading or writing the local file. xx CID error code (octal) (Refer to Volume 4 of the NOS Version 2 Reference Set.) filename The affected file reason Abbreviated error description: ERROR ON FILE-OPEN. DEVICE FULL. PARITY ERROR. TAPE BLOCK SIZE ERROR.	Correct problem and retry transfer. Inform site analyst if problem persists.	FIP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
FIP - CONNECTION BROKEN ON ACN acn.	The network connection has been broken unexpectedly. The system has halted the file transfer. acn Application connection number (octal)	Rerun your job. Inform site analyst.	FIP
FIP - CONVERT MODE N/A FOR filename.	The application timed out waiting for resources to become available. filename The affected file	Try again. If problem persists, inform site analyst.	FIP
FIP - DISABLE WARNING RECEIVED.	The system has halted the file transfer because the network is shutting down immediately.	Retry the file transfer after the network is reactivated.	FIP
FIP - GT 4 FILE TRANSFERS INITIATED.	The system is attempting too many file transfers simultaneously. The system has not initiated the file transfer you are requesting.	Inform site analyst.	FIP
FIP - INITIATING XFR OF filename.	The system has initiated the transfer of file filename.	None.	FIP
FIP - NETOFF DURING FILE TRANSFER.	An internal error occurred during your file transfer. The file transfer was not completed successfully.	Inform site analyst.	FIP
FIP - NETXFR ERROR, ALL TRANSFER TABLES IN USE.	NETXFR aborted the application after finding all internal tables in use.	Retry transfer. Inform site analyst if problem persists.	FIP
FIP - NETXFR ERROR, BAD ATTRIB. xxxx FOR COMMAND yy.	NETXFR aborted the application after finding a faulty internal attribute value. xxxx Attribute value (hexadecimal) yy Associated command value (decimal)	Retry transfer. Inform site analyst if problem persists.	FIP
FIP - NETXFR ERROR, BAD INTERNAL TRANSFER STATE = xxB.	NETXFR detected an internal fault and aborted the application. xx Internal transfer state (octal)	Retry transfer. Inform site analyst if problem persists.	FIP
FIP - NETXFR ERROR, BAD PARAMETER DD = xxB.	NETXFR detected a faulty DD parameter in the application's NETXFR call and aborted the application. xx DD parameter value (octal)	Inform site analyst or correct application program.	FIP
FIP - filename ON INVALID DEVICE.	File filename is assigned to an inaccessible device. The system has halted the file transfer. filename File being transferred	Reassign file filename to an accessible device and rerun job. Inform site analyst if the problem persists.	FIP
FIP - OUTPUT BLOCK NOT DEL ON ACN acn.	The remote system did not receive the network message or data block before the time-out period elapsed. The system has halted the file transfer. acn Application connection number (octal)	Retry the file transfer. Inform site analyst if the problem persists.	FIP
FIP - PREMATURE TERMINATION RCVD ON acn.	The system detected an error during a file transfer and halted the file transfer. acn Application connection number (octal)	Retry file transfer. Inform site analyst if the problem persists.	FIP
FIP - PROTOCOL ERROR DETECTED.	An unrecognized or unexpected network message has been received. The file transfer is ended.	Retry file transfer. Inform system analyst if problem persists.	FIP
FIP - RHF I/O ERROR xxB ON filename. reason	NETXFR has terminated file transfer because of a fault while reading or writing file data to or from the network. filename The affected file xx RHF I/O error code (octal) [Refer to the Remote Host Facility Access Method Reference Manual for the descriptions of the error codes.] reason Abbreviated error description: BLOCK NOT 80-BIT MULTIPLE. BLOCK LENGTH MISMATCH. CONTROL WORD FORMAT ERROR. HOST ABN MISCOMPARE. INVALID PARTIAL BLOCK. LAST BLOCK TOO BIG. NETWORK ABN MISCOMPARE. RHF FET PARAMETER ERROR. UNKNOWN BLOCK ERROR.	Retry transfer. Inform site analyst if problem persists.	FIP
FIP - SECOND FILE XFR ON ACN acn.	The system attempted a file transfer on a connection that already has a file transfer in progress. The system halted the second file transfer. acn Application connection number (octal)	Inform site analyst.	FIP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
FIP - SYSTEM ERROR. reason. status.	A system error (indicated by reason) has interrupted NETXFR, which is either attempting or abandoning recovery (indicated by status). Subsequent dayfile messages indicate whether the file transfer completed. reason CPU TIME LIMIT. PP ABORT. OPERATOR DROP. OPERATOR RERUN. EM PARITY ERROR. MASS STORAGE LIMIT. I/O LIMIT. status TERMINAL INTERRUPT. FIP REPRIEVING FIP REPRIEVES EXHAUSTED	Retry the file transfer after correcting the problem, or inform site analyst.	FIP
FIP - TIMED OUT WAITING FOR NETWORK.	The network failed to respond before the time-out period elapsed. The system halted the file transfer.	Ensure that the network and remote system are active and retry the file transfer. Inform site analyst if the problem persists.	FIP
FIP - TRANSFER OF filename COMPLETE.	Self explanatory. filename File being transferred	None.	FIP
FIP - TRANSFER OF filename IN PROGRESS.	Self explanatory. filename File being transferred	None.	FIP
FIP - XFR COMPLETE, NO ERR, IDLEDOWN.	The network is shutting down, but your file transfer completed successfully.	None.	FIP
FIP - XFR TERM WITH ERR, IDLEDOWN.	The network is shutting down and your file transfer ended unsuccessfully.	Retry the file transfer when the network becomes active.	FIP
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 (EXTENDED MEMORY).	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 a larger field length specification.	1AJ
FLE TOO SHORT FOR LOAD.	You attempted to load extended memory data beyond the job's extended memory field length.	None.	1AJ
**** FORMAT ERROR.	Your entry is not recognized as valid system input. The line of input is disregarded.	Correct the entry in error.	MODVAL
FORMAT ERROR.	Message resulted from one of the following: - The GTR command format was incorrect. - An incorrect library type was specified. - A record name longer than seven characters was specified.	Refer to the description of the GTR command.	GTR
FORMAT ERROR IN INPUT.	The charge and/or project numbers read from the input file were not in the correct format.	Enter charge and project numbers in the correct format.	VALEX
FORMAT ERROR ON COMMAND.	An error was detected in the format of the command.	Check the description of the command format.	1AJ
FORMAT REQUIRES UNLABELED TAPE.	The format specified (F) is valid only for unlabeled tapes.	The tape must be assigned as an unlabeled tape.	RESEX
FORWARDING JOB MONITOR	The submitted job is being forwarded.	None.	HELLO7
pfh FOUND. or un FOUND.	The file name specified on the CATLIST (LO=0) command was found in your catalog or the file name or user name specified on the CATLIST (LO=P) command was granted permission.	None.	CATLIST
FOUND SECTION section.	This is the last line of a three-line message: MULTIFILE NOT FOUND, filename AT address. REQUESTED SECTION section. FOUND SECTION section. The system has reached the end of a multifile set.	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 sjsn: message	Another interactive user with the JSN sjsn sent this message using the DIAL command.	If validated to use the access subsystem, you may respond to this message using DIAL.	IAFEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
FROM NDP...message.	The network operator sent this message.	If necessary, you can respond using the MS command.	CS
FUNCTION REJECT, filename AT address.	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 .GT. LWA	You specified a first word address greater than the last word address.	Correct addresses and retry.	DEBUG
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 ZZZZZLD 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 (ZZZZZLD).	Enter the LIBRARY command with no parameters to clear the global library set processing, or regenerate ZZZZZLD 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.	1RI
GO FROM OPERATOR	A GO command was issued to the job by the console operator.	None.	1RI
GTR ERRORS.	There are errors on the GTR command.	Check the description of the GTR command.	GTR
H VALUE INCORRECT.	The H parameter was zero or greater than the buffer length.	Correct parameter and retry.	L072
HARDWARE RERUN THRESHOLD EXCEEDED.	The job has been rerun the maximum number of times due to a hardware error. The maximum is n-1, where the default is n=2 (defined by HRTL in common deck CDMSMSC).	Analyze job dumps, maintenance registers, and the error log or HPA reports to determine the cause of the error.	1AJ
HOST AVAILABLE CONTROL CHARACTER=char ENTER INPUT TO CONNECT TO HOST	Your selected host is available. char The network control character for your terminal	Enter any input to initiate your connection to the host.	CCP
HOST BUSY CONTROL CHARACTER=char ENTER char HD TO SEE HOST STATUS	The host rejected your attempt to connect. char The network control character for your terminal	Check the status of the hosts in your configuration.	CCP
HOST CONNECTED CONTROL CHARACTER= char READY FOR INPUT	You are connected to the host. char The network control character for your terminal	Proceed with your login to the host.	CCP
HOST CONNECTED CONTROL CHARACTER=char TERMINAL DISABLED BY NDP	The network operator has disabled your terminal. No further input is possible. char .The network control character for your terminal	None.	CCP
HOST DISCONNECTED CONTROL CHARACTER=char ENTER INPUT TO CONNECT TO HOST	You have been disconnected from the host. char The network control character for your terminal	None.	CCP
HOST DISCONNECTED CONTROL CHARACTER=char TERMINAL DISABLED BY NDP	The network operator has disabled your terminal. No further input is possible. char The network control character for your terminal	None.	CCP
HOST ID NOT ALLOWED	The user requested the host ID as the ID to log into.	Enter an alternate mainframe LID or type END to terminate the HELLO7 session.	HELLO7
HOST UNAVAILABLE CONTROL CHARACTER=char ENTER char HD TO SEE HOST STATUS	You have attempted a connection to a host that is not up. char The network control character for your terminal	Check the status of the hosts in your configuration.	CCP
HTIME xxxxxxxxxxxxxx.xxx 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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
I/O LIMITS EXCEEDED.	Your validated job I/O limits have been exceeded during the current RECLAIM session.	Log in and retry the operation. If it fails again, check your validated I/O limits for tape mounts to ensure that you have not exceeded your limits.	RECLAIM
I/O ON EXECUTE ONLY FILE.	You attempted to load an overlay from an execute-only file, but your main overlay was not loaded from the same file.	Do not attempt to load overlays from an execute-only file if the main overlay was not loaded from the same file.	IAJ
I/O SEQUENCE ERROR.	Action was requested on a local file that was busy.	Write a PSR.	LFM PFM IAJ
I/O SEQUENCE ERROR ON FILE filename AT address.	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.	IMS
IDLE	The system is waiting for you to enter a command.	Enter a command.	IAFEX
ILLEGAL PW SPECIFICATION - IGNORED.	The value specified with the PW command option is not numeric.	Correct the parameter and retry.	ITEMIZE
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
cmd ILLEGAL VALUE	You specified an invalid value in the terminal definition command cmd.	Choose a valid value (refer to appendix J).	NETVAL
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 non-system origin job.	Ensure accuracy of request.	RESEX
IMPROPER LOGIN, TRY AGAIN.	You have entered an incorrect family name, user name, or password during login.	Check parameters and retry.	IAFEX
IMPROPER VALIDATION.	You entered incorrect or insufficient validation information.	Verify that the USER command precedes rest of job (followed by CHARGE, if required) and is the first command after the Job command.	IAJ
IMPROPER VALIDATION.	A CHARGE command is required to continue.	Enter a CHARGE command.	IAJ
INCORRECT ABORT CODE xx.	A batch job was entered from a card reader or routed from another mainframe which could not execute for unknown reasons.	Contact site personnel and write a PSR.	IAJ
INCORRECT ACCESS LEVEL FOR EQUIPMENT.	You have specified a level outside of the equipment access level limits.	Use access level within required equipment's access level limits or use equipment with access level required.	RESEX
INCORRECT ARGUMENT.	On a SETFAL, SETJAL, SETPFAC, or SETPFAL command, you specified an incorrect argument other than access level or access category.	Reenter the command with correct arguments.	MLSEXEC
INCORRECT ARGUMENT VALUE.	A directive keyword was equated to an incorrect value such as a file name longer than seven characters or an alphabetic character in a numeric string.	Correct the value and retry.	RECLAIM
INCORRECT CHARACTER.	Incorrect character read from 9-track tape. If a 1 is detected in bit 7 of a translated character, the character is incorrect. The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Verify the following: <ul style="list-style-type: none"> - Tape contains coded data rather than binary. If binary, use binary copy operation. - Tape is ASCII rather than EBCDIC. If EBCDIC, specify CV=EB on LABEL command. - Tape is 7-bit rather than 8-bit ASCII. If 8-bit, use FCOPY to convert to 7-bit. <p>If all the above are verified (that is, tape is 7-bit ASCII), notify site analyst.</p>	1MT
INCORRECT CHARACTER NUMBER.	In a copy request, one of the following was detected: <ul style="list-style-type: none"> - 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.	COPYC

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INCORRECT COMBINATION OF QN/SI.	SI was specified and QN was not. If SI is specified, then QN must also be specified.	Specify QN and rerun the job.	LISTLB
xx INCORRECT COMMAND	The characters xx must be a terminal definition mnemonic or interactive status command.	Ensure the accuracy of your entry and retry. Specified a CHARGE command CCP	
INCORRECT COMMAND.	The system could not identify the command.	Check validity of command.	IAJ
INCORRECT COMMAND.	One of the following occurred: - You specified a CHARGE command with an incorrect parameter or without a terminator. - You entered an incorrect command or a command you were not validated to use. - The job was not in a state to process the command at that time.	Use the LIMITS command to check your validation, and if you are validated to use the command, correct the format of the command and reenter it.	IAFEX MODVAL
INCORRECT COPY.	File and/or conversion types do not meet copy requirements.	Correct error and retry.	COPYB
INCORRECT COUNT.	The copy file count is nonnumeric or specified as zero.	Correct command and retry.	COPYC
INCORRECT CPU PRIORITY VALUE.	You specified a CPU priority value which exceeds the maximum value allowed.	Reenter the command with a valid CPU priority.	CONTROL
INCORRECT DC CODE.	The disposition code specified on the command is not valid.	Correct and retry.	LDI
INCORRECT DC PARAMETER - dc.	You specified an incorrect DC=dc parameter.	Correct the DC=dc parameter	DROP OGET
INCORRECT 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.	Examine auxiliary device request and ensure its accuracy.	PFM
INCORRECT END OF JOB OPTION SPECIFIED.	The option specified is not valid for a noninteractive job.	Retry with a different option.	CPM
INCORRECT ENTRY.	You entered something other than GO, RELIST, or a valid job sequence name.	Enter GO, RELIST, or a valid job sequence name.	RECOVER
INCORRECT ENTRY, TRY AGAIN	You entered incorrect characters in response to an SRU limit or time limit.	Enter correct characters.	IAFEX
INCORRECT 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
INCORRECT EQUIPMENT ON FILE filename AT address.	A system error.	Inform site analyst.	IMS
INCORRECT ERROR EXIT ADDRESS.	Error exit address is beyond the job's current field length.	None.	IAJ
INCORRECT EXTERNAL CALL.	RESEX did not recognize an external call.	Inform site analyst.	RESEX
INCORRECT FILE MODE.	You tried to unlock a file that was attached in execute-only mode.	None.	LFM
INCORRECT FILE NAME filename AT address.	The file name does not conform to established rules. The file name may have a maximum of seven alphanumeric characters and must be left-justified with binary zero fill.	Use valid file name.	IMS
INCORRECT FILE STATUS.	Specified file status is incorrect.	Check value.	LFM
INCORRECT FILE STATUS.	You specified a file status in a SETFS command other than AD or NAD.	Retry with a proper file status parameter.	MFILES
INCORRECT FILE TYPE.	The specified file is of a type not allowed in the requested operation. Possible causes include attempts to do the following: - Change a nonlocal file to file type library. - Designate a direct access file as the primary file. - Route the primary file. - Change the access level of a nonlocal file.	Verify that the file type is appropriate.	LFM
pfh INCORRECT FILE TYPE.	The operation requested cannot be performed on the specified file because it is the wrong file type. This message is issued when one of the following occurs: -You attempt to define a direct access file with the same local file name as a file currently assigned to the job that is of a file type other than local. -The local file being saved is a direct access file attached in read/allow modify or read/allow append mode.	Implement the action that applies: - Define the file using a unique name. - Return the file and reattach it in a different mode.	PFM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INCORRECT FILE TYPE FOR SPECIFIED CHARACTER SET.	ASCFL or EBCFL was specified as the character set for a non-tape file or any other character set for a stranger tape file.	Correct the parameter and retry.	FCOPY
INCORRECT I/O REQUEST ON FILE filename AT address.	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. You attempted to write on a local file that resides on a device whose highest access level is below the current access level of your job. The system provides a dump of the FET on file OUTPUT.	Verify CIO function code being used. Lower the access level of your job, if possible, or raise the access level of the file.	IMS
INCORRECT ID CODE.	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
INCORRECT 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 job cannot be packed.	Verify that the file being packed is not assigned to an interactive terminal.	PACK
INCORRECT INPUT FILE SYSTEM SECTOR.	Job terminated without exit processing due to system error.	Contact your site support or CYBER Software Support.	1AJ
INCORRECT INSTRUCTION.	The CPU attempted to execute an incorrect or unavailable instruction.	Analyze job output and dumps to determine the cause of the error.	1AJ
INCORRECT JOB ORIGIN TYPE FOR TRMDEF.	The TRMDEF command was entered from a job that was not of interactive origin.	Retry from an interactive origin job or remove the TRMDEF command.	TRMDEF
INCORRECT JSN ARGUMENT.	The JSN specified is not four characters.	Fix JSN and retry.	CLASS
INCORRECT LABEL TYPE, filename AT address.	Incorrect label type. The only valid label types are ANSI labeled and nonstandard labeled.	Use correct label type.	1MT
INCORRECT LID.	A batch job was entered from a card reader or routed from another mainframe with an ST parameter on the job command, but the value specified for the ST parameter was not correct.	Correct ST parameter and retry.	1AJ
INCORRECT 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: (0x+Nx).GT.H (Ix+Nx).GT.(buffer length) where 1 .LE. x .LE. b	Correct parameter and retry.	L072
INCORRECT LIST OPTION.	The list option specified in a CATLIST command is incorrect.	Refer to description of the CATLIST command for valid list options.	CATLIST
INCORRECT LOAD ADDRESS.	The load address is less than 2.	Specify larger load address and retry.	1AJ
INCORRECT LOCAL FILE NAME - lfn.	The FN=lfn parameter on the QGET command contains an error. lfn The local file name for the queued output	Correct the parameter.	QGET
INCORRECT MODIFICATION OF filename AT address.	Either you have attempted to shorten a modify-only file or the file cannot be modified at all.	Determine whether the file can be modified.	IMS
INCORRECT NOISE SIZE ON filename.	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 COPYB, 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 S1 format conversions).	COPYB
INCORRECT -NR- PARAMETER.	You specified something other than NR to inhibit the rewinding of the file.	Correct the NOTE command and retry.	NOTE
INCORRECT OP PARAMETER - op.	You specified an incorrect OP=op parameter.	Correct the OP=op parameter.	QGET DROP
INCORRECT */* OPTION.	An incorrect command option followed the start.	Correct the option and retry.	MFILES
INCORRECT OPTION ARGUMENT.	Command contains an incorrect argument for an option.	Check command description and retry.	CLASS
INCORRECT OPTION X.	The option specified is not defined for ENQUIRE.	Check parameters on command and retry.	ENQUIRE
INCORRECT OUTPUT FILENAME.	Command contained a file name that was more than seven characters long or contained a non-alphanumeric character.	Correct file name and retry.	CLASS

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INCORRECT PARAMETER.	Message resulted from one of the following: - Parameter value was out of range. - You specified a parameter that cannot be included on the command. - Command was incorrect.	Ensure accuracy of command.	CATLIST DROP IAFEX LISTLB QGET
INCORRECT PARAMETER.	You specified an incorrect parameter on the current command.	Ensure the accuracy of your parameter.	CONTROL GTR
INCORRECT PARAMETER.	The S or L parameter was entered as zero on the L072 command.	Retry command, adding a correct S or L parameter value.	L072
INCORRECT PARAMETER LENGTH OR SEPARATOR.	You specified a parameter longer than seven characters or used an incorrect separator.	Ensure the accuracy of your parameter specifications.	CONTROL
INCORRECT PASSWORD.	One of the following PASSWOR command errors: - The new password entered is greater than seven characters or contains an incorrect character. - The incorrect old password was specified.	Correct error and retry.	MODVAL PFILES
**** INCORRECT PASSWORD.	One of the following PASSWOR command errors has occurred: - The old password entered did not match the actual old password. - The new password entered was either too long, too short or contained incorrect characters.	Correct error and retry.	MODVAL
INCORRECT QUEUE SPECIFIED.	You specified an incorrect disposition code.	Retry with valid disposition code.	SUBMIT
INCORRECT QUEUE TYPE.	System error.	Write a PSR and include support materials to allow CDC to duplicate the problem.	ENQUIRE
INCORRECT -R/NR- PARAMETER.	You entered a value other than R or NR for the second parameter of the BLOCK command.	Correct the R/NR parameter and resubmit.	NOTE
INCORRECT REPLY LIMIT REACHED.	The number of consecutive incorrect entries exceeds the limit.	None.	RECOVER
INCORRECT REQUEST.	No parameters were specified on a DMPECS or DMDECS command.	Retry job with corrected command.	CPMEM
INCORRECT 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
INCORRECT RESOURCE COUNT.	Total resource demand exceeds maximum allowed, as defined by installation parameter MAXD (defined in RESEX).	Reduce resource demands.	RESEX
INCORRECT SERVICE CLASS.	On the CHVAL command, the two character service class was not valid for the user or not valid for the origin type.	Correct and retry.	MODVAL
INCORRECT SERVICE CLASS.	You entered an invalid service class for the SC parameter of the CLASS command, or the UC parameter of the CHVAL command. Either you are not validated to use the service class you specified or the service class is not valid for the origin type of the job you specified in the JSN parameter.	Correct the service class entry and resubmit.	CLASS
INCORRECT SERVICE CLASS.	A batch job was entered from a card reader or routed from another mainframe with an incorrect service class specified on the job command.	Correct and retry.	1AJ
INCORRECT SORT PARAMETER.	An incorrect parameter has been specified on the SORT command.	Consult description of SORT command for valid parameters.	SORT
INCORRECT SYNTAX.	A required parameter is missing, parameters are given in the wrong order, or an incorrect separator has been encountered.	Check proper command format and reenter the command.	IEDIT
INCORRECT TAPE DENSITY.	An incorrect NDS tape density has been requested for the dump option.	Correct the tape specification and reenter.	RECLAIM
INCORRECT TAPE FORMAT.	An incorrect NDS tape format has been requested for the dump option.	Correct the tape specification and reenter.	RECLAIM
INCORRECT TERMINAL.	You have exhausted your login retry count.	Check your family name, user name, and password. If the problem persists, contact your site administrator.	IAFEX
cmd INCORRECT TERMINAL CLASS	The value you specified in the terminal definition command cmd is not valid for your terminal.	Choose an appropriate value (refer to appendix J).	CCP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INCORRECT TERMINAL REQUEST.	Informative message indicating that an unidentified request was encountered or auto recall was not requested by the calling job.	Correct erroneous request or rewrite program to use recall.	TLX
INCORRECT TERMINAL TYPE FOR TRMDEF.	The TRMDEF command was entered from a terminal that was not a network terminal.	Retry from a network terminal.	TRMDEF
INCORRECT 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.	NVF
INCORRECT USER COMMAND.	The user name or password could not be validated, or a secondary USER command to a different family was attempted and you are not validated to change families.	Verify that the user name and password are valid.	CPM
cmd INCORRECT VALUE	You specified an invalid value in the terminal definition command cmd.	Choose a correct value (refer to appendix J).	CCP
INCORRECT VALUE FOR - xx.	An incorrect value was specified for attribute xx.	Correct and retry.	TRMDEF
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
INCORRECTLY FORMATTED TRMDEF 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
INDEX TOO LARGE ON OPEN, FILE filename AT address.	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.	IMS
INITIAL COMMAND LIMIT.	The number of commands processed by the job exceeded the limit for which you are validated. At this point, an additional eight commands are allowed for error processing.	Split the job into two or more jobs, reduce the number of commands in the job, or request a larger command limit from site personnel.	1AJ
INITIAL MESSAGE LIMIT.	Monitor has detected that the message limit for the job has expired.	Decrease job size or increase limit.	1AJ
INPUT DISCARDED ..	The network has discarded the last input from your terminal. May be caused by entering a user break or interactive status command while awaiting a response to a previous user break or interactive status command.	Reenter your last input.	CCP
INPUT ERROR LIMIT.	You have entered incorrect charge and project numbers several times.	Enter another CHARGE command specifying valid charge and project numbers.	VALEX
INPUT FILE ERROR.	An error on the input file was encountered during the unpack operation.	Correct and retry.	L072
INPUT FILE IN NO RERUN STATUS.	Informative message.	None.	QFM
INPUT FILE IN RERUN STATUS.	Informative message.	None.	QFM
INPUT FILE NOT FOUND.	The input file must be either a local or an attached direct access file.	Create the file and retry.	FCOPY
INPUT LOST, REENTER LAST LINE.	Your last line of input was lost due to an IAF internal error.	Reenter the last line. If the error recurs, inform site analyst or customer engineer.	IAFEX
INPUT NOT ASSIGNED TO TERMINAL.	You entered the PASSWOR command from a non-terminal input file when password randomization is required.	Resubmit the PASSWOR command in response to the system prompt.	MODVAL
INSERTED LINES OVERLAP EXISTING LINES, AT nnnnn. 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 nnnnn.	Type Y to complete the edit command or type N to terminate the edit command.	IEDIT
INSUFFICIENT RESOURCES ON SYSTEM.	Resource demand exceeds the number of units physically available on the system.	Reduce resource demand.	RESEX
INSUFFICIENT RESOURCES ... RE-ENTER	The system has insufficient resources to handle the current request.	Retry the request.	HELLO7
INSUFFICIENT STORAGE FOR LIBRARY GENERATION.	Additional memory is required for the LIBGEN command.	Increase field length and retry.	LIBGEN
INTERACTIVE JOB STEP, DUMP DISABLED.	You entered a memory dump command directly from a terminal. This is not allowed.	To get a memory dump from an interactive job, place the dump command in a procedure or in an ENTER command.	1AJ
pfn INTERLOCKED.	The file is currently not available for WRITE, MODIFY, UPDATE, or APPEND modes of access because a system utility has attached the file in utility mode.	Retry operation later.	PFM
INTERRUPTED	Informative message indicating a user break one has been detected.	None.	IAFEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
INVALID APPLICATION, TRY AGAIN.	The requested application is unknown or unavailable to the user.	Check the accuracy of the entry. If problems persist, contact installation personnel concerning validations.	NVF
INVALID COMMAND	The user has issued an invalid command to HELLO7.	Check the command and reenter.	HELL07
INVALID DISPLAY DATA FROM SCOPE 2	A hardware or system error occurred.	Reenter the command. If the same message is returned, inform the site analyst.	QUEUE7
pfm IS DIRECT ACCESS.	An indirect access file operation was attempted on a direct access file.	Use the appropriate direct access file request.	PFM
filename 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
pfm 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
ITF, ACN acn, APPLICATION LINKAGE ERROR WITH HOST pid. CN acn, REASON CODE = nn - xxxx.	The host physical identifier pid has detected one of the following anomalies on the application connection number acn: nn xxxx 01 UNRECOGNIZED COMMAND. 02 INVALID QUALIFIER. 03 INVALID ATTRIBUTE. 04 REQUIRED ATTRIBUTE MISSING. 05 CONNECTION TIMED OUT. 06 FC/BRK RECEIVED. 07 FC/NAK RETRY COUNT EXCEEDED. 08 BLOCKS OUT OF SEQUENCE. 09 UNEXPECTED COMMAND.	Inform site analyst.	ITF
ITF, ACN acn, APPLICATION LINKAGE REJECTED BY HOST pid.	Message indicates ITFS application on remote host physical identifier has rejected linkage request. acn Application connection number pid Physical identifier	If problem persists, inform site analyst.	ITF
ITF, ACN acn, RETRANSMITTING TO HOST pid.	The RHF subsystem has repeatedly rejected the transmission of a network block by ITF. acn Application connection number pid Physical identifier	If problem persists, inform site analyst.	ITF
ITF, ACN acn, TRANSMISSION RESUMED TO HOST pid.	ITF has successfully transmitted a network block which had been previously rejected by the RHF subsystem. acn Application connection number pid Physical identifier	None.	ITF
ITF, CONNECTED TO HOST pid ON ACN c/TCN yy.	Message indicates state of terminal connection. pid is the remote host, c is the RHF connection number, and yy is the virtual connection number.	None.	ITF
ITF, COULD NOT CONNECT TO HOST pid. ITF, REASON CODE = nn - xxxx.	ITF was not able to establish an RHF connection to remote host pid due to one of the following conditions: nn xxxx 01 LID UNKNOWN TO SUBSYSTEM. 02 REJECTED BY HOST. 03 NETWORK RESOURCE LIMIT. 04 SERVICER UNAVAILABLE. 05 PATH DISABLED OR DOWN. 06 HOST NOT RESPONDING. other CONNECTION REJECT.	If problem persists, contact your local operator to ensure RHF subsystem and the LCN are operational, and/or operator for remote host pid to ensure RHF subsystem and the ITF servicer application are operational.	ITF
ITF, COULD NOT CONNECT TO RHF SUBSYSTEM. ITF, REASON CODE = nn - xxxx.	ITF was not able to establish communications with the RHF subsystem due to one of the following conditions: nn xxxx 01 SUBSYSTEM UNAVAILABLE. 02 ITF UNAVAILABLE. 03 ITF DISABLED. other NETON FAILURE.	Contact local operator.	ITF
ITF, HOST pid HAS RESPONDED.	Indicates state of connection to host pid.	None.	ITF
ITF, NAM NETWORK IDLEDOWN IN PROGRESS.	The operator has initiated subsystem idledown. ITF will not accept any new connections.	None.	ITF
ITF, RHF NETWORK IDLEDOWN IN PROGRESS.	The operator has initiated subsystem idledown. ITF will not accept any new connections.	None.	ITF

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ITF, RHF NETWORK SHUTDOWN.	The subsystem is terminating immediately. All connections are aborted.	None.	ITF
ITF, Tnnnn, TERMINAL CONNECTION ABORTED BY HOST pid.	Message indicates state of terminal connection.	None.	ITF
ITF, Tnnnn, TERMINAL CONNECTION ABORTED TO HOST pid.	Message indicates state of terminal connection.	None.	ITF
ITF, Tnnnn, TERMINAL CONNECTION ENDED BY HOST pid.	Message indicates state of terminal connection.	None.	ITF
ITF, Tnnnn, TERMINAL CONNECTION ENDED TO HOST pid.	Message indicates state of terminal connection.	None.	ITF
ITF, Tnnnn, TERMINAL CONNECTION REJECTED BY HOST pid.	Message indicates state of terminal connection.	None.	ITF
ITF, TERMINAL TIMEOUT.	Terminal inactivity forced disconnection.	None.	ITF
ITF, TERMINAL WILL DISCONNECT IN ONE MINUTE.	Due to lack of activity, your terminal will be disconnected.	Resume activity or else terminal will be disconnected.	ITF
ITF, WAITING FOR HOST pid TO RESPOND.	Message indicates state of connection to host pid. pid Remote host identifier.	None.	ITF
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
JOB ALREADY WAITING ON SERVICE CLASS.	Change cannot be made because the job is already waiting for a CLASS command in the job to complete.	Wait and retry later.	CLASS
JOB CANNOT ACCESS FILE.	The permanent file being processed has either an access level or access category that is not valid for your job.	None. Your job cannot access the file.	PFM
JOB COMMAND ERROR.(jobname)	The Job command of a routed job contains an error. jobname First 20 characters of the Job command in error.	Correct the Job command and retry.	DSP
JOB COMMAND 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 DROP IN PROCESS	You have issued a DROP,J request which is being processed.	Wait for the user prompt.	HELLO7
JOB HUNG IN AUTO RECALL.	System error.	Inform site analyst.	RECLAIM
JOB HUNG IN AUTORECALL	System detected an autorecall call to an incomplete word and there was no activity to make the word complete.	Correct the program to issue a correct recall request.	1AJ
JOB IN NORERUN STATE ON RECOVERY. or JOB IN NO RERUN STATE ON RECOVERY.	Identifies a job recovered on level 0 or level 3 deadstart that was aborted because it was in a norerun 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. The JSN specified is not associated with a job.	Retry command. Recheck the JSN.	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. Recheck the JSN.	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 NOT SYSTEM ORIGIN.	The job must be of system origin type.	None.	CPMEM
JOB NOT VALID ON REMOTE MAINFRAME.	A batch job was entered from a card reader or routed from another mainframe with an ST parameter on the job command, but the job could not run on the specified remote mainframe.	Contact site personnel concerning your validations.	1AJ
JOB NOT VALIDATED TO USE LINK.	A batch job was entered from a card reader or routed from another mainframe with an ST parameter on the job command, but the job could not run on the specified remote mainframe because the specified user name was not validated to transfer queue files.	Contact site personnel concerning your validations.	1AJ
JOB REPRIEVED.	The job has been successfully reprieved.	None.	1AJ
JOB RERUN.	The job has been terminated and requeued for input.	None.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
JOB STEP ABORT.	Job step aborted due to a system problem such as a rollout file was corrupted due to a mass storage failure.	Retry the job step. If the problem recurs, inform site analyst.	1AJ
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
JOB TERMINATED.	The job had not yet completed login processing when you entered "DROP,, ALL" from another terminal session.	None.	IAFEX
JOBNAME IS jsn	An LDI command queued a file for input. The entered job has the job sequence name jsn.	None.	LDI
JSN MORE THAN FOUR CHARACTERS.	You specified a job sequence name (JSN) of more than four characters.	Correct the JSN=jsn parameter.	DROP QGET
JSN NOT FOUND.	For the JSN parameter of the CLASS command, you specified a JSN that is not in the system or that does not belong to you.	Check to see if you entered the correct JSN.	CLASS
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
LABEL CONTENT ERROR, filename AT address.	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 and retry.	1MT
LABEL MISSING, filename AT address.	During a read operation, a required label was missing.	Ensure that the tape has a label.	1MT
LABEL NOT EXPIRED.	You attempted to write on a tape with an unexpired label.	If current contents of the tape can be sacrificed, have operator blank label tape. Otherwise, wait until label has expired.	1MT
LABEL PARAMETER ERROR ON OPEN, filename AT address.	Label fields did not match on the 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
LDD - ARGUMENT ERROR - address.	One of the following conditions is true in the specified parameter block address: - FET address is less than 2. - FET address is greater than FL-2. - Auto recall was not set when LDD was called.	Write a PSR and include support materials to allow CDC to duplicate the problem.	FDL
LDD - DEVICE ERROR - filename AT address.	An unrecovered device error occurred on file filename.	Contact a customer engineer.	FDL
LDD - I/O SEQUENCE ERROR - filename AT address.	I/O was attempted on a file that is currently active on another I/O request.	Check program for concurrent I/O.	FDL
LDI ID AND FM/UN CONFLICT.	Both the ID and FM/UN parameters were specified on the command. ID may not be specified with FM or UN.	Correct and retry.	LDI
LDQ - ARGUMENT ERROR - address.	The FET address is less than 2 or greater than FL-4.	Write a PSR and include support materials to allow CDC to duplicate the problem.	FDL
LDQ - DEVICE ERROR - filename AT address.	An unrecovered device error occurred on file filename.	Contact a customer engineer.	FDL
LDQ - I/O SEQUENCE ERROR - filename AT address.	I/O was attempted on a file that is currently active on another I/O request.	Check program for concurrent I/O.	FDL
LDR-ENTRY POINT NOT FOUND.	The specified entry point could not be located by 1AJ.	Check validity of call.	1AJ
LDR ERROR.	LDR was unable to load the requested overlay. This message is followed in the job's dayfile by a more detailed message.	See the significance and action for the message which follows this message in the job's dayfile.	1AJ
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 INCORRECT REQUEST.	One of the following occurred: - The function detected was not recognized as a valid LFM function. - An LFM function was issued without the auto recall bit set.	Verify that a valid LFM request is being used.	LFM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
LIBEDIT ARGUMENT ERROR(S).	The LIBEDIT command contains an incorrect parameter.	Check the LIBEDIT command description for the correct format and retry.	LIBEDIT
LIBGEN ARGUMENT ERROR.	An incorrect parameter was used on the LIBGEN command.	Check the format of the LIBGEN command and retry.	LIBGEN
LIBGEN FILE NAME CONFLICT.	The LIBGEN command named the same file as the input file and output file.	Change the input file or output file name.	LIBGEN
LIBRARY FILE NOT GENERATED.	The file to be processed did not contain any of the proper record types.	Check for proper source file.	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 the generation of a library is completed.	None.	LIBGEN
LIBRARY GENERATION FILE EMPTY.	The file to be processed is empty.	Verify that the 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
LID - xxx DOES NOT EXIST IN THE LID TABLE.	In a LISTLID command, you specified a LID that either does not exist or that the site had defined as not listable through LISTLID.	Check to see that you entered the correct LID.	LISTLID
LID=xxx OR ST=xxx IS REQUIRED.	The LID parameter or the ST parameter was specified with no value.	Retry, specifying a value for LID or ST.	LISTLID
LID TABLE ENTRY.	The system LID table did not contain any entries.	Contact site personnel.	LISTLID
LINE LIMIT EXCEEDED.	Your job generated more lines of print than that for which you are authorized. If you do not fill all pages, the system still assesses a minimum number of lines per page.	Ensure that your print files do not contain undesired line feeds and page ejects (refer to appendix H). If necessary, request a higher limit from your site.	QAP
LINE NUMBER INCREMENT ERROR.	Line number increment specified is less than one or greater than 4095.	Reenter command with correct line number increment.	IEDIT
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.	IEDIT
LINE NUMBER OVERLAP ERROR.	Two or more line ranges have at least one line number in common.	Reenter command with non-overlapping line ranges.	IEDIT
LINE RANGE ERROR.	An incorrect line number was specified in a line range.	Reenter command with correct line range.	IEDIT
LINE TERMINATORS NOT USED ON A STRANGER TAPE.	A line terminator parameter was specified for a file residing on a stranger tape.	Remove the line terminator parameter and retry.	FCOPY
LINE lineno TRUNCATED.	The indicated line of the file was truncated during RESEQ processing.	Check file to determine whether relevant data was lost.	RESEQ
LINE WAS DISCONNECTED WHILE LOANED.	The line connection to your secondary application was broken.	Enter another APPSW command to reestablish the connection.	TLX
nnnnnnnnnn LINE(S) TRUNCATED.	nnnnnnnnnn lines were truncated because they were longer than 150 characters.	Determine if relevant data was lost in the truncation. If possible, split the long lines and repeat the copy operation.	COPYC
nnn LINES TRUNCATED.	Your file contained nnn lines which were 140 characters or longer. These lines were truncated by SECHDR.	None.	SECHDR
xxxxxx LINES TRUNCATED.	1 or more lines (xxxxxx) were truncated when they were found to be longer than the fixed length line length (FL parameter) on an FCOPY to a stranger tape.	If data lost was significant, retry command with a larger value for the FL parameter.	FCOPY
nnnnn LINES TRUNCATED ON FILE filename.	nnnnn lines on file filename have been truncated.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLOAD on the device.	IEDIT
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 CYBER Software Support.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
LIST OPTION ERROR.	An incorrect list option was specified for the *LD* parameter in the LIBEDIT command.	Change the *LD* parameter to specify only valid options.	LIBEDIT
LISTLB ABORT.	A fatal error 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
LISTLID COMMAND ERROR.	An incorrect argument was specified.	Correct and retry.	LISTLID
LITERAL STRING IS NOT ALLOWED FOR THIS PARAMETER.	Self-explanatory.	Correct parameter and retry.	TRMDEF
LOADER MISSING.	IAJ could not find BEGIN or LDR*.	Contact your site support or CYBER Software Support.	IAJ
LOCAL FILE xxxxxxxx ALREADY EXISTS.	The file could not be attached because a local file already exists with the name xxxxxxxx.	Use a different name in the request or return the local file.	OGET
LOCAL FILE LIMIT.	The job tried to create too many local files.	Reduce local file usage.	IAJ
LOCAL FILE LIMIT.	Your ATTACH, GET, or DLD request cannot be processed, because you already have the maximum number of local files.	Return some local files and retry.	PFM
LOG - PPU BUSY.	DSD cannot assign a PP to process an entry at this time.	Erase command and retry. Inform site analyst if the problem persists.	DSD
LOST 7000 COMMUNICATIONS	Communications between NOS and the SCOPE 2 system have been temporarily lost.	Enter a RESUME directive to attempt recovery.	HELL07
LO72 COMPLETE.	Informative message indicating that the program has completed processing.	None.	LO72
LRC ERROR.	The longitudinal redundancy check character was read incorrectly (8-track NRZI). The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Retry or inform site analyst.	1MT
M.T. NOT AVAILABLE ON FILE filename AT address.	The magnetic tape executive is not executing.	Inform site operator. Bring up MAGNET.	1MS
MAGNETIC TAPE SUBSYSTEM NOT ACTIVE.	One of the following occurred during a magnetic tape request: - There was no UDT address in the FST. - The UDT address in the FST was incorrect. - The MAG subsystem was not present.	Contact site operator to have MAGNET brought up. Otherwise, contact CYBER Software Support.	LFM RESEX
MAINFRAME:	Prompt requesting the LID of the SCOPE 2 mainframe you want to access.	Enter a valid SCOPE 2 mainframe LID.	HELL07
MAINFRAME ID NOT IN ID TABLE	The mainframe you requested is not available.	Enter an alternate mainframe LID or type END to terminate the HELL07 session.	HELL07
MAINFRAME UNAVAILABLE	The mainframe you specified is not currently available.	Enter a QUEUE7, ID command to list PIDs currently active. Then reenter the command with an active mainframe PID.	QUEUE7
MARGINALLY WRITTEN TAPE, filename AT address.	Some or all of the previous good blocks used to create the recovery block ID window cannot be read with error correction enabled during repositioning of the tape for write recovery. The tape is marginal and will possibly fail on a read. The job will terminate to prevent the possibility of bad data in the library.	Clean the tape drive. Have the tape cleaned and certified or use a new tape.	1MT
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.	An error was encountered in reading a portion of the permanent file catalog or permit information (error log and dayfile message). This indicates a hardware problem with a disk pack or disk drive.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLDAD on the device.	PFM
MASS STORAGE ERROR.	Job was terminated without exit processing.	Resubmit job.	IAJ
MASS STORAGE LIMIT.	You have exceeded your mass storage validation limits.	Return all unneeded files and try again. If the message occurs again, your validation limit is too low to allow the operation you are attempting.	RECLAIM
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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MEMORY OVERFLOW.	Insufficient field length has been specified for KRONREF to execute.	Increase field length with RFL command and retry.	KRONREF
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 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 EXTENDED MEMORY MINIMUM CM FL.	To use extended memory, you must have a required minimum amount of central memory field length. This message indicates that you do not have the required central memory field length.	Increase central memory field length.	1MA
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
MFLINK - APPLICATION CONNECTION BROKEN.	The connection with the remote host was broken by the network or remote host.	Inform site analyst. MFLINK retries unless EP specified.	MFLINK
MFLINK - APPLICATION CONNECTION TIMEOUT.	The remote host did not respond in the allotted time.	Inform site analyst. MFLINK retries unless EP specified.	MFLINK
MFLINK - APPLICATION DISABLED.	The operator has disabled PTF for the network subsystem (RHF or NAM).	Contact site operator. MFLINK retries unless RT specified.	MFLINK
MFLINK - BLOCK TOO LARGE.	The remote host or the network sent a block or message that was too large.	Inform site analyst.	MFLINK
MFLINK - CANNOT READ FROM filename.	The file filename does not have read permission. The file type is not supported by MFLINK or the file does not exist.	Correct and resubmit.	MFLINK
MFLINK - CANNOT WRITE ON filename.	The file filename does not have write permission or the file type is not supported by MFLINK.	Correct and resubmit.	MFLINK
MFLINK - CONNECTING TO lid.	Informative message. You are being connected to the remote host you specified.	None.	MFLINK
MFLINK - CONNECTION REJECTED BY REMOTE HOST.	The remote host you specified has rejected the connection.	Inform remote analyst.	MFLINK
MFLINK - CONTINUATION BLOCK DID NOT FOLLOW.	The continuation block did not follow.	Inform site analyst.	MFLINK
MFLINK - ERR/LGL RECEIVED FROM SUBSYSTEM.	The network subsystem (RHF or NAM) detected a logic error in communication.	Inform site analyst.	MFLINK
MFLINK - FC/BRK RECEIVED RC=rc.	The remote host has sent a break with reason code rc.	Inform site analyst.	MFLINK
MFLINK - FC/NAK RETRY LIMIT.	MFLINK was unable to transmit a block after a system-defined number of attempts. Each attempt was rejected by the network subsystem (RHF or NAM).	Inform site analyst.	MFLINK
MFLINK - FILE RETRANSMIT REQUESTED.	The remote host has requested that the host retransmit the file.	None.	MFLINK
MFLINK - INCORRECT LID.	No mainframe in the network has the specified logical identifier (LID), or the LID specified contains an illegal character or is not three characters long.	Use the LISTLID command to select the proper LID.	MFLINK
MFLINK - INVALID ACCESS VALIDATION.	Your user name does not have the required validation to access the remote mainframe.	Contact your site administrator to get the authorization.	MFLINK
MFLINK - INVALID COMMAND cmd.	The system received an invalid command [cmd] from the remote host or received a command out of sequence.	Inform site analyst.	MFLINK
MFLINK - INVALID CONTROL STATEMENT.	You specified an incorrect parameter or value on the MFLINK command.	Correct the MFLINK command and retry.	MFLINK
MFLINK - INVALID DATA DECLARATION.	You specified an incorrect DD=dd parameter on the MFLINK command.	Correct the MFLINK command and retry.	MFLINK
MFLINK - INVALID DIRECTIVE FILE NAME.	You specified an incorrect I=dirfile parameter on the MFLINK command.	Correct the MFLINK command and retry.	MFLINK
MFLINK - INVALID FILE NAME.	The file name you specified on the MFLINK command is incorrect.	Correct the MFLINK command and retry.	MFLINK
MFLINK - INVALID FILE TYPE.	The file you attempted to transfer to a remote host does not have a local file type.	Ensure that the file has the correct type and retry.	MFLINK
MFLINK - INVALID SUPERVISORY MESSAGE.	The remote host or the network sent an incorrect or unsupported supervisory message.	Inform site analyst.	MFLINK
MFLINK - LID CURRENTLY UNAVAILABLE.	The network subsystem (RHF or NAM) is unable to complete the connection to the remote host.	Contact site operator. MFLINK retries unless RT specified.	MFLINK

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MFLINK - LID DISABLED.	The mainframe with the specified logical identifier (LID) has been disabled by the system.	Contact site operator. MFLINK retries unless RT specified.	MFLINK
MFLINK - LID NO LONGER AVAILABLE.	The status of the LID has changed. The lid is no longer defined.	Contact site operator to define the LID and try again. MFLINK retries unless RT specified.	MFLINK
MFLINK - LID UNKNOWN TO SUBSYSTEM.	The network subsystem (RHF or NAM) has rejected a connection request to the specified LID even though the system LID table indicates a network path is available.	If the problem persists, inform site analyst.	MFLINK
MFLINK - LOCAL NETWORK RESOURCE LIMIT.	The network subsystem (RHF or NAM) is temporarily unable to complete the connection to the requested remote host.	If the problem persists, inform site analyst. MFLINK retries unless RT specified.	MFLINK
MFLINK - NETON REJECT = rc.	A system or network error occurred. rc Reason code	Inform site analyst if problem persists.	MFLINK
MFLINK - NETWORK SEQUENCE ERROR.	A network message block was found for the connection before the connection was completed.	Inform site analyst.	MFLINK
MFLINK - NETWORK SHUTDOWN.	The operator has disabled the PTF application in the network subsystem (RHF or NAM). MFLINK terminates the network connection.	Contact site operator.	MFLINK
MFLINK - NETXFR ERROR rc - rejmess	An error occurred during file transfer. rc Reason code rejmess Reject message rc rejmess	Inform site analyst if problem persists. MFLINK retries unless EP specified.	MFLINK
	03 CONNECTION BROKEN. 04 PROTOCOL ERROR. 05 TIMEOUT EXPIRED. 06 BLOCK NOT SENT. 07 TOO MANY TRANSFERS. 08 ACN OUT-OF-RANGE. 09 CID ERROR. 13 IDLEDDOWN ERROR. 14 SHUTDOWN. 15 RHF I/O ERROR. 16 INCORRECT DEVICE. 17 CODE CONV N/A. (not avail) 18 CONTROL WORD ERROR. 19 INCORRECT DEVICE. 20 EMPTY FILE. 21 NAM INTERFACE ERR. 22 BLK NUM MISCOMPARE. 23 INCORRECT BLK SIZE. 25 BLOCK SEQUENCE ERR. 26 PRU BOUNDS ERROR. 27 INTRA-HOST PRU XFR. 28 ERROR DURING XFR. 29 ABL OUT OF RANGE. 30 APPL NOT VALIDATED. 31 FILE TYPE ERROR. 32 JOB ORIGIN ERROR.		
MFLINK - NETXFR STATUS = 0.	A system or network error occurred.	Inform site analyst.	MFLINK
MFLINK - NO CONNECTION RECEIVED.	The network failed to respond to the application connection request before the timeout period elapsed.	Inform site analyst if problem persists.	MFLINK
MFLINK - NO LID SPECIFIED.	You failed to specify the ST=lid parameter on the first MFLINK command of the MFLINK session.	Correct the MFLINK command and retry.	MFLINK
MFLINK - NO PATHS AVAILABLE TO LID.	No paths are currently available to any remote host with the specified LID through either network subsystem (RHF or NAM).	If message persists, contact site operator. MFLINK retries if RT not specified.	MFLINK
MFLINK - NO REMOTE DIRECTIVES FOUND.	The system is unable to find your MFLINK directives.	Correct the MFLINK command and retry.	MFLINK
MFLINK - PROTOCOL ERROR IN xx.	MFLINK received an incorrect value for the parameter xx.	Inform site analyst if problems result.	MFLINK
MFLINK - RECOVERING ST lid.	An informative message. The system is recovering the files for the mainframe specified by lid.	None.	MFLINK
MFLINK - REMOTE SUBSYSTEM NOT RESPONDING.	The network subsystem (RHF or NAM) cannot establish communication with the remote host.	Contact site operator. MFLINK retries unless RT is specified.	MFLINK
MFLINK - REMOTE SUBSYSTEM RESOURCE LIMIT.	The network subsystem (RHF or NAM) on the remote host is temporarily unable to complete the connection.	Retry. If message persists, contact site operator.	MFLINK
MFLINK - SUBSYSTEM FULL.	The network subsystem (RHF or NAM) is temporarily too busy to process your request.	Contact site operator. MFLINK retries unless RT is specified.	MFLINK

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MFLINK - SUBSYSTEM UNAVAILABLE.	The network subsystem is temporarily too busy to process your request.	Contact site operator. MFLINK retries unless RT is specified.	MFLINK
MFLINK - UNKNOWN NETWORK INTERFACE TYPE.	An internal error was detected by MFLINK.	Inform site analyst.	MFLINK
MFLINK - WRONG REMOTE APPLICATION LEVEL. MFLINK - TRANSFER IMPOSSIBLE TO ST lid.	You entered an MFLINK command with ST=lid; the transfer failed because MFLINK's protocol version or level is incompatible with the Permanent File Transfer Facility (PTF) software at the remote host.	Inform site analyst.	MFLINK
MFQUEUE - COMPLETE. jsn = ROUTED FILE NAME.	Job jsn was successfully routed.	None.	MFQUEUE
MFQUEUE - CONTROL STATEMENT ERROR.	MFQUEUE was unable to process the command. Usually an invalid character or successive separators is the cause of this problem.	Correct the command.	MFQUEUE
MFQUEUE - DIRECTIVE EXCEEDS 256 CHARACTERS	The user supplied remote directive for this command exceeds 256 characters.	Correct the directive.	MFQUEUE
MFQUEUE - DUPLICATE PARAMETER.	Multiple occurrences are not allowed for any of the MFQUEUE parameters.	Correct the command.	MFQUEUE
MFQUEUE - FIRST PARAMETER NOT VALID LFN.	The first parameter must be a valid file name.	Correct the command.	MFQUEUE
MFQUEUE - ILLEGAL DC VALUE SPECIFIED.	The DC=dc parameter is not supported.	Correct the command.	MFQUEUE
MFQUEUE - ILLEGAL DD VALUE SPECIFIED.	Your DD=dd specification is not supported.	Correct the command.	MFQUEUE
MFQUEUE - ILLEGAL ST VALUE SPECIFIED.	The logical identifier (LID) you specified in the ST=lid parameter is not three alphanumeric characters.	Correct the command.	MFQUEUE
MFQUEUE - INVALID DIRECTIVE NAME.	The directive file name must be a valid file name.	Correct the command.	MFQUEUE
MFQUEUE - INVALID PARAMETER.	You specified a parameter that is not allowed on the MFQUEUE command.	Correct the command.	MFQUEUE
MFQUEUE - NO REMOTE DIRECTIVES FOUND.	The directive file was empty or you did not supply remote directives via the PC parameter.	Fix the job so that the directive file is not empty.	MFQUEUE
MFQUEUE - ST PARAMETER MUST BE SPECIFIED.	The ST parameter is not optional. It must appear on each MFQUEUE command.	Correct the command.	MFQUEUE
MFSTAT REJECTED REQUEST	You did not enter the correct PID for the SCOPE 2 mainframe.	Be sure to use only the primary physical identifier (PID) to reference a SCOPE 2 mainframe. You may enter a QUEUE7, ID command to list PIDs currently active.	QUEUE7
MISSING DEMAND FILE ENTRY.	Dayfile message indicating a system problem. The overcommitment algorithm was initiated without a demand file entry having been defined previously.	Rerun the job. If the problem reoccurs, inform a site analyst, write a PSR and include support material so that CDC can duplicate the problem.	RESEX
jsn MISSING RESOURCE.	Dayfile message indicating that a system problem occurred. The system expected but did not find a resource unit assigned to the job with job sequence name jsn. This could occur if the MAG subsystem was stopped while tapes were assigned.	Rerun the job. If the problem reoccurs, inform a site analyst, write a PSR and include support material so that CDC can duplicate the problem.	RESEX
MISSING VSN OR EQUIPMENT ASSIGNMENT.	Dayfile message indicating a system malfunction (expected VSN or equipment assignment was not found).	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
MODE ERROR.	An arithmetic error or CPU/CM parity error has been encountered during processing of job.	Examine relieve 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 a monitor request: - The monitor request call was not recognized by the system. - Bad parameters were supplied in the request. - An incorrect user access was encountered. - An incorrect request was included in the request.	Examine program to determine why an incorrect monitor request call is being made.	1AJ
MTest, ASSIGNED TO filename, VSN=vsnn.	The specified file name is assigned to the tape volume with the specified VSN that is mounted on tape drive TTest (the seven-track (MT) or a nine-track (NT) tape unit with EST ordinal est).	Informative message. Processing of the tape file can begin.	RESEX
MT,Ccc,Eec,Hhhhhhhhh, BCR.	Magnetic tape controller controlware restarted. BCR identifies B.C. RESTART.	None.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MT,Ccc,Eec,Hhhhhhhhh, BFR.	Refer to the READ ID BURST FAILURE message.		
MT,Ccc,Eec,Hhhhhhhhh,BFW.	Refer to the WRITE ID BURST FAILURE message.		
MT,Ccc,Eec,Hhhhhhhhh,BTL.	Refer to the BLOCK TOO LARGE message.		
MT,Ccc,Eec,Hhhhhhhhh, BUSY.	Unit was still busy after one second.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhhh,CMF.	Refer to the CHANNEL MALFUNCTION message.		
MT,Ccc,Eec,Hhhhhhhhh, CON. REJ. OFF.	Connect reject; unable to connect to unit. Unit turned off.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhhh,ERA.	Refer to the ERASE LIMIT message.		
MT,Ccc,Eec,Hhhhhhhhh, FNffff,Pyyyy.	Function ffff was rejected by the controller; yyyy is the address in 1MT where the function was initiated.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhhh, Lbbbb,Bnnnnnn.	The length [bbbb] and block number [nnnnnn] 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,Hhhhhhhhh, LCH.	Load sequence failed on the unit. LCH identifies LOAD CHECK.	Push CLEAR button and reload tape, or inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhhh, MDW.	Indicates controller failure. Channel has been logically turned off and maintenance is required. MDW identifies MARGINAL, DOWN.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhhh, MOP.	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. MOP identifies MARGINAL, OFF.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhhh, MWT.	Refer to the MARGINALLY WRITTEN TAPE message.		
MT,Ccc,Eec,Hhhhhhhhh, NBE.	A noise block was skipped on the tape. NBE identifies NOISE.	None.	1MT
MT,Ccc,Eec,Hhhhhhhhh, NO EOP.	No end-of-operation detected from unit within 1 second.	Inform customer engineer.	1MT
MT,Ccc,Eec,Hhhhhhhhh, NOT READY.	Tape unit dropped ready status.	Make unit ready.	1MT
MT,Ccc,Eec,Hhhhhhhhh, OPA.	Refer to the WRONG PARITY message.		
MT,Ccc,Eec,Hhhhhhhhh, OTF.	Error was corrected as the data was read. OTF identifies ON THE FLY.	None.	1MT
MT,Ccc,Eec,Hhhhhhhhh, PLD.	Refer to the POSITION LOST message.		
MT,Ccc,Eec,Hhhhhhhhh, RCE.	Refer to the REPOSITION CORRECTED ERROR message.		
MT,Ccc,Eec,Hhhhhhhhh, REC.	Previously reported error has been successfully recovered. REC identifies RECOVERED.	None.	1MT
MT,Ccc,Eec,Hhhhhhhhh, SMB.	Single block misposition error occurred during tape positioning backward. This message is informative and the error is not fatal. SMB identifies BID RECOVERY-B.	None.	1MT
MT,Ccc,Eec,Hhhhhhhhh, SMF.	Single block misposition error occurred during tape positioning forward. This message is informative and the error is not fatal. SMF identifies BID RECOVERY-F.	None.	1MT
MT,Ccc,Eec,Hhhhhhhhh, STE.	Error type cannot be determined so actual controller status is returned. STE identifies STATUS.	Inform site analyst.	1MT
MT,Ccc,Eec,Hhhhhhhhh,TCF.	Refer to the TAPE CLEANER FAULT message.		
MT,Ccc,Eec,Hhhhhhhhh,TME.	Refer to the WRITE TAPE MARK FAILURE message.		
MT,Ccc,Eec,Hhhhhhhhh,WVF.	Refer to the WRITE VERIFY FAILURE message.		
MT,Ccc-e-uu,vsn,rw,est,ss,GSgggggggg MT,Ccc,Dddd...d. MT,Ccc,Uuu...u,Tttt. MT,Ccc,Aaaaaaaa. MT,Ccc,Fff,Iii,Bnnnnnn,Lbbbb,Ppppppppp. MT,Ccc,Eec,Hhhhhhhhh,type. or MT,Ccc-e-uu,vsn,rw,est,ss,GSgggggggg. MT,Ccc,Dddd...d. MT,Ccc,Aaaaaaaa. MT,Ccc,Fff,Iii,Bnnnnnn,Lbbbb,Ppppppppp. MT,Ccc,Eec,Hhhhhhhhh,type. or MT,Ccc,-e-uu,vsn,rw,est,ss,GSgggggggg. MT,Ccc,Dddd...d. MT,Ccc,Fff,Iii,Bnnnnnn,Lbbbb,Ppppppppp. MT,Ccc,Eec,Hhhhhhhhh,type. or	Four, five, or six-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 optionally issued to 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. vsn Volume serial number associated with tape on the specified unit.	Refer to the separate listing of the last line message (MT,...,type.) for the appropriate action.	1MT

MESSAGE

SIGNIFICANCE

ACTION

ROUTINE

MT,Ccc-e-uu,vsn,rw,est,Ss,GSggggggggg.
 MT,Ccc,Dddd...d.
 MT,Ccc,Uuu...u,Tttt.
 MT,Ccc,Fff,Iii,Bnnnnnn,Lbbbb,Ppppppppp.
 MT,Ccc,Eec,Hhhhhhhh,type.

rw Read (RD) or write (WR) operation; any operation not involving an actual read or write is listed as a read.

est EST ordinal of the unit on which the tape was last labeled. This is provided only for labeled tapes generated under NOS; otherwise, the field is blank. This field does not represent the EST ordinal of the tape unit that the tape is currently mounted on.

s Channel status.

gggggggg General status of magnetic tape unit. Last byte is block ID. Refer to the tape drive's hardware reference manual for general status information.

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. Refer to the tape drive's hardware reference manual for general status information.

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 message.

uu...u Detailed unit status. Refer to the tape drive's hardware reference manual for general status information.

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,Aaaaaaaa line (for FSC only) contains the additional sense byte status not placed in the detailed status or unit status fields.

cc Channel number.

aaaaaaa Sense byte.

The MT,Ccc,Fff,...Ppppppppp line of the message provides the following information.

cc Channel number; repeated to associate this message with the previous message.

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 occurred.

bbbb Length of block on which error occurred in octal bytes.

pppppppp IMT internal error parameters.

The last line of each message provides the following information.

cc Channel number; repeated to associate this message with the previous message.

ec Octal error code value.

hhhhhhh 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).

type Additional description of the error. Refer to individual listing of the last line message.

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 bit 56 of FET+10 octal. If dt=MT and bit 56 is set, the message is issued.

Ensure accuracy of command.

BLANK RESEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
MULTI-FILE NOT FOUND, filename AT address.	Either LISTLB has reached the end of the multifile set or the requested file was not found. The following additional messages are also given. REQUESTED SECTION xxxx. FOUND SECTION yyyy. or FILE IDENTIFIER NOT FOUND. The filename, address, xxxx, and yyyy given can be ignored.	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 files. 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 1000 cpi, making recovery impossible. The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Retry or inform site analyst.	1MT
MULTIPLE CONCURRENT LOGINS PROHIBITED.	Another terminal session is active under your user name and you are not validated to have more than one terminal session simultaneously.	If you need to do this, contact your site personnel.	NVF
MULTIPLE 7000 JOBS DISALLOWED	You have attempted to submit a second job file when you already have an outstanding SCOPE 2 job in processing.	Wait for the first job to complete before submitting another job.	HELLO7
NETWORK APPLICATION ABORTED, RC = 68. NFETCH: ILLEGAL FIELD MNEMONIC.	Either the field or value parameter in the indicated call was not found.	Correct the call and rerun the job.	FIP
NETWORK APPLICATION ABORTED, RC = 69. NSTORE: ILLEGAL FIELD MNEMONIC.	Either the field or value parameter in the indicated call was not found.	Correct the call and rerun the job.	FIP
NETWORK TYPE DOES NOT SUPPORT ATTRIBUTE COMMAND.	You are attempting to enter NAM/CDCNET commands from a NAM/CCP connection.	Ensure that the attributes you are trying to change can be changed in your network by TRMDEF.	TRMDEF
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 ALTERNATE STORAGE COPY OF FILE.	The disk space for the file cannot be released because the alternate storage image is labeled obsolete or the alternate storage address is not specified in the permanent file catalog.	None.	PFM
NO CLOSING DELIMITER FOUND.	No closing delimiter was found on a string. A string is considered to end after the last nonblank character.	None.	IEDIT
NO DATA FOUND FOR USER NAME.	RECLAIM could not find the calling user in the directory of the database.	Check for correct database file specification.	RECLAIM
NO EXTENDED MEMORY.	A DMPECS or DMDECS command or DED or DEP request was entered and no extended memory field length is assigned to you.	None.	CPMEM
NO FIELD LENGTH TO DUMP.	Self-explanatory.	None.	CPMEM
NO FILE NAME SPECIFIED.	You entered a SECHDR command without specifying a file name.	Reenter the command with the correct file name.	SECHDR
NO FILES FOUND.	No files were found that matched the specified criteria.	None.	CATLIST
NO FILES FOUND FOR SPECIFIED DUMP FILE.	For a COPY, LOAD, or LIST directive, no entries were found in the database for the dump tape or file specified. RECLAIM may try to read the dump file and enter it into the database.	Check your database file specification.	RECLAIM
NO FILES PROCESSED.	See nnn FILE(S) PROCESSED.	None.	MAGNET
NO FILES PROCESSED.	Refer to nnn FILE(S) PROCESSED.	None.	MFILES
NO FILES PURGED.	No files were found that met the specified criteria for purging.	None.	PURGALL
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 HOST AVAILABLE CONTROL CHARACTER+char ENTER char HD TO SEE HOST STATUS	No hosts are available. char The network control character for your terminal	Check the status of the hosts in your configuration.	CCP
NO HOST CONNECTED CONTROL CHARACTER+char ENTER INPUT TO CONNECT TO HOST	You have attempted to disconnect your terminal from a host when you have no connection to a host.	Enter a HD terminal definition command to determine the status of hosts in the system, or enter any input to initiate your connection to the host.	CCP
NO HOST CONNECTED CONTROL CHARACTER+char TERMINAL DISABLED BY NOP	The network operator has disabled your terminal. No further input is possible. char The network control character for your terminal	None.	CCP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
NO HOST SELECTED. CONTROL CHARACTER=char ENTER char HN=NODE NUMBER or HS=NAME TO SELECT HOST	You have not selected a host and a host is available. char The network control character for your terminal	Select a host.	CCP
NO JOB CHARACTERISTICS SPECIFIED.	You specified no parameters or made incomplete parameter specifications on the SETJOB command.	Do not enter the command if you do not want to change any job characteristics. Otherwise, specify the parameter correctly.	CONTROL
NO JOBS FOUND.	The system can find none of your jobs.	None.	ENQUIRE
NO JSN-S RECOVERED.	The system encountered an EDR, EDF, or EOI while reading input directives.	None.	RECOVER
NO LIDS FOUND.	Either the LID specified was not listable, or there were no listable LIDS found for a given PID.	Check to see that you have specified the correct LID and/or PID.	LISTLID
NO LINE NUMBER ON LINE.	Either the primary file or a sequenced READ file contains a line without a line number.	Use an editor to correct the bad line.	IEDIT
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 line length limit.	Check the format of the input file.	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.	COPYC
NO LINES FOUND.	No line in the edit file meets the criteria specified by the edit command.	None.	IEDIT
NO MASS STORAGE AVAILABLE.	No mass storage space of the requested type could be found.	Retry later.	LPM
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 PERMITS.	No permit entries exist for the specified file.	None.	CATLIST
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 READ/DESTINATION FILE SPECIFIED.	No read or destination file was specified in the edit command.	Reenter command with a specified read or destination file.	IEDIT
NO READ FILE FOUND - filename	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 REFERENCES LISTED.	DPL did not reference any system text symbols.	Check for empty or invalid DPL or bad system text.	KRONREF
NO RESPONSE FROM OTHER MAINFRAME	Communications between SCOPE 2 and the SCOPE 2 station have been interrupted by a system problem.	Try to reenter the command at a later time.	QUEUE7
NO SOURCE FILE SPECIFIED.	No file name was specified on the SUBMIT command.	Specify a file name on the SUBMIT command.	SUBMIT
NO SUCH FILE	The file requested in the SUBMIT directive cannot be found.	Check to ensure that the correct file name was entered and that the file exists as a local NOS file.	HELLO7
filename - NO TEMP DEVICE FOUND.	You attempted to create a local copy of an indirect access permanent file, but the file's access level was not within the range of valid access levels for any active temporary storage device.	Inform your site security administrator. The security administrator needs to configure the system so that temporary devices are available for all valid access levels.	PFM
NO TERMINATOR IN COMMAND CALL.	No terminator was found before the first delimiter in a NOTE, BLOCK, or ENTER command.	Correct command and retry.	NOTE
NO USER COMMAND EXECUTED.	You entered a CHARGE command without first entering a valid USER command.	Enter a valid USER command. Then enter the CHARGE command.	VALEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
NO VALID DUMP FOUND ON DUMP FILE.	For a COPY, LOAD, or LIST operation, RECLAIM determined that the file was not a RECLAIM dump file; the directive that caused this message is ignored.	None.	RECLAIM
NO WRITE ENABLE, filename AT address.	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, PD=W on the LABEL command). Otherwise, refer to the description of the message in the additional message line.	1MT
NO 7000 JOB ACTIVE	You issued a STATUS or RESUME directive when no SCOPE 2 job can be associated with your user name.	None.	HELLO7
NOISE BLOCK PROCESSING IN EFFECT.	Noise processing was automatically selected because the output tape noise size exceeded the input tape noise size on a COPY where no noise size was specified.	None.	COPYB
nnnnnnnnnnnnnnnn NOISE BLOCKS DELETED. nnnnnnnnnnnnnnnn NOISE BLOCKS PADDED.	Blocks smaller than noise size on S or L output tape have been deleted/padded.	None.	COPYB
NOISE SIZE IGNORED FOR PE/GE.	An informative message indicating that you entered a LABEL, REQUEST, or ASSIGN command with a noise size (NS parameter) specified for a PE or GE density tape on an ISMT subsystem. The NS parameter is ignored for those tape densities on an ISMT subsystem.	None.	RESEX
NON-BUFFERED EQ-S CHECKPOINTED. RECOVERY ABORTED.	In the recovery process of aborting a CM recovery, 1CK was called to process outstanding checkpoint requests for all nonbuffered devices.	Level 0 deadstart is required.	1CK
NON-EXISTENT EDIT VERB SPECIFIED.	Internal error.	Submit a PSR with information on how to duplicate the problem.	IEDIT
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
NDRERUN/RERUN INCORRECT FROM INTERACTIVE JOBS.	You entered a NDRERUN or RERUN command from an interactive terminal. The command is ignored.	None.	QFM CONTROL
NOT ALL EQUIPMENT SERVICEABLE.	The number of batch I/O equipments defined in the EST exceeds an assembly constant. It will not be possible to access some equipment.	Modify the EQP deck or contact CYBER Software Support.	110.
filename NOT DECLARED NRANDOM.	An EOF was encountered on the nonrandom file filename.	Verify that filename is in correct format.	LIBEDIT
jsn/ujn NOT FOUND.	The system could not find a file with the specified job sequence name (jsn) and user job name (ujn).	Check to see if you specified the correct JSN and UJN.	QGET DROP
filename NOT FOUND.	RESTART was unable to retrieve the named file.	Verify that the filename is correct.	RESTART
filename NOT FOUND. or username NOT FOUND.	One of the following occurred: - 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
n NOT FOUND.	n, the specified UJN or JSN, was not found.	None.	ENQUIRE
jsn NOT FOUND.	The system could not find a file with the specified job sequence name (jsn).	Check to see if the file is still in the queue.	QGET

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ujn NOT FOUND.	The system could not find a file with the specified user job name (ujn).	Check to see if the file is still in the queue.	QGET
jsn NOT FOUND.	The system could not find the job with the specified JSN.	Check to see if you specified the correct JSN.	DROP
ujn NOT FOUND.	The system could not find the job with the specified UJN.	Check to see if you specified the correct UJN.	DROP
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 write a PSR.	SFP
NOT MULTI-MAINFRAME ENVIRONMENT	The user called QUEUE7 in a non-multi-mainframe environment.	None. QUEUE7 cannot be used in a non-multimainframe environment.	QUEUE7
filename 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
filename NOT ON MASS STORAGE.	filename was not processed because it could not be found on mass storage.	Move file to mass storage or correct the command and retry.	MFILES
jsn NOT RECOVERABLE - SECURITY CONFLICT.	The specified job had a higher access level limit than the current job, and therefore could not be recovered by the current job.	None.	RECOVER
NOT VALID TO DOWNGRADE DATA.	You attempted to lower the access level of a permanent file, but you are not validated to downgrade files.	Contact your site security administrator if you need to be able to downgrade data.	PFM
NOT VALIDATED FOR REQUESTED ACCESS LEVEL.	User has specified an access level outside the user's access level validation.	Use a validated access level or site must validate user for additional access level.	RESEX
NOT VALIDATED FOR WRITING UNLABELED TAPES.	User has not been validated for writing on unlabeled tapes.	Either use a labeled tape or site must validate user for SAV=CULT MODVAL privilege.	RESEX
NOT VALIDATED TO SET XD/XT.	You specified an XD or XT parameter in a CHANGE, DEFINE or PERMIT command or macro without having validation to use these parameters.	Retry without XD or XT parameter, or obtain validation and retry.	PFM
NOTICE*** DATA READ ERROR.	Read error caused loss of words in the dayfile.	None.	DAYFILE
MTest, ASSIGNED TO filename, VSN=vsn	See MTest, ASSIGNED TO filename, VSN=vsn.	None.	RESEX
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 occurred: - 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 conflicts with currently assigned resources (job would deadlock itself).	Check the description of the RESOURC command in section 7. Reduce resource demand which causes conflict.	RESEX
nnnnn OF ttttt FILES PURGED.	Out of a total of ttttt files in your permanent file catalog, nnnnn files were purged.	None.	PURGALL
OFFSW, x FROM JOB jsn	An OFFSW command was issued to the job by the job jsn. x represents the sense switches that have been turned off.	None.	1RI
OFFSW, x FROM OPERATOR	An OFFSW command was issued to the job by the console operator. x represents the sense switches that have been turned off.	None.	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
ONLY CORRECT PARAMETERS ARE *ON* OR *OFF*.	The parameter you specified was not recognized.	Check command description and retry.	CONTROL
ONSW,x FROM JOB jsn.	An ONSW command was issued to the job by the job jsn. x represents the sense switches that have been turned on.	None.	1RI

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ONSW, x FROM OPERATOR	An ONSW command was issued to the job by the console operator. x represents the sense switches that have been turned on.	None.	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 DROP.	Informative message indicating that the operator dropped the job.	None.	1AJ
OPERATOR DROP.	The system operator dropped RECLAIM.	Inform site analyst.	RECLAIM
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 exit processing unless the job has extended reprieve. A job with extended reprieve processing is reprieved once. Exit processing is allowed.	Correct job as needed and rerun.	RECLAIM 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 system operator restarted the job.	None.	RECLAIM
OUTPUT INCOMPLETE	A hardware or software problem occurred.	Reenter the command. If the same message is returned, inform the site analyst.	QUEUE7
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 the 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.	Contact your site support or CYBER Software Support.	1AJ
OVERLAY NOT FOUND.	The specified overlay was not found.	Verify that the file with the specified overlay is local to the job.	1AJ
OVERLAY NOT FOUND IN LIBRARY - ovlname.	The overlay ovlname was not found in the system library.	Verify that the call is to a valid overlay.	1AJ
DVL	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 COMPLETE.	Informative message indicating that the program has completed processing.	None.	PACK
PACK PARAMETER ERROR.	The PACK command contains too many or no file names.	Check the format of the PACK command in section 9.	PACK
PARAMETER *ON* OR *OFF* REQUIRED.	No parameter was specified on ERRMSG command.	Retry with a parameter.	CONTROL
PARAMETERS CE AND CP ARE MUTUALLY EXCLUSIVE.	You attempted to specify both the CE and the CP parameter on the same CHANGE command. They cannot be specified together.	Do two separate CHANGE commands to perform the two functions.	PFILES
PARAMETERS *ON* AND *OFF* ARE MUTUALLY EXCLUSIVE.	You specified both the ON and OFF parameters on the same command. Only one is allowed.	Retry with either ON or OFF.	CONTROL
nnnnnnnnnn 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 correctly recovered.	COPYB
PARITY ERROR.	The tape could not be read/written correctly. The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Retry or inform site analyst.	1MT
PARITY ERROR, JOB HUNG.	A CM parity error occurred in the job's field length. The job is hung to freeze the affected area.	Contact your site support or CYBER Software Support.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
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 EXPIRATION CHANGE INCORRECT.	The expiration date or expiration term value you entered is incorrect on your PASSWOR command or you are not allowed to change your password expiration date.	Retry using the correct expiration date or term.	MODVAL
PASSWORD RANDOMIZATION NOT PERMITTED FROM BATCH.	You entered a PASSWOR command from a non-interactive job when password randomization is required. When password randomization is required, PASSWOR can only be entered from an interactive job.	Resubmit PASSWOR from an interactive job.	MODVAL
PASSWORD TOO LONG.	You entered a PASSWOR command with a new password that is too long. The new password cannot exceed seven characters including randomization characters.	Correct the new password specification and resubmit.	MODVAL
PASSWORD TOO SHORT.	Either you have password randomization required and specified fewer than two nonrandom characters for the new password specification in a PASSWOR command or the new password was shorter than the minimum password length required by the site.	Correct the new password specification and resubmit.	MODVAL
PAUSE FROM JOB jsn	A PAUSE command was issued to the job by the job jsn.	None.	IRI
PAUSE FROM OPERATOR	A PAUSE command was issued to the job by the console operator.	None.	IRI
pfm PERMANENT ERROR.	The specified file resides on alternate storage 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. If the file is on MSE, the site analyst may be able to correct the problem with SSDEBUG; otherwise, the file must be purged or reloaded.	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 file must be staged from alternate storage to disk and either CARTRIDGE PF STAGING or TAPE PF STAGING, or both, have been disabled.	Contact site personnel to determine why staging has been disabled.	PFM
PF UTILITY ACTIVE.	The operation was not attempted because a permanent file utility was currently active.	Wait until the PF utility is not active and retry.	PFM
PFM ABORTED.	PFM was unable to continue due to an operator OVERRIDE, or CPUPFM aborted prior to completing its transfer.	Inform site analyst.	PFM
PID - yyy DOES NOT EXIST IN THE LID TABLE.	The specified PID could not be found in the LID table.	Check to see if you have specified the correct PID.	LISTLID
PL ERROR IN DECK deckname.	The program library specified by the P option in the KRONREF command was incorrectly formatted.	Check that the PL is a MODIFY OPL.	KRONREF
POSITION ERROR ON-filename.	File filename was not repositioned after being checkpointed because CHKPT detected an address error.	None.	CHKPT
POSITION LOST, filename AT address.	A block ID mismatch exists or the block ID window is all ones. This indicates that a NOS terminator label mismatch exists. In either case, the position on the tape cannot be established and the job will terminate.	Clean the tape path.	1MT
POSTAMBLE ERROR.	A missing or defective postamble was detected at 1800 cpi. The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Retry or inform site analyst.	1MT
PP ABORT.	System error in PPU.	Examine the job dayfile to determine the cause.	RECLAIM
PP CALL ERROR.	The monitor detected an error in a CPU request for PP action.	Verify that the correct PP 1AJ call is issued.	RECLAIM
PP CALL ERROR.	Error detected in PP call due to hardware or RECLAIM error.	Inform site analyst.	RECLAIM
PREMATURE EOR/EOF ON SOURCE FILE.	The system encountered an end-of-record (EOR) or end-of-file (EOF) in the source file before it processed all the information.	Remove the extra EORs and EOFs from the source file.	DOCMENT
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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
PROCEDURE pppppp HAS NO FORMAL PARAMETERS	A line mode call requesting help was made to an interactive procedure that has no parameters defined. Execution continues following the message.	None.	CCL
PROCESSING OPTION CONFLICT.	Both ring enforcement options (PD=R and PD=W) or more than one EDT option (PD=I, PD=P, and PD=S) has been specified. This message can also occur if PD=I or PD=P is specified for an I- or SI-format tape.	Retry using the correct parameter and syntax.	RESEX
PROCESSING OPTION NOT APPLICABLE.	The processing options PD=R and PD=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 PD=T is allowed on a COPYB command that generates an E or B format tape.	Refer to the COPY or COPYB command description in section 9, correct the command, and retry.	COPYB
PROCESSOR DETECTED MALFUNCTION.	A hardware error (error code=20) such as a data or address parity error on double bit error has occurred.	Analyze job dumps, maintenance registers, and the error log or HPA reports to determine the cause of the error.	1AJ
PROCESSOR STATE ERROR.	A hardware error, or software error resulting in a hardware error, occurred.	Analyze job output, dumps, listings, maintenance registers and the error log to determine the cause of the error.	1AJ
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 address.	The monitor detected a program stop instruction at the specified address.	None.	1AJ
PROJECT EPILOGUE NOT FOUND.	The epilogue file defined for your charge and project numbers could not be found.	Contact site personnel.	VALEX
PROJECT NUMBER EXPIRED.	Dayfile and output file message indicating that the project number expiration date has occurred.	None.	VALEX
PROJECT PROLOGUE FILE NOT LOCAL.	The local file ZZZZPR1 was returned prior to the time that it was needed.	Contact site personnel.	VALEX
PROJECT PROLOGUE NOT FOUND.	The prologue file defined for your charge and project numbers could not be found.	Contact site personnel.	VALEX
PROLOGUE AND SHELL CONFLICT.	A shell program with the no-abort option set was active at the time you attempted to specify charge and project numbers which had a prologue defined.	Specify charge and project numbers which do not have a prologue defined.	VALEX
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.	The specified queue file could not be attached to your job, because you have used too much local file space.	Return some local files and try again.	QGET
PRU LIMIT, FILE filename AT address.	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.	1MS
PRUS REQUESTED UNAVAILABLE.	On a DEFINE, no device currently has available the amount of space requested by the S parameter. On a secured system, no device with the proper access level currently has the amount of space requested.	If possible, reduce the number of PRUs specified by the S parameter or retry at another time when space might be available.	PFM
PTFS - APPLICATION CONNECTION TIMEDOUT.	The remote host did not respond in the allotted time.	None.	PTFS
PTFS - CHARGE REQUIRED.	You must specify a CHARGE directive and you did not do so.	Restart your session and include a CHARGE directive.	PTFS
PTFS - CHARGE RESTRICTED TO DEFAULT.	On a CHARGE command, you specified a charge and project number which was not the default that you were validated for and your validation restricts you to that default.	Use CHARGE(*) rather than specifying a charge and project number.	PTFS
PTFS - DEFAULT CHARGE IS NULL.	The default charge number for the current user name is undefined and a change is required.	Use a valid charge and project number.	PTFS
PTFS - DIRECTIVE TOO LONG.	Your directive is too long. A directive cannot exceed 80 characters.	Shorten your directive and retry.	PTFS
PTFS - ERROR IN ARGUMENTS.	Either one or both of the following occurred on a SAVE or a DEFINE directive on a MFLINK command: - You specified both XD and XP parameters. - You specified an XD or XP parameter without specifying a PW parameter.	Correct the SAVE or DEFINE directive and reenter the MFLINK command.	PTFS

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
PTFS - ERROR IN EXPIRATION DATE.	There was an error in the expiration date specified on a SAVE or a DEFINE directive on a MFLINK command.	Correct the expiration date on the SAVE or the DEFINE directive and reenter the MFLINK command.	PTFS
PTFS - EXECUTE ONLY FILE.	You attempted to transfer an execute-only file.	Change permanent file access to allow read permission.	PTFS
PTFS - FILE ALREADY PERMANENT.	You attempted to save a file that is already permanent.	Purge the file and retry.	PTFS
PTFS - FILE IS DIRECT ACCESS.	You attempted to alter a direct access file with a REPLACE or APPEND directive. This is not allowed.	Change directive and retry.	PTFS
PTFS - FL TOO SHORT FOR PROGRAM.	There is a system error in the remote host.	Inform site analyst.	PTFS
PTFS - HOST NOT SPECIFIED TYPE.	Your job assumes a non-NOS remote host and you have been linked to a NOS remote host.	Inform site analyst.	PTFS
PTFS - INCOMPLETE CONTINUED DIRECTIVE.	The remote host expected the text string to be a continuation of the previous string and it was not.	Correct the syntax of your directives record with emphasis on the continuation lines.	PTFS
PTFS - INVALID keyword=value.	You specified an incorrect parameter (keyword=value).	Correct and retry.	PTFS
PTFS - INVALID ACCESS VALIDATION.	Your USER directive has an incorrect user name or the specified user name does not have job processing privileges on the remote host.	Ensure that the user name is correct and retry. Inform site analyst if the problem persists.	PTFS
PTFS - INVALID ALTERNATE CATLIST PERMISSION.	You specified an incorrect value for the AC parameter on a SAVE, DEFINE or CHANGE command.	Specify a correct value and retry.	PTFS
PTFS - INVALID BACKUP/RESIDENCE REQUIREMENT.	You specified an incorrect BR=br or PR=pr parameter on one of your directives.	Correct the parameter and retry.	PTFS
PTFS - INVALID DATA DECLARATION TYPE.	You specified an incorrect DD=dd parameter.	Correct the DD=dd parameter and retry.	PTFS
PTFS - INVALID DEVICE SPECIFICATION.	You specified an incorrect R=r parameter on your directive.	Correct and retry.	PTFS
PTFS - INVALID DIRECTIVE.	The remote host does not recognize the directive you specified.	Ensure that the syntax of the directive is correct and retry.	PTFS
PTFS - INVALID MODE/CATEGORY.	The mode or category you specified for the file are not defined.	Correct and retry.	PTFS
PTFS - INVALID xxxxxx=NO VALUE.	Argument xxxxxx was entered without a value.	Correct remote directive and retry.	PTFS
PTFS - LID UNAVAILABLE.	The requested LID is not a host LID, is disabled, or is a store-forward LID.	Inform site analyst.	PTFS
PTFS - MISSING CHARGE/PROJECT.	The CHARGE directive did not have the required charge and project number.	Correct and retry.	PTFS
PTFS - MISSING USER NAME.	The USER directive did not contain the required user name.	Correct and retry.	PTFS
PTFS - MULTIPLE FILE TRANSFERS REQUESTED.	You have more than one file transfer directive in a given directive record. This is not allowed.	Place the file transfer directives in separate records, one file transfer directive per record.	PTFS
PTFS - error message returned by NOS	An error message has been received from NOS by PTFS. PTFS takes this message, prefaces it with a 'PTFS - ', and returns the new message to you.	Refer to the message returned by NOS for the appropriate action (that is ignore the letters "PTFS -" and look up the message itself alphabetically in this appendix).	PTFS
PTFS - PARAMETERS CE AND CP ARE MUTUALLY EXCLUSIVE.	The parameters CE and CP may not both be specified on the same CHANGE command.	Use two separate CHANGE commands, with one specifying CP and the other specifying CE.	PTFS
PTFS - PF NAME REQUIRED.	You did not specify the permanent file to be processed.	Correct and retry.	PTFS
PTFS - PF REQUEST COMPLETE.	The remote host successfully processed your permanent file request.	None.	PTFS
PTFS - RECOVERY DIRECTIVE AFTER FILE TRANSFER.	Your MFLINK directives are not in the proper sequence. The USER, CHARGE, or PACKNAM directive must precede the file transfer directive.	Resequence your directives and retry.	PTFS
PTFS - SECONDARY USER DIRECTIVES DISABLED.	You are not authorized to specify secondary USER directives.	Get site personnel to authorize you or do not attempt to use secondary USER directives.	PTFS

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
PTFS - UNKNOWN ACCESS LEVEL NAME.	You entered an unknown access level name on a SAVE or a DEFINE directive through a MFLINK command.	Correct the access level name on the SAVE or the DEFINE directive and reenter the MFLINK command.	PTFS
PTFS - USER DIRECTIVE REQUIRED FIRST.	You must first specify a USER directive to access permanent files on remote NOS hosts.	Correct and retry.	PTFS
PURGALL ABORTED.	PURGALL aborted prematurely.	Check dayfile for more detailed messages.	PURGALL
PURGALL COMPLETE.	The files have been purged as requested.	None.	PURGALL
PURGING #####	Informative message issued by PURGALL as each file is purged, if the DF parameter is specified.	None.	PURGALL
PURGING filename.	File filename has been purged. PURGALL issues this message if the DF parameter is specified.	None.	PURGALL
QAC ERROR ENCOUNTERED.	System error.	Inform site analyst.	ENQUIRE
QAP - INCORRECT REQUEST.	Bad parameter in QAP call.	Inform CYBER Software Support.	QAP
QAP - USER ACCESS NOT VALID.	You tried to perform an operation for which you are not authorized (for example, trying to use PFC arrays when not authorized to do so).	Ensure accuracy of command or determine proper validation requirements.	QAP
OFM FILE NAME ERROR.	The file name specified is not a valid file name.	Verify file name.	OFM
OGET COMPLETE.	Informative only.	None.	OGET
QTF, -n- qfn ACQUIRED, DC=dc, ST=did, DD=sid.	The system has acquired local queue file qfn on connection number n from the local queue with disposition code dc for transfer to destination LID did with source LID sid.	None.	QTF
QTF, -n- qfn CONNECTING TO pid.	The system is attempting to establish communications with remote host pid. n QTF connection number qfn Queued file name pid Physical identifier	None.	QTF
QTF, -n- qfn CONNECTION REJECTED, REASON CODE = xxx.	The system was unable to establish a connection to the remote host. n QTF connection number qfn Queued file name xxx Reason code (decimal) (For more details about reason codes, refer to the descriptions of Application Connection Reject Messages in the Supervisory Messages chapters of the Network Access Method Reference Manual (60499500) and the RHF Access Method Reference Manual (60459990).)	If problem persists, contact remote operator to ensure remote subsystem and QTFs are operational; otherwise contact site analyst.	QTF
QTF, -n- qfn EVICTED.	The system evicted the local queue file qfn to connection number n from the local queue.	None.	QTF
QTF, -n- qfn INVALID QUALIFIER OR PARAMETER.	The system received an invalid network message from the remote host.	Inform site analyst.	QTF
QTF, -n- qfn REQUIRED PARAMETER MISSING.	The remote host failed to send a required parameter on a network message. n QTF connection number qfn Queued file name	Inform site analyst.	QTF
QTF, -n- qfn SENT TO PID xxx, PID yyy RESPONDING.	The subsystem network description table does not agree with the remote host PID returned by the remote host QTFs. The file transfer may complete. n QTF connection number qfn Queued file name xxx Expected physical identifier yyy Actual physical identifier	Inform site analyst.	QTF
QTF, -n- qfn UNEXPECTED ACQUIRE ERROR CODE = nnnB.	A system error. n QTF connection number qfn Queued file name	Inform site analyst.	QTF
QTF, -n- qfn UNRECOGNIZED DISPOSITION CODE.	The system has acquired a file from the local queue with an unrecognized disposition code. See previous message for disposition code value. n QTF connection number qfn Queued file name	None.	QTF
QTF(S), -n- qfn CONNECTION BROKEN.	The remote host partner or the remote subsystem broke the network connection usually due to an involuntary termination or failing network hardware. n QTF connection number qfn Queued file name	If error persists, inform site analyst and customer engineer.	QTF QTFs

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
QTF(S), -n- qfn CONNECTION ESTABLISHED.	Indicates state of connection to remote host partner. n QTF connection number qfn Queued file name	None.	QTF QTF5
QTF(S), -n- qfn CONNECTION TIMED OUT.	Remote host partner did not respond within the allowed time span. Transfer will be retried. n QTF connection number qfn Queued file name	If error persists, inform site analyst and disable remote host pid in subsystem ID table.	QTF QTF5
QTF(S), -n- qfn CONNECTION TO pid ABORTED.	Indicates state of connection to remote host partner. n QTF connection number qfn Queued file name pid Physical identifier	None.	QTF QTF5
QTF(S), -n- qfn CONNECTION TO pid ENDED.	Indicates state of connection to remote host partner. n QTF connection number qfn Queued file name pid Physical identifier	None.	QTF QTF5
QTF(S), COUNT OF ACTIVE CONNECTIONS DECREMENTED BELOW ZERO.	The application received a disconnect but had no paths active.	Inform site analyst.	QTF QTF5
QTF(S), -n- qfn ERROR IN FILE TRANSFER.	Transfer of file qfn was unsuccessful. n QTF connection number qfn Queued file name	If error persists, inform site analyst.	QTF QTF5
QTF(S), -n- qfn FC/BRK RECEIVED.	The remote host partner broke the network connection. Usual cause is detection of an unrecoverable protocol anomaly. n QTF connection number qfn Queued file name	If error persists, inform site analyst.	QTF QTF5
QTF(S), -n- qfn FC/NAK RETRY COUNT EXCEEDED.	The subsystem was unable to deliver a network message. n QTF connection number qfn Queued file name	Inform site analyst.	QTF QTF5
QTF(S) -n- qfn FIRST KEYWORD IS NOT *ROUTE*.	The MFQUEUE routing directive must begin with the keyword ROUTE. n QTF connection number qfn Queued file name	Correct the MFQUEUE routing directive.	QTF QTF5
QTF(S) -n- qfn INCORRECT EC FOR THIS QUEUE TYPE.	The external characteristics [EC] value specified on a MFQUEUE routing directive does not match the disposition code of the file. n QTF connection number qfn Queued file name	Correct the MFQUEUE routing directive using a print-type EC value if the file is a print file, or a punch-type EC value if the file is to be punched.	QTF QTF5
QTF(S) -n- qfn INCORRECT VALUE (xxx).	You specified an incorrect keyword xxx on an MFQUEUE routing directive. n QTF connection number qfn Queued file name	Correct the MFQUEUE routing directive.	QTF QTF5
QTF(S) -n- qfn INCORRECT VALUE (xxx=yyy).	You specified an incorrect value yyy for keyword xxx on an MFQUEUE routing directive. n QTF connection number qfn Queued file name	Correct the MFQUEUE routing directive.	QTF QTF5
QTF5, -n- qfn INVALID ATTRIBUTE IGNORED.	QTF5 received a network message from the remote host QTF with an unrecognized parameter. Transfer may complete. n QTF connection number qfn Queued file name	Inform site analyst.	QTF5
QTF(S), -n- qfn INVALID SEQUENCE RECEIVED.	A network message from the remote host partner was received that was not in sequence. n QTF connection number qfn Queued file name	Inform site analyst.	QTF QTF5
QTF(S), JOB ORIGIN ERROR.	QTF or QTF5 was improperly initiated by user command.	None.	QTF QTF5
QTF(S), -n- qfn MESSAGE FROM REMOTE HOST - QTF(S), -n- qfn xxx.	xxx is the text of a message received from the remote host partner. n QTF connection number qfn Queued file name	None.	QTF QTF5
QTF(S), -n- qfn NETWORK BLOCKS OUT OF SEQUENCE.	A network message from the remote host partner was received that was not in sequence. n QTF connection number qfn Queued file name	Inform site analyst.	QTF QTF5

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
QTF(S), -n- qfn NETXFR ERROR rc - rejmess	An error occurred during file transfer. n QTF connection number qfn Queued file name rc Reason code rejmess Reject message rc rejmess 03 CONNECTION BROKEN. 04 PROTOCOL ERROR. 05 TIMEOUT EXPIRED. 06 BLOCK NOT SENT. 07 TOO MANY TRANSFERS. 08 ACN OUT-OF-RANGE. 09 CID ERROR. 13 IDLEDOWN ERROR. 14 SHUTDOWN. 15 RHF I/O ERROR. 16 INCORRECT DEVICE. 17 CODE CONV N/A. (not avail) 18 CONTROL WORD ERROR. 19 INCORRECT DEVICE. 20 EMPTY FILE. 21 NAM INTERFACE ERR. 22 BLK NUM MISCOMPARE. 23 INCORRECT BLK SIZE. 25 BLOCK SEQUENCE ERR. 26 PRU BOUNDS ERROR. 27 INTRA-MDST PRU XFR. 28 ERROR DURING XFR. 29 ABL OUT OF RANGE. 30 APPL NOT VALIDATED. 31 FILE TYPE ERROR. 32 JOB ORIGIN ERROR.	Inform site analyst if problem persists.	QTF QTF5
QTF(S) -n- qfn NO USERNAME FOR *DC*WT/TT* FILE.	QTF5 is unable to accept a file for the wait queue because no user name exists with which to associate the file. n QTF connection number	Specify UN=username parameter on MFQUEUE routing directive.	QTF QTF5
QTF(S) -n- qfn ROUTE ERROR nnB-xxxx.	QTF5 was unable to queue the queued file [qfn] due to one of the reasons in the following list. nn is the DSP error code, and xxxx is the reason. INVALID DISPOSITION CODE. INVALID ST OR DO LID. INVALID TERMINAL ID. INVALID FORMS CODE. INVALID JOB COMMAND. INVALID DATA DECLARATION. INVALID INTERNAL CHAR. INVALID EXTERNAL CHAR. INVALID SPACING CODE. TOO MANY DEFERRED JOBS. INVALID USER ACCESS. INVALID USER COMMAND. QTF FULL [RETRY LATER]. DISK FULL [RETRY LATER]. INVALID OWNER USER. INVALID CREATION USER. FILE IS EMPTY. Refer to section 10 of Volume 4 of the NOS Version 2 Reference Set for DSP error codes and descriptions.	If RETRY LATER, no action is required; QTF will periodically retry to transfer the file until the temporary condition clears. For all other reasons, the file is evicted and the log file is returned to originator. Correct the ROUTE command or MFQUEUE routing directive, if possible; otherwise inform site analyst.	QTF QTF5
QTF(S), SUPERVISORY MESSAGE RECEIVED FOR INACTIVE ACN. QTF(S), HDR = first two words of header	A supervisory message that was received contained a connection number that was not currently active.	Inform site analyst.	QTF QTF5
QTF(S), -n- qfn TRANSFER xxxxxxxxxx BY REMOTE HOST. QTF(S), -n- qfn REASON CODE = nnnnnn. or QTF(S), -n- qfn REASON CODE = nnnnnn - yyy.	QTF or QTF5 has been informed by its remote host partner that the current file transfer cannot be initiated or completed. xxxxxxxxxx may be either REJECTED, TERMINATED, or ABORTED. n is the QTF connection number, nnnnnn is the reason code, and yyy is one of the following reasons. A description follows each yyy entry. {CONTACT SITE ANALYST}. An unexpected reason code was received. PROTOCOL ERROR. The remote host partner detected a protocol anomaly. TIME-OUT MATURED. Remote host partner did not receive message within the allowed time span. SENDER PROBLEMS. Unspecified problems were encountered on the sending (QTF) side. RECEIVER PROBLEMS. Unspecified problems were encountered on the receiving (QTF5) side. FILE SIZE TOO BIG. The file is too large to be accommodated on the remote host or not enough space was preallocated. INVALID USERNAME/ACCOUNT. An invalid or missing account or username was detected by the remote host partner.	If fatal transfer error, file is evicted and QTF log file returned to originator. In all other cases, transfer will be retried. If error persists, inform site analyst.	QTF QTF5

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
	<p>UNSPECIFIC TRANSFER Insufficient information was received by the remote host partner to allow the transfer to continue.</p> <p>QUEUE TYPE UNAVAILABLE. The disposition code requested on the ROUTE command or MFQUEUE routing directive could not be processed on the remote host. For example, a print file was attempted to be transferred to a CYBER 200 remote host.</p> <p>UNACCEPTABLE ATTRIBUTES. An invalid attribute value was received by the remote host partner.</p>		
QTF(S), -n- qfn UNKNOWN *PI* PARAMETER, DEFAULT USED.	<p>The remote NOS 2 system received a print file with an unknown PI (print-image) parameter. The file will be printed using the default PI parameter.</p> <p>n QTF connection number qfn Queued file name</p>	None.	QTF QTF5
QTF(S), -n- qfn UNKNOWN STATE-OF-TRANSFER SENT BY REMOTE HOST. QTF(S), -n- qfn REASON CODE = nnnnnn - (CONTACT SITE ANALYST).	<p>An unrecognized value, nnnnnn, for the state-of-transfer attribute was received on a network message from the remote host partner.</p> <p>n QTF connection number qfn Queued file name</p>	Inform site analyst.	QTF QTF5
QTF(S), -n- qfn UNRECOGNIZED COMMAND RECEIVED xx.	<p>The command sent by the remote host partner is not recognized.</p> <p>n QTF connection number qfn Queued file name xx Command number received</p>	Inform site analyst.	QTF QTF5
QTF(S), -n- qfn UNRECOGNIZED MESSAGE RECEIVED.	<p>The network message received from the subsystem is not recognized.</p> <p>n QTF connection number qfn Queued file name</p>	Inform site analyst.	QTF QTF5
QUEUE LIMITS EXCEEDED ON REMOTE MAINFRAME.	<p>A batch job was entered from a card reader or routed from another mainframe with an ST parameter on the job command, but the job could not run on the specified remote mainframe because the user had too many jobs active on that mainframe already.</p>	Retry later.	1AJ
QUEUE7 ARGUMENT ERROR	<p>You entered the QUEUE7 command incorrectly.</p>	<p>Check the command format, then reenter the command. You may enter a QUEUE7,SYNTAX command to display the command format.</p>	QUEUE7
RANDOM INDEX ERROR.	<p>The random disk address of the permit sector is in error (error log and dayfile message). This may be a problem with your program.</p>	<p>Check your program. If it does not use the CATLIST macro, inform site analyst.</p>	PFM
READ AFTER WRITE, filename AT address.	<p>You attempted to read a tape on which the last operation was a write.</p>	<p>Ensure accuracy of tape positioning commands [BKSP, BKSPRU, SKIPFB, or REWIND required to read after write].</p>	1MT
READ FILE BUSY - filename	<p>The read file is found to be busy (direct access file only).</p>	<p>Retry after file is not busy.</p>	SUBMIT
READ FILE EMPTY.	<p>The READ file specified does not exist or contains no data.</p>	<p>Reenter command with a correct read file.</p>	IEDIT
READ ID BURST FAILURE, filename AT address.	<p>This message occurs if either of the following is true:</p> <ul style="list-style-type: none"> - The tape is assigned to a control point in PE (1600 cpi) mode and the drive is unable to read the ID burst at load point. - The tape is assigned to a control point in GE (6250 cpi) mode and one of the following has occurred: <ul style="list-style-type: none"> - The drive is unable to read the ID burst at load point. - The drive is unable to set the automatic gain control using the automatic read amplitude (ARA) burst. 	<p>Load the tape on another drive or use a different tape.</p>	1MT
READY.	<p>The system is ready to process the next command. This message is not issued in the BATCH subsystem.</p>	<p>Enter your next command.</p>	EDIT IAFEX
READY DROP, filename AT address.	<p>Unit dropped ready status.</p>	None.	1MT
RECLAIM ABORTED.	<p>RECLAIM has been aborted due to an error or user intervention.</p>	<p>Check the previous message for a description of the error.</p>	RECLAIM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
RECLAIM ARGUMENT ERROR.	An invalid argument was detected in the RECLAIM command.	Check the RECLAIM command for correct parameters, values, etc.	RECLAIM
RECLAIM COMPLETE.	Normal completion.	None.	RECLAIM
nnnnnnnnnn RECORD SPLITS OCCURRED.	Multiple blocks per record have been written on an S or L output tape.	None.	COPYB
RECORD TOO LARGE ON filename.	An input record was encountered that exceeded S or L output tape block size.	Reduce input record size, use COPY command to increase S or L tape block size, or allow record splitting with PD parameter.	COPYB
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
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 a *IGNORE directive listing the records that are not to be used.	LIBEDIT
RECOVERY COMPLETE...CONTINUE	Job recovery was successful.	Issue a STATUS directive to determine the state of the job, then proceed.	HELL07
RECOVERY RERUN.	The system is rerunning the job following a level 3 deadstart.	No action required.	IAJ
REMOVABLE PACKS OVERCOMMITMENT.	Removable pack request without NA selected caused resource overcommitment.	Retry later or retry with NA parameter on ATTACH, DEFINE, etc., with PN parameter specified.	RESEX
RENAME NOT ALLOWED FOR PROC OR TEXT RECORD.	A RENAME of a PROC or TEXT type record was attempted; this is not allowed.	None.	LIBEDIT
REPEAT..	Data will now be accepted by the network.	None.	CCP
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).	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLoad on the device.	PFM
REPOSITION CORRECTED ERROR, filename AT address.	The tape drive is writing in a marginal state. This message is intended to prevent the generation of a marginal tape by the subsystem and its devices. This message is to be considered a marginal drive indicator (MDI) and requires immediate action.	Clean the drive and test to insure proper operation. Inform the customer engineer.	1MT
REPRIEVE BLOCK ERROR.	An address is out of range or there is an incorrect 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. (IAJ issues this message for all errors except terminal interrupts processed by IRI.)	Ensure parameter block is correct.	IAJ
REPRIEVE CHECKSUM BAD.	The computed checksum does not agree with the checksum specified in the parameter block at the time of the error. (IAJ issues this message for all errors except terminal interrupts processed by IRI.)	Ensure interrupt handler is still intact. Ensure that code in the area for which checksum was computed has not changed.	IAJ
REQUEST IS VALID ONLY FROM A NETWORK TERMINAL.	You attempted to enter an APPSW command from a job that was not of interactive origin type or from a terminal not connected to NAM (for example, through RDF). To use the APPSW command, your job must be of interactive origin type and your terminal must be connected to NAM.	Login to IAF from a terminal connected to NAM.	TLX
REQUEST UNDEFINED ON DEVICE, filename AT address.	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 a valid device IMS is specified.	
REQUESTED SECTION section.	This is the second line of a three-line message: MULTIFILE NOT FOUND, filename AT ADDRESS. REQUESTED SECTION section. FOUND SECTION section. The system has reached the end of a multifile set.	None.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
REQUESTING DUMP FILE.	RECLAIM is requesting the dump file specified by the TN or DN option. RECLAIM will try to read it and rebuild the database entries for this dump file.	Wait for RECLAIM to finish the request and issue another message.	RECLAIM
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
RESEQ COMMAND 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 (ZZZZGO-ZZZZG7).	Rename file and reenter command.	IEDIT
RESERVED FILE NAME.	File name specified on SDRT command is reserved for use by the editor (ZZZZGO, ZZZZG1).	Rename file and rerun.	SDRT
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 was 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 FAILURE.	The resource executive (RESEX) has detected a fatal error.	Write a PSR.	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.	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
RESOURCE NEGATIVE SHARE COUNT.	The resource overcommitment algorithm indicates a greater number of users are sharing a removable pack than are actually sharing the pack.	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
RESOURCE PF ERROR ec filename.	PFM error ec occurred when attaching resource file filename.	Rerun the job. If the problem reoccurs, inform a site analyst, write a PSR and include support material so that CDC can duplicate the problem.	RESEX
RESOURCE SCRATCH FILE ERROR.	An empty entry has been found in the overcommitment algorithm scratch file.	Rerun the job. If the problem reoccurs, inform a site analyst, write a PSR and include support material so that CDC can duplicate the problem.	RESEX
RESOURCE TYPE ERROR.	You specified an incorrect 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
RESUMING 7000 MODE.	HELLO7 is attempting to reestablish connections with a SCOPE 2 job as a result of a RESUME directive.	Wait for the HELLO7 prompt to appear. If the prompt does not appear, check to ensure that you have a SCOPE 2 job active.	HELLO7
RETRY APPLICATION SWITCH.	An attempt to switch applications failed due to software error.	Retry the APPSW command. If the error recurs, inform your Site analyst.	TLX
RFL BEYOND MFL.	The RFL request is greater than the maximum field length for a job step.	Increase maximum field length with MFL command or SETMFL macro.	CPM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
RHF, jobn APPLICATION IS NOT NETTED ON	The application with job name or job sequence name jobn requested an RHF function before requesting a NETON to RHF. RHF aborts the application.	None.	RHF
RHF, DUPLICATE NETON REQUEST.	Two NETON requests were made for the same application without an intervening NETOFF request. RHF aborts the application.	Remove the second NETON request or add the missing NETOFF request.	RHF
RHF, FATAL SSF ERROR. FC=fc, RC=rc.	System error.	Inform site analyst.	RHF
RHF, INVALID APPLICATION CALL TO RHF.	An application issued an invalid RHF call. The call may contain an incorrect RHF function, a request (other than NETON) from an application with an end-of-job connect status, or an incorrect word count in the RHF call. RHF aborts the application.	Correct error in application and retry.	RHF
RHF, INVALID APPLICATION NAME ON NETON.	An application issued a NETON request using an application name that was not in RHF's configuration or that contained incorrect characters. RHF aborts the application.	Correct name in the application NETON call or add the application name to RHF's configuration.	RHF
RHF, INVALID APPLICATION TABLE ADDRESS.	In an RHF call, an application used an incorrect application table address. The address may be out of range or may point to another application table. RHF aborts the application.	Correct error in application.	RHF
RHF, jsn INVALID FET PRAM. FET = address.	System error.	Inform site analyst.	RHF
RHF, jsn INVALID HEADER ADDRESS address	System error.	Inform site analyst.	RHF
RHF, INVALID MINACN/MAXACN ON NETON.	The value for the minimum or maximum ACN in the NETON request is outside the range specified for the application. RHF aborts the application.	Correct the minimum or maximum ACN in the application's NETON request.	RHF
RHF, INVALID SSF UCP ADDRESS address	System error.	Inform site analyst.	RHF
RHF, INVALID TEXT ADDRESS address.	System error.	Inform site analyst.	RHF
RHF, NETON SECURITY VIOLATION	An application is not validated to do a NETON request. RHF aborts the application.	None.	RHF
RHF, NO APPLICATION ADDRESS IN RHF CALL--EXTRA CHARGE	Informative message indicating an application issued an RHF request (other than NETON) without specifying an application table address in the RHF call. The application is charged less if it specifies its application table address in each RHF call.	None.	RHF
RHF, QUEUED MESSAGE LIMIT EXCEEDED	An application exceeded the maximum number of supervisory messages that are queued in RHF. RHF aborts the application.	Modify the application to issue more frequent NETGETs for the supervisory messages queued in RHF.	RHF
ROUTE COMMAND ERROR.	Format of the command is incorrect.	Check the ROUTE command format in section 9.	ROUTE
ROUTE COMPLETE.	A ROUTE command with DC=SC specified has successfully removed a file from a queue.	None.	ROUTE
ROUTE COMPLETE. JSN IS jsn.	A ROUTE command placed a job or file in a queue. The routed job or file has the job sequence name jsn.	None.	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.	ROUTE
ROUTE INCORRECT *DD* PARAMETER.	The DD=lid parameter on the ROUTE command has an incorrect format. lid Logical identifier of remote mainframe.	Correct the format of your DD=lid specification.	ROUTE
ROUTE INCORRECT *EC* PARAMETER.	You specified an incorrect external characteristics code on the ROUTE command.	Correctly specify the EC=ec parameter on the ROUTE command.	ROUTE
ROUTE INCORRECT *FC* PARAMETER.	The forms code specified on the command is not valid.	Check the command format and reenter command.	ROUTE
ROUTE INCORRECT *IC* PARAMETER.	You specified an incorrect internal characteristics code on the ROUTE command.	Correctly specify the IC=ic parameter on the ROUTE command.	ROUTE
ROUTE INCORRECT *JSN* PARAMETER.	You specified a JSN which was not exactly three characters long.	Specify a three-character JSN.	ROUTE
ROUTE INCORRECT KEYWORD.	Command contains a duplicate or incorrect keyword.	Check the ROUTE command format in section 9.	ROUTE
ROUTE INCORRECT *OT* PARAMETER.	A system table contains an incorrect value for the job's origin type.	Write a PSR and include support materials to allow CDC to duplicate the problem.	ROUTE
ROUTE INCORRECT 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

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
ROUTE INCORRECT *ST* PARAMETER.	The ST=lid parameter on the ROUTE command has an incorrect format. lid Logical identifier of remote mainframe.	Correct the format of your ST=lid specification.	ROUTE
ROUTE *JSN* NOT ALLOWED.	Your job is not system origin, and only system origin jobs can specify the JSN parameter.	Do not attempt to use the JSN parameter unless your job is system origin.	ROUTE
ROUTE *OT* NOT ALLOWED.	Your job is not system origin and only system origin jobs can use the OT parameter.	Do not attempt to use the OT parameter unless your job is system origin.	ROUTE
ROUTE *PRI* IGNORED.	Informative message; this parameter is allowed (but ignored) for NOS/BE compatibility.	None.	ROUTE
ROUTE *REP* GT 63. DEFAULT USED.	The repeat count specified was greater than 63; it has been set to 0. This condition does not abort the program.	Use a repeat count less than or equal to 63.	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 UNDEFINED SERVICE CLASS.	The specified service class was not recognized.	Correct and retry.	ROUTE
ROUTE UNKNOWN *PI* PARAMETER.	The ROUTE command contains an incorrect PI parameter.	Check with site personnel to determine the correct value to be specified.	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
RUN COMPLETE.	Completion message for jobs in all subsystems except batch.	None.	IAFEX
SC ONLY PARAMETER VALID WITH JSN.	You included an OT, L, or OP parameter along with a JSN parameter on a CLASS command. If you specify a JSN parameter, the only other valid parameter for CLASS is the SC parameter.	Correct the CLASS command parameters and resubmit.	CLASS
SCP INCORRECT TRANSFER ADDRESS.	The SCP SSCR (RA + 51B) word contains incorrect parameters or the completion address for an SF.REGR, SF.SWPI or SF.SWPO function is no longer within the SCP field length.	Correct code error in SCP.	1AJ
SECONDARY APPLICATION ABORTED CONNECTION.	You intentionally or unintentionally aborted the secondary connection. Possible causes are security conflict or entry of a BYE,ABDRT command.	If the abort was unintentional and you cannot establish a cause, see your site analyst.	TLX
SECONDARY APPLICATION CONNECTION LIMIT.	The application you specified on an APPSW command is busy with the maximum allowable number of terminal connections.	Try again later.	TLX
SECONDARY APPLICATION FAILED.	Your secondary application has failed, probably due to a system problem.	Check with your site analyst or try again later.	TLX
SECONDARY APPLICATION NETTED OFF.	The application was brought down by the network operator.	Check with your site analyst or try again later.	TLX
SECONDARY APPLICATION NOT AVAILABLE.	The secondary application requested on an APPSW command is either temporarily unavailable or is not supported at your site.	Check with your site analyst.	TLX
SECONDARY APPLICATION PROCESSING COMPLETE.	The connection to your secondary application has been terminated normally.	None.	TLX
SECONDARY APPLICATION REFUSED CONNECTION.	The application refused to accept your connection, possibly as a result of a validation problem or security conflict.	Check with your site analyst.	TLX
SECONDARY USER COMMANDS DISABLED.	You attempted to specify a USER command to a second user name, but your site has disabled this usage.	Log in again or submit a separate batch job, specifying the second user name.	VALEX
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 NOS Reference Set, Volume 3 or user Field Length Dump Request in Volume 4. To	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
SECURE PASSWORD ENTRY REQUIRED.	No parameters may be entered on the PASSWOR command if password randomization is required.	obtain memory dumps from an interactive job, include the dump command in a procedure or in an ENTER command. Retry PASSWOR without any parameters.	MODVAL
SECURITY CONFLICT.	An attempted operation within the job would have resulted in a violation of security access levels or categories. The cause is described in the immediately preceding dayfile message.	Correct and retry.	1AJ
SEE DAYFILE-UNABLE TO LOAD pfn.	RECLAIM is unable to load file pfn due to bad tape, database in error, the file is already permanent, or some other problem as indicated in the dayfile.	Check the dayfile for the specific error. Check your permanent file catalog to ensure that the specified file is not already in the catalog. If the file is not already permanent, check the database file and dump tape.	RECLAIM
SELECT BY NUMBER OR TYPE Q TO QUIT	Normal prompt for a menu procedure called in line mode.	Make the appropriate menu selection in response to the prompt(s).	CCL
SELECTION DOES NOT MATCH MENU LIST	On a line mode menu procedure call or response, the value entered does not represent a valid menu selection.	Make the appropriate menu selection in response to the prompt, or reissue the command with a valid selection.	CCL
SERVICE CLASS FULL.	You attempted to change a job to a service class for which the service class limit has already been reached.	Resubmit the CLASS command at a later time.	CLASS
SERVICE CLASS FULL.	The number of jobs in a particular service class has reached its limit.	Retry logging in at a later time or contact site administrator to increase the number of jobs allowed in particular service class.	IAFEX
SERVICE CLASS REQUIRED WITH JSN.	You entered a CLASS command with a JSN parameter but did not include an SC parameter. If the JSN parameter is specified, an SC parameter must also be included.	Correct the CLASS command parameters and resubmit.	CLASS
SERVICE LIMIT EXCEEDED.	Your job exceeded the field length limits for a job in its service class.	Reduce the job's field length requirements and resubmit.	1AJ
SFM ABORTED.	An error flag was set at SFM's control point.	Inform site analyst.	SFM
SFM FAST ATTACH SYSTEM SECTOR ERROR.	An unrecoverable error was detected when trying to read the system sector of the file being entered into fast attach status. The file was entered, but the system sector pointers were not set. This file may not be dumped properly by PFDUMP when this occurs.	Inform site analyst.	SFM
SFM FILE EMPTY.	The file specified was empty.	Verify that the file contains data and retry.	LFM SFM
SFM I/O SEQUENCE ERROR.	Action has been requested on a busy file.	Wait until file is not busy and retry.	SFM
SFP CALL ERRDR.	System error.	Write a PSR, including a dump or support materials to allow CDC to duplicate the problem.	SFP
SFP/DOO BAD ERROR TEXT.	The specified system text is not in the correct format.	Rebuild system text using correct format.	SFP
SFP/DOO ERROR TEXT NOT FOUND.	The specified system text could not be found.	Ensure the system text specified is correctly installed on the system. If it is, write a PSR.	SFP
SFP/DOO ERRDR TEXT NOT ON MASS STORAGE.	The specified system text does not reside on mass storage.	Move system text to mass storage.	SFP
SFP/DOO INVALID MESSAGE NUMBER.	The specified message number could not be found on the system text.	Contact CYBER Software Support.	SFP
SFP/xxx INCORRECT FUNCTION CODE.	The specified function code to the program xxx is not defined.	Ensure the correct function code was used.	SFP
SFP/xxx INCORRECT ORIGIN CODE.	Dayfile message indicating that the function was incorrect for the job's origin.	Write a PSR.	SFP

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
SFP/xxx PARAMETER ERROR.	Dayfile message indicating that the parameter address was outside the field length.	If the code is yours, find the source of the error and fix it. If the code is CDC's, write a PSR.	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/SRP SPECIAL REQUEST PROCESSING ERROR.	Dayfile message indicating that the SPCW word was busy.	Write a PSR, including a dump or support materials to allow CDC to duplicate the problem.	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.	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
JSN SHARE TABLE MISMATCH.	A system error occurred while processing the job with the specified job sequence name (jsn).	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
SINGLE FRAME ERROR.	A frame (NRZI only) containing all zeros was read; data will be at least one frame short. The dayfile message STATUS ERROR, filename AT address precedes this message, and specifies the file (filename) and the address.	Retry or inform site analyst.	1MT
SL NOT VALIDATED.	The SRU limit requested is zero.	Request a non-zero 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
SPECIFY EXACTLY 3 CHARACTERS ON LID KEYWORD.	The value specified for the LID keyword or the ST keyword must be exactly three characters long.	Correct and retry.	LISTLID
SPECIFY EXACTLY 3 CHARACTERS ON PID KEYWORD.	The value specified for the PID keyword must be exactly three characters long.	Correct and retry.	LISTLID
SRU LIMIT ENTER S TO CONTINUE OR STOP TO END JOB STEP. TYPED AHEAD INPUT MUST BE REENTERED.	You have exceeded your system resource unit (SRU) limit during a terminal session.	To continue, allocate nnnnn additional SRUs with S,nnnnn command or enter S alone to allocate an installation defined number of SRUs.	IAFEX
STACK PURGING NOT DESELECTABLE.	You attempted to change the purging bit with the EREXIT function. This cannot be done on 8000 or CYBER 70 systems.	None.	CPM
pfN STAGE INITIATED.	The file does not reside on disk. It is currently being staged from alternate storage to disk.	Wait until stage operation is complete.	PFM
STATION DOWN	The SCOPE 2 station is not currently executing.	Try to reenter the command at a later time.	QUEUE7
STATUS ERROR, filename AT address.	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. JSN IS jobseq	The SUBMIT command successfully queued the file for input. The job sequence name (JSN) for the job is jobseq.	None.	SUBMIT
SUBMIT ERROR - RETRY	An internal error occurred during SUBMIT processing.	Reenter the SUBMIT directive. If the error recurs, inform the site analyst.	HELLO7
SUBMIT FILE EMPTY.	An EDR or EOF was encountered on the submit file before any data was found.	Rewind submit file, verify the format of the data and that the file is a local file, and retry operation.	SUBMIT
SUBMIT NOT CALLED FROM HELLO7	A SCOPE 2 job requiring terminal I/O was not submitted through HELLO7.	Resubmit the job through HELLO7.	HELLO7
SUBSYSTEM ABORTED.	Your job was connected (either long term connection or wait response set) to a subsystem which aborted.	Retry later.	1AJ

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
SYMBOL TABLE FULL.	You cannot define any more symbols because the symbol table is full.	Undefine one or more symbols to make room for the new one.	DEBUG
SYSTEM ABORT.	Possible causes include the following: <ul style="list-style-type: none"> - The job contained an incorrect USER command. - The job attempted to access a restricted subsystem. - Operator evicted job. - Unrecognizable error flag appeared. - SSJ: block outside field length. - IRI detected a bad rollout file. - IRD detected an unrecoverable extended memory parity error during rollout of a job's extended memory field length. 	If the cause was an incorrect USER command or an attempt to access a restricted subsystem, correct the job and rerun. Otherwise, inform a site analyst.	1AJ
SYSTEM CHECKPOINT ABORT.	A subsystem has aborted due to a CHECK POINT SYSTEM request initiated by the operator.	None.	1CK
SYSTEM ERROR.	A software or hardware system error occurred. This message follows a more specific message in the dayfile.	Refer to action for the associated message.	PFM MODVAL
SYSTEM ERROR.	The command is not recognized by the system.	Retry command.	IAFEX
SYSTEM ERROR.	LFM cannot complete the requested LFM function because the calling program has a DMP: entry point.	Inform site analyst or write a PSR and include support material so that CDC can duplicate the problem.	BLANK RESEX
SYSTEM ERROR.	Requested function cannot be completed because the calling program has a DMP: entry point.	Contact CYBER Software Support.	LFM
SYSTEM ERROR.	The system terminated the job due to an internal error.	Log in again. Inform site analyst or write a PSR and include support material so that CDC can duplicate the problem.	IAFEX
SYSTEM ERROR.	Your job was aborted because of an operating system software error.	Contact site personnel and write a PSR.	1AJ
SYSTEM ERROR - NOTIFY SITE ANALYST.	An internal software error has occurred.	Write a PSR and include support materials to allow CDC to duplicate the problem.	DRDP QGET
SYSTEM FULL.	The queued file table (QFT) is full; no new jobs can log in.	Wait a few minutes and retry login. If the problem persists, inform site analyst.	IAFEX
SYSTEM PROLOGUE NOT FOUND.	The system prologue file defined for your user name could not be found.	Contact site personnel.	VALEX
SYSTEM SECTOR ERROR.	The system sector of an indirect access permanent file contains an error (error log and dayfile message). This indicates that the file has been destroyed.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, total INITIALIZE, and full PFLoad on the device.	PFM
SYSTEM SOFTWARE RERUN THRESHOLD EXCEEDED.	The job has been rerun the maximum number of times due to a software error. The maximum is n-1, where the default for n is 2 (defined by MSER in common deck CDMSMSC).	Analyze the job output, dumps, and listings to determine the cause of the error.	1AJ
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 a 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 CLEANER FAULT, filename AT address.	This error message appears in the dayfile if one of the following has occurred: <ul style="list-style-type: none"> - Attempts to recover from the CLEANER FAULT message in the E,P display failed, and TERMINATE,est. was entered. - The tape is not assigned to a job, and the tape cleaner failed to return to the parked or stowed position after a tape load operation or a high speed rewind operation. MAGNET will unload the tape and turn off the drive. 	Suspend use of the tape drive and notify a customer engineer of the problem.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
TAPE FORMAT PROBABLY WRONG.	This message is issued in addition to one of the following messages: BLOCK SEQUENCE ERROR, filename AT addr. BLOCK TOO LARGE, filename AT addr.	Ensure accuracy of format (F) parameter and label (LB) parameter on command or macro. If the problem still exists, clean the tape drive or use a new tape.	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
TAPE NUMBER FILE EMPTY.	Error in RECLAIM scratch file.	Inform site analyst.	RECLAIM
TAPE READ/WRITE ERROR (BLANK)	The BLANK label operation was unable to complete successfully.	Retry with the same tape or use a different tape.	BLANK
TCS INCORRECT REQUEST.	The translate commands (TCS) program was called with an incorrect function code.	Correct the call to TCS and retry.	1AJ
pfm TEMPORARY ERROR, TRY LATER.	The system has detected a temporary problem which prevents it from staging your file (pfm) from permanent storage.	Wait a few minutes and try to attach the file again.	PFM
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 DISABLED BY NOP.	The network operator has disabled your terminal. No further input is possible.	None.	CCP
TERMINAL INTERRUPT.	RECLAIM processing was terminated due to a terminal user break.	None.	RECLAIM
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 in the installation defined timeout period.	Attempt recovery or log in again.	IAFEX
TERMINAL TIMEDOUT IN nn SECONDS.	No input was received for several minutes. The system will log you out after nn seconds if no input is entered.	Enter a carriage return to prevent logout.	IAFEX
THE STATION IS NOT LOGGED INTO SCOPE 2	The SCOPE 2 station is not active.	Reenter the command later when the station is executing.	QUEUE7
THIS CONNECTION IS ALREADY LOANED.	This error indicates an abnormal condition is occurring in IAF.	Inform site analyst.	TLX
THIS WOULD DESTROY DEBUGGER.	You tried to change a memory location within the debugger.	Correct address and retry.	DEBUG
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 you are 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 STOP TO END JOB STEP. TYPED AHEAD INPUT MUST BE REENTERED.	The system detected that the time limit for the job step has expired.	To continue, reset time limit with a T,nnnnn command or enter T alone to set time limit to an installation defined default.	IAFEX
TIME LIMIT.	The job has reached its CPU time limit.	Increase your job's time limit.	RECLAIM
TIMEOUT.	The user has 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.	NVF
TL NOT VALIDATED.	The time limit requested is zero.	Request a non-zero time limit.	CPM
TLX - RPV ALREADY SET.	Your program tried to set extended reprieve when it was already set.	None.	TLX
TN OR DN MUST BE SPECIFIED.	A RECLAIM operation was attempted which required a dump file name or tape number, but neither was specified.	Retry after specifying the file name or tape number.	RECLAIM
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 1AJ
TOO MANY FILENAMES IN LIST.	The number of filenames listed after a PF** parameter exceeded 999. The directive was ignored.	Reduce the number of files specified and retry.	RECLAIM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
TOD MANY FILES	You have attempted to route a second file to SCOPE 2 when you already have a file routed to SCOPE 2.	Wait until the first job completes before submitting another job.	HELLO7
TOD 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
TOD MANY LOCAL FILES.	A RECLAIM attempt to create a local file exceeded your site-determined limit on the allowable number of local files. [RECLAIM uses six local files as internal scratch files.]	Reduce the number of files to your job or reduce the number of files to be processed by RECLAIM.	RECLAIM
TOD MANY PARAMETERS.	More parameters were entered (including null parameters) than are allowed for the command.	Ensure accuracy of entry.	GTR IAFEX NOTE
TOD MANY PERMANENT FILES.	The number of files in your catalog exceeds your validation limit.	Purge one or more permanent files to allow you to save or define additional files.	PFM
TOD 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
TOD MANY 7000 JOBS...SUBMIT LATER	The SCOPE 2 limit for the number of concurrent connected jobs has been reached. The system will continue to reject the SUBMIT directive until the connected job count is reduced.	Retry later.	HELLO7
TOD MUCH INDIRECT ACCESS FILE SPACE.	The cumulative size of the indirect access files in your catalog exceeds your validation limit.	Purge or shorten one or more indirect access files to allow additional permanent file space.	PFM
TOD SMALL BUFFER SPACE.	Internal error.	Submit a PSR with information on how to duplicate the problem.	IEDIT
TOTAL ASSIGNED COUNT ERROR.	System error occurred (sum of individual resource assigned counts differs from total assigned count in demand file entry).	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
TOTAL DEMAND COUNT ERROR.	System error occurred (sum of individual resource demand counts differs from total demand count in demand file entry).	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
TRACK LIMIT.	No allocatable tracks remain on your permanent file equipment (error log and dayfile message).	None; job will continue as tracks become available. If problem persists, contact CYBER Software Support.	PFM
TRACK LIMIT, LVLX.	IMS is waiting for temporary file mass storage space on any MASS storage device with access level LVLX.	Inform site analyst.	IMS
TRACK NOT RESERVED.	There is no track reserved for the file. The data for the file has been lost.	Inform site analyst. If many of these errors occur, the site analyst should perform a full PFDUMP, deadstart INITIALIZE and full PFLDUMP on the device.	PFM
TRAILER BLOCK COUNT ERROR, filename AT address.	The block count in the EOF1 or EDV1 label did not match the block count maintained by the tape executive during the read operation.	Inform site analyst.	1MT
TRMDEF COMPLETE.	The TRMDEF command was processed with no errors.	None.	TRMDEF
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 REQUIRES JOB TO BE ON-LINE	A TRMDEF command was issued while the job was suspended. This caused the job to be suspended. When the job is recovered, this message is issued and the command processed.	None.	TRMDEF
TRUNCATED - REQUESTED LIST TOO LONG.	The entry point list exceeds the length of the ITEMIZE internal buffer.	Shorten list.	ITEMIZE

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
T76 RETURNED UNKNOWN ERROR OF nn	A system error has occurred between NOS and the SCOPE 2 station. nn is a two-digit error code meaningful to the site analyst.	Reenter the command. If the same message is returned, inform the site analyst.	QUEUE7
UEAD, 123456.789KUNS.	Informative message issued to job dayfile at job termination, listing the final value of the SRU adder accumulator for the job (in units of one thousand).	None.	1AJ
UECP, 123456.789SECS.	Informative message issued to job dayfile at job termination, listing the number of CPU seconds used by the job.	None.	1AJ
UEMS, 123456.789KUNS.	Informative message issued to job dayfile at job termination, listing the number of mass storage PRUs read and/or written by the job (in thousands of PRUs).	None.	1AJ
UEMT, 123456.789KUNS.	Informative message issued to job dayfile at job termination, listing the number of magnetic tape PRUs read and/or written by the job (in thousands of PRUs).	None.	1AJ
UEPF, 123456.789KUNS.	Informative message issued to job dayfile at job termination, listing the amount of permanent file activity performed by the job (in units of one thousand).	None.	1AJ
UNABLE TO DROP JOB.	You attempted to drop the job associated with your terminal. It is not possible.	If you want to drop another one of your jobs, use ENQUIRE,JSN to ascertain their job names.	DROP
UNABLE TO PROCESS TERMINAL FILE.	The file you specified in a SECHDR command is currently assigned to a terminal.	Assign the file to an appropriate device and reenter the command.	SECHDR
UNABLE TO RECOVER YOUR JOB	The SCOPE 2 job was either in execution and could not be rerun, or, due to certain file functions performed prior to the recovery mode, was not recoverable.	None.	HELLO7
UNDEFINED ORIGIN TYPE.	Origin type mnemonic specified is undefined.	Correct origin type and retry.	CLASS
UNDEFINED SERVICE CLASS.	You specified an undefined service class value for the SC parameter on the CLASS command or the UC parameter on the CHVAL command.	Correct the SC or UC parameter value and resubmit.	CLASS
UNDEFINED SERVICE CLASS.	A batch job was entered from a card reader or routed from another mainframe with an undefined service class specified on the job command.	Correct and retry.	1AJ
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, filename AT address precedes this message and specifies the file (filename) and the address.	Resubmit your job, using a different tape unit or inform site analyst.	1MT
UNIT HUNG UP ON EOP OR BUSY, filename AT address.	Unit did not receive EOP on unit busy.	Inform site analyst.	1MT
UNIT PROBLEMS.	Unit check bit is set in detailed status (87x units only). The dayfile message STATUS ERROR, filename AT address precedes this message and specifies the file (filename) and the address.	Try another unit or inform site analyst.	1MT
UNKNOWN ACCESS CATEGORY NAME.	You specified an undefined access category name on a SETPFAC command.	Reenter the command with the correct access category name.	MLSEEXEC
UNKNOWN ACCESS LEVEL NAME.	The job specified an undefined access level name.	Reenter the command with the correct access level name.	MLSEEXEC MSI PFILES PFS RESEX
UNKNOWN CODE cccc, FILE filename, RECORD recordnumber, LINE linenumber.	Character code cccc is not a valid character code in the character code set specified by PC=csl in the FCOPY command.	Correct file and retry.	FCOPY
	filename File that contains the code recordnumber Record that contains the code linenumber Line that contains the code		
UNKNOWN DEVICE TYPE - filename.	File filename 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
UNKNOWN DUMP FILE WILL BE OVERWRITTEN.	When preparing to do an incremental (EI-type) dump, RECLAIM determined that the specified dump file does not contain valid dump information. The file is either empty or contains information other than RECLAIM dump information. If the user is executing interactively, RECLAIM will also issue the prompt " IS THIS OK (YES OR NO)?", and will wait for a response.	If interactive, respond by entering Y to overwrite the file or N to abort the operation.	RECLAIM

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
UNKNOWH HOLL. CODE, RECxxx CDyyyy.	Command yyyy in record xxx was found to contain an incorrect Hollerith code. The job was terminated without exit processing.	Correct command in error.	1AJ
UNKNOWH OPTION - x.	You specified an undefined option in a SECHDR command. x Undefined option specified	Reenter the command with correct option.	SECHDR
UNKNOWH SECONDARY APPLICATION.	On an APPSW command, you specified a secondary application that does not exist.	Correctly specify a secondary application.	TLX
UNLABELED TAPE REQUIRED - filename.	An S tape used for E, B, or X tape conversion must be unlabeled.	Use unlabeled tape.	COPYB
UNNECESSARY CIO FCT. nnn ON filename AT address.	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.	IMS
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 COPYB command description, correct the error, and retry.	COPYB
UNRECOGNIZED CODE SET PARAMETER.	Parameters PC=cs1 and NC=cs2 specify an unsupported conversion.	Verify conversion and retry.	FCOPY
pppppp UNRECOGNIZED PARAMETER	The parameter name "pppppp", which was entered as a keyword parameter, is not defined as a parameter in the procedure being called.	Check the parameters for the procedure being used, correct and reissue the command.	CCL
UNRECOGNIZED TERMINATION CONDITION.	The specified termination record count or record type was not recognized.	Refer to the COPYB command description, correct the error, and retry.	COPYB
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.	COPYB
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 ATTRIBUTE - xx.	The value of the attribute is recognized but not supported by TRMDEF. xx Attribute mnemonic	None.	TRMDEF
UNTERMINATED PROCEDURE CALL.	A BEGIN or name call to an interactive procedure did not have a terminator. One or more parameters may be incorrect or omitted.	Check parameters displayed on screen and verify or correct before proceeding.	CCL
UNUSUAL END-OF-FILE ENCOUNTERED.	GTR detected an EOF mark not preceded by an EFR mark.	Dump file to determine file format error.	GTR
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
UPROC COMPLETE.	The prolog you specified in the UPROC command has been successfully assigned to your user name.	None.	MODVAL
**** 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 a *FILINFO* call to obtain the desired data.	SFP
USE SCOPY FOR FILE STRUCTURE.	You entered the 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 ACCESS NOT POSSIBLE-CONTACT NETWORK ADMIN.	Either the application rejected the connection, the Security count was exhausted, or an invalid application name was specified in the local configuration file (LCF).	Contact the network administrator.	NVF
USER ACCESS NOT VALID.	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 authorized to access.	Ensure accuracy of command or determine proper validation requirements via LIMITS command.	LFM
USER ACCESS NOT VALID.	You tried to perform an operation for which you are not authorized. Possible causes include attempts to do the following: - Run a system origin job from non-system origin. - Access a restricted subsystem without proper validation. - Use the V carriage control character without validation.	Ensure accuracy of command or macro, or determine proper validation requirements via LIMITS command.	IAFEX MSI QFSP RESEX 1MA

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
USER ACCESS NOT VALID.	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
USER ACCESS NOT VALID.	You attempted to use RECOVER in a noninteractive job.	Initiate an interactive job to use RECOVER.	RECOVER
USER ACCESS NOT VALID.	You made an unauthorized call to a utility reserved for system origin privileged users.	Ensure that you have system origin privileges before using this utility.	FOTD SUBSYST
USER ACCESS NOT VALID.	The SRU or time limit request is outside of your validated range.	Retry with a lower value.	CONTRDL
USER ACCESS NOT VALID.	One of the following occurred: - You are not authorized to create direct access or indirect access files. - You are not authorized to access auxiliary devices. - Your job made a CATLIST request for permit information on a file that has a higher access level than that of your job.	Contact your site administration about obtaining the necessary validations. For CATLIST request, resubmit the request after raising your job's access level.	PFM
USER BREAK TWO ENCOUNTERED.	Informative message.	None.	1AJ
USER CANNOT SWITCH FROM A SECONDARY APPLICATION.	You attempted to enter an APPSW command from a secondary application.	To return to your primary application, log out of your secondary application. If you wish to switch to a third application, you can enter another APPSW command from your primary application.	TLX
USER CANNOT SWITCH TO CURRENT APPLICATION.	You tried to switch to the application you are already using.	Try a different application.	TLX
USER COMMAND HAS NOT BEEN EXECUTED.	A SUBMIT cUSER directive could not be processed because your job had not processed a USER command before calling SUBMIT.	Enter a valid USER command, then retry SUBMIT.	SUBMIT
USER DATABASE MISSING.	The specified database file was not found and could not be constructed.	None.	RECLAIM
USER DAYFILE PROCESSED.	The job's dayfile dump is complete.	None.	DAYFILE
USER DIRECTIVE INCORRECT.	Either your job has not yet processed a USER command, or the cUSER directive is positioned incorrectly within the file. The SUBMIT cUSER directive must be the second line in the file (not counting other SUBMIT directives).	Enter a valid USER command, if you have not yet done so. Ensure that your file is formatted correctly.	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 NAME/PASSWORD NOT VALID.	A batch job was entered from a card reader or routed from another mainframe with an incorrect user name or password.	Correct and retry.	1AJ
USER NOT VALIDATED FOR DETACHED SERVICE CLASS.	User is not validated for detached service class (DI).	Have site administrator validate user for detached service class if appropriate.	IAFEX
USER NOT VALIDATED FOR INTERACTIVE SERVICE CLASS.	User is not validated for interactive service class (TS).	Have site administrator validate user for interactive service class if appropriate.	IAFEX
USER NOT VALIDATED FOR SECONDARY APPLICATION.	You submitted an APPSW command specifying an application for which you are not validated.	To check your current validations, enter a LIMITS command. If you require validation for additional applications, see your site administrator.	TLX
USER NOT VALIDATED TO SPECIFY *ST*.	A batch job was entered from a card reader or routed from another mainframe with an ST parameter on the job command, but the specified user name was not validated to specify the ST parameter.	If you need to be able to specify the ST parameter, contact site personnel concerning your validations.	1AJ
USER PROLOGUE FILE NOT LOCAL.	The local file ZZZZPR2 was returned prior to the time that it was needed.	Contact site personnel.	VALEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
USER PROLOGUE NOT FOUND.	Although a user prologue was defined, it could not be accessed.	The user should either delete the prologue definition or create a prologue file.	VALEX
USER PROLOGUE NOT FOUND.	The prologue file defined for your user name could not be found.	Recreate your user prologue file.	VALEX
USER RETRY LIMIT.	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
USER SECURITY COUNT EXHAUSTED.	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
USER SECURITY COUNT EXHAUSTED.	A batch job was entered from a card reader or routed from another mainframe which could not execute because the user security count for the specified user name had been exhausted.	Contact site personnel concerning your user security count.	IAJ
USER VALIDATION ERROR.	You are not validated for this subsystem.	Try a different subsystem or contact your site administrator to get validated for the desired subsystem.	IAFEX
USER VALIDATION LIMIT EXCEEDED.	A validation limit has been reached for a job step or account block. The limit is either a time limit or an SRU limit.	If appropriate, have site administrator increase validation limits.	IAFEX
VALIDATION FILE BUSY - TRY AGAIN LATER	The validation routine (MODVAL) was unable to access the validation file to perform the requested UPROC, PASSWOR, or CHVAL operation.	Try again later.	MODVAL
cmd VALUE INAPPROPRIATE	The value you specified in the terminal definition command cmd is not appropriate for your terminal.	Choose an appropriate value (refer to appendix J).	CCP
VE ENTRY REQUIRED	A level 0 deadstart is required to add or change a CMRDECK VE entry.	Level 0 deadstart is required.	REC
VERIFICATION ERROR. PASSWORD NOT CHANGED.	Password was entered incorrectly when verification of the new password was requested.	Retry PASSWOR with the correct password.	MODVAL
VERIFY ARGUMENT ERROR.	One or more of the parameters specified in a VERIFY was not recognized.	Compare the parameters specified with the command description.	VERIFY
VERIFY ERRORS.	Verification errors were encountered during comparison of date/records/files.	Check output file for errors.	VERIFY
VERIFY FILE NAME CONFLICT - filename.	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.	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
VERIFY GOOD.	Verify completed with no errors found.	None.	VERIFY
VSN FILE ERROR.	System error (VSN file entry does not match job identification).	Reenter the command or rerun the job. If the problem reoccurs, inform your site analyst or contact CYBER Software Support.	RESEX
WAIT	The system is waiting for system resources to become available.	None.	IAFEX
WAIT..	The network is unable to accept any further data from your terminal. Data that was entered is discarded. This is normally a temporary situation caused by buffer shortages in the host or the network.	Wait for the REPEAT.. message.	CCP
pfn WAITING ALTERNATE STORAGE EXEC.	The file must be staged from alternate storage to disk and the alternate storage executive is not currently available to perform the stage operation.	Wait for executive to be started or inform site operator that the alternate storage executive should be initialized.	PFM
WAITING FOR DATABASE NON-BUSY.	The specified database is busy (for example, it may be attached in write mode to another job). RECLAIM continues to attempt to attach the file at ten second intervals until the file becomes available or you interrupt RECLAIM.	Wait for the specified database to become available or terminate RECLAIM processing.	RECLAIM
WAITING FOR DOWN TAPE UNIT.	The job is waiting for a DOWN tape unit to be turned DN.	Wait for the operator to turn on the tape unit or terminate the job.	RESEX

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
WAITING FOR MAGNET.	The job is waiting for the magnetic tape subsystem to be activated.	Wait for the operator to activate MAGNET or terminate the job.	RESEX
WAITING FOR NFL.	Informative message stating that your job has been delayed and/or rolled out waiting for NFL space. The job has not been aborted; it will eventually continue. The message is visible via ENQUIRE.	None.	PFM
WAITING FOR PN=packname, type.	The job is waiting for the operator to mount pack packname on device type type.	Wait until the operator mounts the requested pack or terminate the 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 the 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 previous data in the demand file 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 the job.	RESEX
WAITING FOR SERVICE CLASS TO CHANGE TO x.	Service class x has reached its service limit and a job is waiting for a place in it.	None.	CLASS
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 the job.	1MA
WAITING FOR STORAGE (EM).	The job is waiting for additional extended memory to be made available.	Wait for additional storage or terminate the job.	1MA
WAITING FOR VALIDATION FILE.	The job is waiting for the validation file to become available to process a UPROC, PASSWOR or CHVAL command.	Try again later.	MODVAL
WAITING FOR VSN= vsn, type.	The job is waiting for the operator to mount the tape with VSN vsn on the specified type (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 - INACCESSIBLE DEVICE.	Your job has been rolled out waiting for an inaccessible mass storage device. The device is off, down, or is encountering hardware errors.	Inform site analyst.	1R0
WAITING - INACCESSIBLE DEVICE.	Access to your permanent file is not possible at this time because a permanent file device has SUSPECT, DFF or DOWN status.	Inform site operators. Retry request when device is accessible.	PFM
WAITING - INACCESSIBLE DEVICE.	Self-explanatory.	Detach the job for later processing or abort the job step.	1TA
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 the job.	RESEX
WAITING - SERVICE LIMIT.	The job has reached a service limit for central memory or extended memory field length.	Detach the job for later processing or abort the job step.	1TA
WRITE-DOWN OF DATA PROHIBITED.	Message resulted from one of the following: <ul style="list-style-type: none"> - On an APPEND or REPLACE command or macro, the access level of the local file is higher than the access level of the permanent file. - On a DEFINE command or macro, the access level of a local file having the same name as the file specified has a lower access level than the job. - On a CHANGE, PERMIT, or SETPFAC command or macro, the access level of the specified permanent file is lower than that of the job. 	Retry using a higher access level for the permanent file (APPEND or REPLACE), or change the job's access level and then retry (DEFINE, CHANGE, PERMIT, or SETPFAC).	PFM
WRITE ID BURST FAILURE, filename AT address.	The tape driver is unable to correctly write the PE (1600 cpi) or GE (6250 cpi) identification tracks or the automatic read amplitude (ARA) burst at load point.	Load the tape on another drive or use a different tape.	1MT

<u>MESSAGE</u>	<u>SIGNIFICANCE</u>	<u>ACTION</u>	<u>ROUTINE</u>
WRITE ON READ-ONLY FILE.	Either you attempted to write on a file with write lock-out or the direct access file was not attached in write mode.	Reattach file in write mode or clear write interlock.	1MS
WRITE ON READ-ONLY FILE.	Either you attempted to write on a file with write lockout, or the direct access file was not attached in write mode.	Reattach the file in write mode or clear write interlock.	IAFEX
WRITE ON READ-ONLY FILE filename AT address.	Either you attempted to write on a file with write lock-out or the direct access file was not attached in write mode.	Reattach file in write mode or clear write interlock.	1MS
WRITE ON READ-ONLY PRIMARY FILE.	Either you attempted to write on a file with write lock-out or the direct access file was not attached in write mode.	Reattach file in write mode or clear write interlock.	1MS
WRITE ON READ-ONLY PRIMARY FILE.	Either you attempted to write on a file with write lockout, or the direct access file was not attached in write mode.	Reattach the file in write mode or clear write interlock.	IAFEX
WRITE OVER LABEL INCORRECT.	You attempted to write over the VOL1 label.	Have the operator blank label the tape.	1MT
WRITE TAPE MARK FAILURE, filename AT address.	An error occurred while trying to write a tape mark and read it back.	Clean the tape drive or use a cleaned or new tape. If it still fails, contact a customer engineer.	1MT
WRITE VERIFY FAILURE, filename AT address.	The tape was positioned properly, but the current block cannot be read due to a triple track error.	Clean the tape drive or use a cleaned or new tape.	1MT
WRONG PARITY, filename AT address.	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
XD/XT EXCEEDS MAXIMUM.	The XD parameter for a password or file permission is more than the maximum number of days past the current date, or the XT parameter is larger than the maximum allowed.	Correct XD or XT parameter and retry.	PFM
XL BUFFER/FET PARAMETER ERROR, filename AT address.	Message resulted from 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
YOUR PASSWORD EXPIRES AT hh.mm.ss ON yy/mm/dd.	You should change your password prior to this date. If you do not change your password prior to this date, you will not be able to login.	Contact site personnel if you are not familiar with your site policy regarding passwords.	VALEX
ZZZZLD NOT ON MASS STORAGE.	The directory file ZZZZLD does not reside on mass storage.	Assign the directory file to mass storage.	1AJ
200 BPI WRITE INCORRECT.	The tape unit (667 or 677) cannot record data at 200 bpi.	Specify a different tape density.	1MT
7000 JOB EARLY ABORT	A submitted job aborted before it was able to connect to the terminal.	Check the job's dayfile for an error indication.	HELL07
7000 JOB ERROR - ERROR=xxx	The SCOPE 2 connected job encountered error condition xxx, as described in the Record Manager section of the SCOPE 2 Diagnostic Handbook.	Refer to the SCOPE 2 Diagnostic Handbook.	HELL07
7000 JOB EXITING	Indicates normal completion of a SCOPE 2 job step.	None.	HELL07
7000 NOT AVAILABLE	The SCOPE 2 mainframe is not available.	Wait until the SCOPE 2 mainframe is available.	HELL07
7000 SYSTEM ERROR	A system error has occurred on SCOPE 2.	Wait for SCOPE 2 to recover.	HELL07

GLOSSARY

C

Abort

To terminate a program, job or job step when an error condition (hardware or software) exists from which the program or computer cannot recover.

Access Category

Refer to File Access Category, Job Access Category Set, and System Access Category Set.

Access Level

A property of each file, job, and equipment on a secured system that is used to indicate the sensitivity of information in the file or job, or the sensitivity of information that can be processed by the equipment. On a secured system, there are up to eight access levels corresponding to increasing levels of sensitivity. You are authorized to access some or all of those levels. Refer also to Equipment Access Levels, File Access Level, Job Access Level, and System Access Levels.

Access Level Limits

Refer to Job Access Level Limits.

Account Block

That portion of a session from login to logoff or entry of a CHARGE command which signals a new account block.

Allocatable Device

A storage device allocated by the system without operator intervention that can be shared by more than one job.

Alphanumeric

Consisting of alphabetic or numeric characters only.

Application

A program resident in a host computer that provides an information storage, retrieval, or processing service to a remote user via the data communication 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. 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.

bpi

An abbreviation for bits per inch. Refer to fci.

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 to a magnetic tape or mass storage file a copy of your job'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.

Control Statement Record

Refer to Command Record.

cpi

Characters per inch. A measure of density.

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, Sort/Merge 5, 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, with some exceptions, when commands are processed. A copy of the dayfile is printed with the output for each 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 the your job. All changes to this file are made on the file itself rather than a temporary copy of the file (compare 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.

ECS

Extended Core Storage. Refer to Extended Memory.

Empty PRU/Record

A PRU that contains no user data. Refer also to 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, a card with a 6/7/8/9 multiple punch in column one. On mass storage devices, 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 (EOT)

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.

Epilogue

A program that executes at the end of your job or, in the case of a project epilogue, at the end of an account block.

Equipment Access Levels

A range of access levels specified for each equipment on a secured system. In order for a file to be stored or output on a given equipment, the file's access level must be within the equipment access levels for the defined equipment.

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 (EST) 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 Core Storage (ECS)

Refer to Extended Memory.

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 memory extended (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 via the alternate family.

Family Name

The name that is associated with a group of permanent file devices.

fci

Flux changes per inch. A measure of density used with magnetic tapes. Each flux change records two bits of information (per track); 3200 fci encodes 6400 bits of information. If you subtract 150 bits for error checking, 3200 fci is equal to 6250 cpi (assuming 1 bpi per track translates into 1 cpi per frame).

FET

Refer to File Environment Table.

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 Access Category

A property of a permanent file used by the creator of the file on a secured system to restrict access of the file to a particular group of users. A secured system supports up to 32 access categories, and you are authorized to use some, all, or none of those categories. Refer also to Job Access Category Set and System Access Category Set.

File Access Level

A property of each file on a secured system used to indicate the sensitivity of information contained on the file. A file is assigned the current job access level by default when it is created or stored; the file creator may specify any access level for that file that is within the set of access levels valid for the job, the system, the file creator, and (for interactive jobs) the communication line to the host mainframe. If you access a file on a secured system, you must be validated for the access level of the file. Refer also to Access Level, Job Access Level, and Job Access Level Limits.

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 or macro. 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 File Information Table.

Flag

A character or bit that signals the occurrence or presence of a particular condition.

FMT

Refer to Local File Name Table.

FORTRAN

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 one 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.

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.

Host

A computer that executes an application.

IAF

Refer to Interactive Facility.

Indirect Access File

A NOS permanent file that you access by making a temporary copy of the file (GET or OLD 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 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.

Interruption Sequence

The character or sequence of characters that causes an executing program to be interrupted (also called the user break 1).

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 Access Category Set

On a secured system, a set of access categories is set when the job is initiated. This set is the intersection of your set of validated access categories and the system access category set. Refer also to File Access Category and System Access Category Set.

Job Access Level

On a secured system, each job has an access level. This is the default access level that is assigned to files that are created or stored in the job. A job's initial access level is the lower access level limit for the job. The job's access level is automatically raised to the access level of any file from which information is read. You can also change the job access level. Refer also to Job Access Level Limits.

Job Access Level Limits

An upper limit and a lower limit that determine the range of access levels that are valid for a particular job on a secured system. All files used in a given job must have an access level within the job's access level limits.

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, procedure or loader sequence. A group of job steps forms a job stream, command record, or procedure file.

Label

A block at the beginning or end of a magnetic tape volume or file, which serves to identify 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.

LID

Refer to Logical Identifier.

Line

A unit of data terminated by a zero-byte terminator. Unit used in interactive input/output, line printer output, and card reader input.

Line Mode

A mode of job interaction in which the basic unit of interaction is a line of data. Contrast with Screen Mode.

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 local file name table.

Local File Name Table (FNT)

A system-managed table that contains the local file name, the file type, and other job control information.

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 Identifier (LID)

A three-character, alphanumeric string used to identify a particular mainframe in a loosely coupled network. LIDs are identified by your site.

Logical Record

A data grouping that consists of zero or more PRUs and ends with a short PRU or a zero-length PRU.

Login

The procedure used at an interactive terminal to gain access to the system.

Logout

The procedure used to end a terminal session.

Loosely Coupled Network (LCN)

A network of physically connected computer systems. The LCN environment allows jobs, data files, and messages to be transmitted from one computer system to another.

Macro

A sequence of source statements that are saved and then assembled whenever needed through a macro call.

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, applications, and mainframes.

Network Operator (NOP)

The network operator manages the communications elements of the network. Network operator functions may be performed at the system console, or at a remote terminal.

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.

Panel

A formatted screen defined using the panel definition utility. An application program uses a panel to display data or to request user input at the terminal.

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 controls access to a particular file by alternate users.

Peripheral Processor (PP)

The hardware unit within the host computer that performs physical input and output through the computer's data channels.

Permanent File

A mass storage file that 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 when they are created.

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 your 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.

Project Epilogue

An epilogue that executes any time you end an account block.

Project Prologue

A prologue that executes any time you initiate a new account block.

Prologue

A program the system executes any time you initiate a job (user prologue and system prologue) or initiate a new account block (project 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 physical record units (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.

Queue Priority

A number that controls the scheduling of a job to and from the rollout queue relative to other jobs of the same origin type.

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, 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. Compare 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 the user program and the system. RAE is the absolute extended memory starting address assigned to a program.

Remote Host

A computer connected to your local host so as to permit permanent file and queue file transfers between hosts.

Remote Host Facility (RHF)

An application program that allows you to transfer files to and from other mainframes connected to a loosely coupled network (LCN).

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.

Screen Mode

A mode of job interaction in which the basic unit of interaction is a screen of data. Contrast with Line Mode.

Secured System

A system in which a mandatory security mechanism has been enabled during deadstart. A secured system protects information by enforcing restrictions based on access levels and access categories, and restricts many sensitive system functions to security administrators.

Security Administrator

A secured system prevents users and operators from performing certain functions that could result in the unauthorized disclosure of information. These functions can only be performed by a person who is designated a security administrator. A security administrator is always authorized to access the highest level of information stored on the system. This person performs functions in the areas of installation, user validation, system operation, and system maintenance.

Security Count

The number of security violations you have left before you are denied access to the system.

Security Unlock Status

This status of the system console applies only to a secured system and must be set by a security administrator. The console must be in security unlock status in order for the security administrator to perform certain functions that are restricted on a secured 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.

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 amount 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

A variation of normal input mode (other than transparent input 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 Access Category Set

On a secured system, a set of access categories is set during level 0 deadstart. This set may consist of some, all, or none of the 32 possible access categories. While the system is

running, you may only use access categories that are within the set of system access categories. Refer also to File Access Category and Job Access Categories.

System Access Levels

On a secured system, a range of access levels is set during level 0 deadstart. This range may contain some or all of the eight possible access levels. While the system is running, you may only use access levels that are within the range of system access levels.

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 resources used for a given job.

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 and vanishes once you release it 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 network operator.

Termination Sequence

The character or sequence of characters that causes an executing program to be terminated (also called the user break 2).

Time Slice

The amount of CPU or CM time a job is allowed to use before the system lowers its priority to allow other jobs to execute.

Timed/Event Rollout

A condition where an executing job that 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

An input 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=X, 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.

Unsecured System

A system in which the security mechanism has not been enabled during deadstart. The restrictions based on access levels and access categories are not enforced on an unsecured system.

User Break 1

Refer to Interruption Sequence.

User Break 2

Refer to 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 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 be comprised of more than one volume.

Volume Serial Number (VSN)

A one- to six-character identifier that identifies the volume of magnetic tape to the system.

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 if you request it.

XMODEM

XMODEM is the industry standard error checking communications protocol used to transfer files to and from microcomputers and mainframes. Also known as the Christensen protocol.

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 6-bit display code, two colons create 12 bits of zeros in the 64-character set. 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 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).

Zero Length PRU

A PRU that contains system information, but no user data. Under NOS, a zero-length PRU defines EOF.

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 Multipunch

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.

```
OUTPUT.  
USER,STARKY,HUTCH.  
GET,PFL.  
ITEMIZE,PFL.
```

This job generates three pages of output. The first page is the banner page (shown on D-2), the second is the output generated by ITEMIZE (shown on D-3), and the third is the dayfile (shown on D-4).

OPERATING SYSTEM = NOS 2 crtndate PRINTED - yy/mm/dd 13.48.33.

UJN - OUTPUT FAMILY - MATH JOB ORIGIN - BATCH.
CREATING JSN - ABNS USER NAME - STARSKY SERVICE CLASS - BATCH.

```

AAAAAAAAAA  JJJJJJJJJJ  LL      AAAAAAAAAA  AAAAAAAAAA  P999999999  NN      NN  SSSSSSSSS
AAAAAAAAAA  JJJJJJJJJJ  LL      AAAAAAAAAA  AAAAAAAAAA  P999999999  NN      NN  SSSSSSSSS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  AA  AA  PR      PR  NN  NN  NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  AA  PR      PR  NN  NN  NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  AA  PR      PR  NN  NN  NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  AA  PR      PR  NN  NN  NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  P999999999  NN      NN  NN  SSSSSSSSS
AAAAAAAAAA  JJ      LL      AAAAAAAAAA  AAAAAAAAAA  P999999999  NN      NN  NN  SSSSSSSSS
AAAAAAAAAA  JJ      LL      AAAAAAAAAA  AAAAAAAAAA  P9      PR  NN      NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  P9      PR  NN      NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  PR      PR  NN      NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  PR      PR  NN      NN  SS
AA      AA  JJ      LL      AA      AA  AA  AA  AA  P9      PR  NN      NN  SS
AA      AA  JJ      LL      LL      AA      AA  AA  AA  P9      PR  NN      NN  SS
AA      AA  JJJJJJ  LLLLLLLLLL  AA      AA  AA  AA  P999999999  NN      NN  SSSSSSSSS
AA      AA  JJJJJ  LLLLLLLLLL  AA      AA  AA  AA  P999999999  NN      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.

REC	ITEMIZE OF PFL NAME	TYPE	FILE LENGTH	1 CKSUM	DATE	** UNLABELED ** COMMENTS	yy/mm/dd. 10.16.57.	PAGE ITEMIZE	1
1	INSTALL	PROC	62	6454					
2	BUILD	PROC	65	3131					
3	UPGRADE	PROC	42	1242					
4	TRMINAL	PROC	442	5125					
5	JOB	PROC	22	6332					
6	ISF	PROC	51	1240					
7	PRO	PROC	46	5362					
8	PL	PROC	56	1450					
9	GROUP	PROC	37	0104					
10	XED	PROC	133	0376					
11	EDPROC	PROC	42	6700					
12	RWF	PROC	17	0002					
13	CALC	PROC	20	6034					
14	EDREC	PROC	72	6640					
15	BACKUP	PROC	107	1373					
16	FTN	PROC	112	3504					
17	AJFILE	PROC	34	2403					
18	AJL	PROC	34	5236					
19	MOD	PROC	203	7326					
20	GENPL	PROC	32	0565					
21	EDITPL	PROC	55	7101					
22	PAS	PROC	152	7203					
23	REL	PROC	37	7154					
24	CMP	PROC	40	5164					
25	PSR	PROC	125	6613					
26	PSRFEAT	PROC	47	3106					
27	HPSR	PROC	21	5610					
28	SPRO	PROC	26	5174					
29	DINDEX	PROC	61	3156					
30	RINDEX	PROC	60	1331					
31	INDEX	PROC	124	4474					
32	AJLIST	PROC	27	2560					
33	* EOF *		SUM =	3332					

```
16.58.55.OUTPUT.
16.58.55.USER,STARKY,.
16.58.55.CHARGE,*
16.58.55.* CHARGE(1514,5612PAY)
16.58.56.$PROLOG,PROC2,,
16.58.56.$SETFS,PROC2/FS=AD.
16.58.57.PROC2.
16.58.58.IFE,OT.EQ.TXO,OTHEROT.
16.58.58.ENDIF,OTHEROT.
16.58.58.IFE,$$.NE.$$,END.
16.58.58.ENDIF,END.
16.58.58.$REVERT.CCL
16.58.58.GET,PFL.
16.58.58.ITEMIZE,PFL.
16.58.58. ITEMIZE COMPLETE.
16.58.58.UEAD,      0.002KUNS.
16.58.58.UEPF,      0.057KUNS.
16.58.58.UEMS,      0.770KUNS.
16.58.58.UECP,      0.070SECS.
16.58.58.AESR,      2.186UNTS.
16.58.58.$OUT(* /OP=E)
16.58.58. NO FILES PROCESSED.
16.58.59.$DAYFILE(OUTPUT,JT=D)
16.59.00.UCLP, 42, 070,      0.256KLNS.
```

The first line lists the job's JSN (ABNS); a site-selected, system or machine identifier; the time of job termination (16.58.59); and the current date. 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, 0.057 kilounit of permanent file activity, 0.770 kilounit of mass storage activity, 0.070 seconds of central processor time, and 2.186 system resource units. The first two numbers after UCLP give the machine identifier as 42 and the EST ordinal of the printer as 070. The job produced 0.256 kiloline (256 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		00 00			
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	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

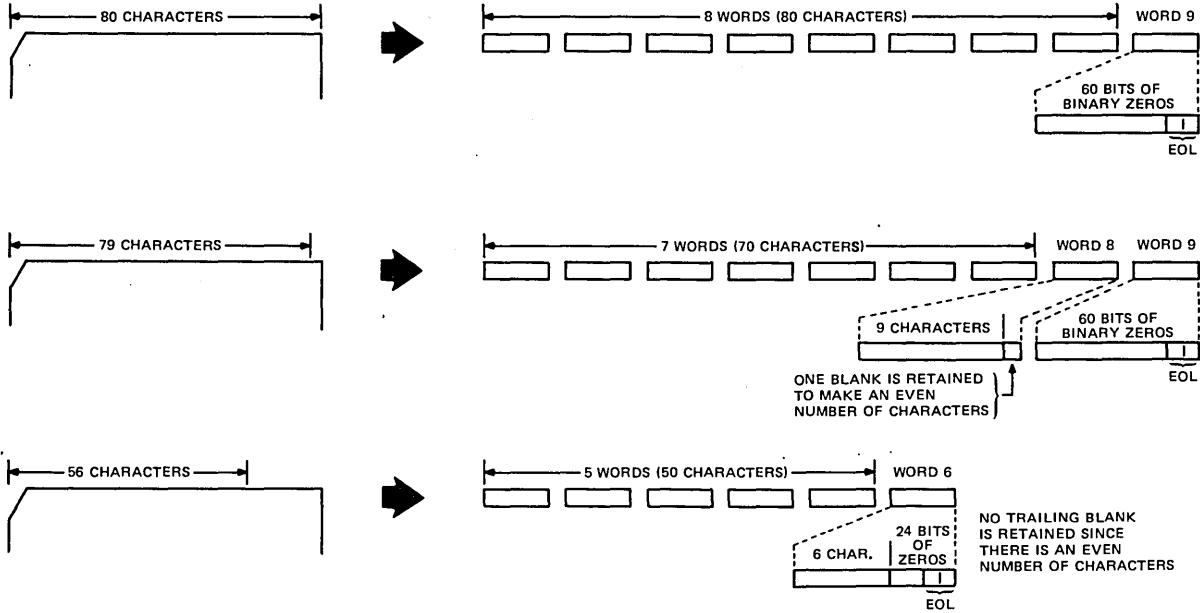
The 5/7/9 card also allows literal input when 4/5/6/7/8/9 is punched in column 2. Literal input allows 80-column binary data to be read while transmitting input in coded mode. Cards are read (16 central memory words per card) until a card identical to the previous 5/7/9 card (4/5/6/7/8/9 in column 2) 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.)

FOR 63 AND 64 CHARACTER SETS



FOR 64 CHARACTER SET ONLY

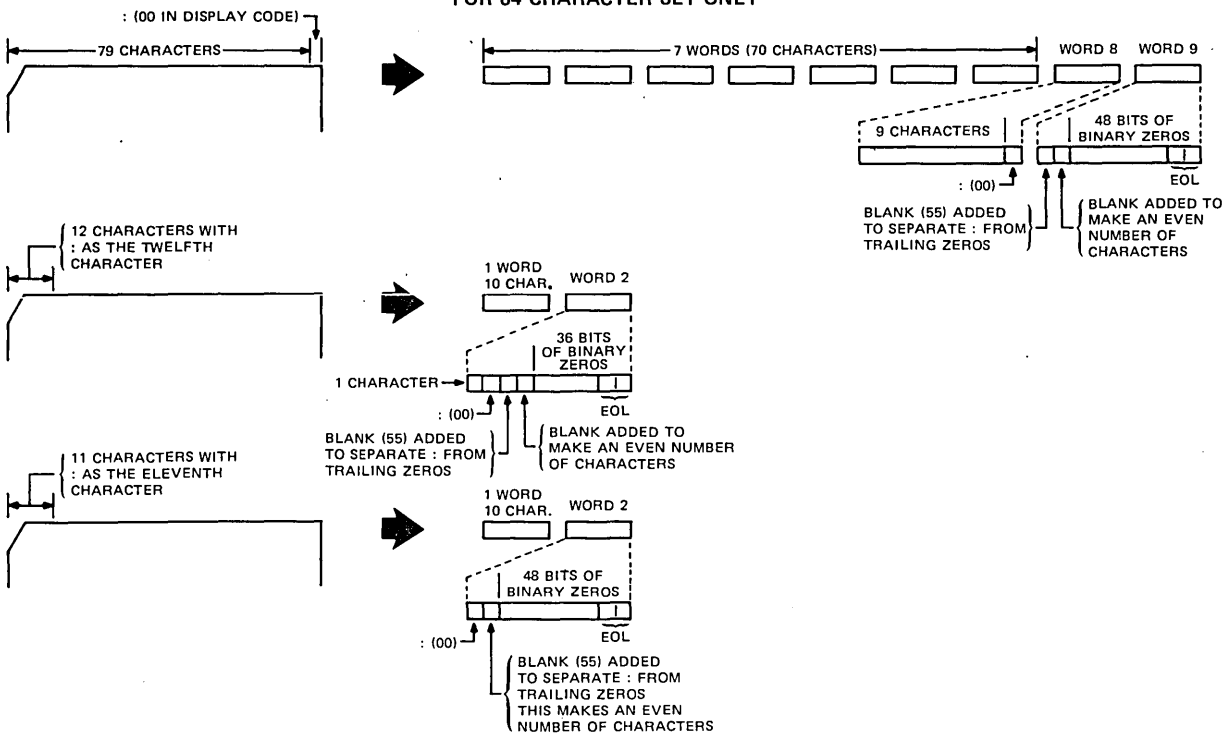


Figure F-1. Examples of Coded Card Conversion

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.

Labels padded to longer than 80 characters and which otherwise meet the ANSI X3.27-1969 standard are not processed by NOS as ANSI labels. Tapes with padded labels can be processed as unlabeled tapes by using the F, S, or L format (in this case, the label data is read or written by a user program).

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 multiframe 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 HDRI 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 field of HDRL.	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 multiframe 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 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-est (ver is the system version number and est 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		

Character Position	Field Name	Length (in Characters)	Contents	Default	Checked on Read
1-3	Label identifier	3	Must be EOV.		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 EOv2 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 EOvN 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.

LINE PRINTER CARRIAGE CONTROL

H

Appendix H provides information required to prepare files for printing.[†] The information includes the following:

- NOS-supported paper lengths.
- Carriage control characters.
- Punched tape channel formats (non-PFC printers only).
- PFC, EVFU, and VFU arrays.

For the 5870 Non-Impact Printer, applicable information in this section is contained under Printed Data below and in table H-1. Note that the Q, R, S, T, and V carriage control characters are not supported by the 5870 printer. If encountered in a print file, these characters are ignored.

Default characteristics for the 5870 printer, such as the print font, page length, and print density are determined by the site. Check with your site administration for this information.

For further information on the functional characteristics of the 5870 printer, refer to the 5870 Site Reference Manual.

PRINTED DATA

All data to be printed is in coded format in a file within the print queue. The data consists of either 6-bit or 12-bit codes. Data recorded using the 6/12-bit display code set (refer to appendix A) should be converted to the 7-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.

PAPER LENGTH

580 LINE PRINTERS

NOS supports two paper lengths for 580 line printers, standard (14 inches x 11 inches) and short (14 inches x 8 1/2 inches). Files routed to a line printer must be properly formatted for the paper length used on the specified printer.

[†]The information in this appendix does not apply to remote batch line printers.

When printing on standard paper size, you have the option of specifying a print density of either 6 or 8 lines per inch. At 6 lines per inch, a maximum of 64 lines per page can be printed on standard paper. At 8 lines per inch, a maximum of 84 lines per page can be printed on standard paper.

All output printed on short paper is printed at a density of eight lines per inch. The carriage control characters used to specify print density (S and T) are ignored. A maximum of 66 lines can be printed on one page of short paper; overflow lines, if any, are printed on the following page.

A system default print density is used if you do not specify one for your print file. The system default is site selectable using the SPD IPRDECK entry; refer to NOS 2 Analysis Handbook for more information on SPD IPRDECK entry.

533/536, 537, AND 585 PRINTERS

The 533/536, 537, and 585 printers are supported by the Printer Support Utility (PSU). These printers can accommodate 8 1/2-, 11-, and 12-inch paper sizes. Modifying the EVFU load file (for 533/536 printers) or the VFU load procedure (for 585 printers) enables these printers to use other paper sizes. Refer to the NOS 2 Analysis Handbook for more information on PSU, and the EVFU load file. Refer to the CDCNET Network Configuration and Site Administration Guide for more information on the VFU load procedure. The 537 printers can use various paper sizes, as determined by settings on the printer. Refer to the NOS 2 Operations Handbook for more information on settings for 537 printers.

Unlike the 580 printers, the 533/536 and 585 printers support both six and eight lines per inch print densities for short paper output, as specified by the S and T carriage control characters. Vertical print density for 537 printers is determined by a switch setting on the printer. Refer to the NOS 2 Operations Handbook for more information on switch settings for 537 printers.

CARRIAGE CONTROL

You can control the format of printer output using the carriage control characters listed in table H-1 and table H-2. The system interprets the first character in a line as the carriage control character and that character is not printed.† The remainder of the line is then printed, except when a Q, R, S, T, or V carriage control character is 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.

†To print a file in which the first character of each line is not a carriage control character, refer to the COPYSBF command in section 9.

The 585 printers support all carriage control characters. The 533/536, 537 printers support all carriage control characters except V. Note that a page eject is issued to the 533/536, 537, and 585 printers whenever the print density is changed or when auto page eject is either set or cleared. Also, for 533/536 printers, auto page eject is set to ON whenever the print density is changed.

The 537 printers defined as printer class C537INT support carriage control characters l, 0, -, +, /, and space. Lines with carriage control characters Q, R, S, and T are suppressed. Carriage control character 3 is treated the same as a l; that is, a page feed is performed before printing. A V causes a page feed, but without printing. All other characters in the carriage control position are translated to a space.

The 537 printers defined as printer class C537BAT support carriage control characters l, 0, -, +, /, A, and space. Period, comma, and characters in the range 0 through 2A (16) result in skipping one line before printing. All other characters cause the line not to be printed.

The system limits the number of lines you can print from a given file (refer to the LIMITS command). Your site can further limit the number of lines you can print by establishing a minimum number of lines per page for which you will be charged. 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 file, the dayfile is subsequently printed.

Table H-1. Carriage Control Characters for 512/580 and 5870 Printers

Character	512/580 Line Charge	5870 Line Charge	Action
space	1	1	Single space.
1	Page Size	1	Eject page before printing.
0	2	2	Skip one line before printing (double space).
-	3	3	Skip two lines before printing (triple space).
+	1	1	Suppress carriage advance before printing (overprint).
/	1	1	Suppress carriage advance after printing (overprint).
2	Page Size/2†	1	Skip to last line of form (format channel 12) before printing.
PM	1	1	Print the rest of the line. Then stop the printer and issue a flashing message to the operator console to indicate a change forms request or other operating instructions.
Q	0	0	Clear auto eject.†† Remainder of line is not printed.
R	0	0	Set auto eject. Remainder of line is not printed.††††
S†††	0	0	Print 6 lines per inch. Remainder of line is not printed.†††††
T†††	0	0	Print 8 lines per inch. Remainder of line is not printed.†††††
V	Page Size†	0	Eject page before printing on a 580 PFC printer. Loads a user-specified PFC array.††††††††
8	1	1	Skip to top of page (format channel 1) before printing.
7	2	1	Skip two lines (format channel 2) before printing.
6	2	1	Skip three lines (format channel 3) before printing.
5	3	1	Skip four lines (format channel 4) before printing.
4	3	1	Skip five lines (format channel 5) before printing.
3	4	1	Skip one page (format channel 6) before printing.
H	1	1	Skip to top of page (format channel 1) after printing.
G	2	1	Skip two lines (format channel 2) after printing.
F	2	1	Skip three lines (format channel 3) after printing.
E	3	1	Skip four lines (format channel 4) after printing.
D	3	1	Skip five lines (format channel 5) after printing.
C	4	1	Skip one page (format channel 6) after printing.

†Page size is 64 lines per page if 6 lines per inch is selected or page size is 85 lines per page if 8 lines per inch selected.
 ††When printing at 8 lines per inch, a Q carriage control character also causes a 580 printer to revert to 6 lines per inch (except when short paper is used).
 †††No line feed after printing. For all other control characters, a line feed is issued after printing. The S and T control characters are ignored when short paper is specified. Short paper output is printed at 8 lines per inch.
 ††††You must be properly validated to use the V carriage control character.
 †††††Not supported by the 5870 Non-Impact Printer.

Table H-2. Carriage Control Characters for 585 Printers (Sheet 1 of 2)

Character	585 Line Charge	Action
space	1	Single space.
1	†	Eject page before printing.
0	2	Skip one line before printing.
-	3	Skip two lines before printing.
+	0	Suppress carriage return before printing (overprint).
/	0	Suppress carriage return after printing (overprint).
2	††	Skip to last line of form before printing.
PM	0	Stop printer and display the rest of the line at the operator console; used to issue a change forms request or other instructions to the operator.
Q	†††	Clear auto eject; remainder of line is not printed.††††
R	†††	Set auto eject; remainder of line is not printed.††††
S	†††	Print 6 lines per inch; remainder of line is not printed.††††
T	†††	Print 8 lines per inch; remainder of line is not printed.††††
V	†††	Eject page before printing; loads a user specified vertical format unit (VFU).††††
8	†	Skip to top of page (format channel 1) before printing.
7	2	Skip two lines (format channel 2) before printing.
6	3	Skip three lines (format channel 3) before printing.
5	4	Skip four lines (format channel 4) before printing.
4	5	Skip five lines (format channel 5) before printing.
3	6	Skip six lines (format channel 6) before printing.
9	7	Skip to format channel 7 before printing.
X	8	Skip to format channel 8 before printing.
Y	9	Skip to format channel 9 before printing.
Z	10	Skip to format channel 10 before printing.

† Number of lines remaining on the page. Page size is 66 if 6 lines per inch is selected and 88 if 8 lines per inch is selected.

†† Number of lines remaining on the page minus 2. If the number of lines remaining on the page is less than 2, count page size.

††† Number of lines according to required paper movement: 0 if format effector is ignored or discarded; 1 if line is printed with single spacing; number of lines remaining if a VFU load with page eject is required.

†††† Causes page eject when action is complete. Unsupported control characters are discarded. Lines containing control characters redefining an existing condition are processed with the unsupported format effector action specified by your site.

Table H-2. Carriage Control Characters for 585 Printers (Sheet 2 of 2)

Character	585 Line Charge	Action
W	†	Skip to format channel 11 before printing.
U	12	Skip to format channel 12 before printing.
A	†	Eject page after printing.
B	††	Skip to last line of form after printing.
H	†	Skip to top of page (format channel 1) after printing.
G	2	Skip two lines (format channel 2) after printing.
F	3	Skip three lines (format channel 3) after printing.
E	4	Skip four lines (format channel 4) after printing.
D	5	Skip five lines (format channel 5) after printing.
C	6	Skip six lines (format channel 6) after printing.
I	7	Skip to format channel 7 after printing.
J	8	Skip to format channel 8 after printing.
K	9	Skip to format channel 9 after printing.
L	10	Skip to format channel 10 after printing.
M	†	Skip to format channel 11 after printing.
N	12	Skip to format channel 12 after printing.

† Number of lines remaining on the page. For example, for a forms length of 11 inches, page size is 66 if 6 lines per inch is selected and 88 if 8 lines per inch is selected.
 †† Number of lines remaining on the page minus 2. If the number of lines remaining on the page is less than 2, count page size.

FORMAT CHANNEL SELECTION ON NON-PFC PRINTERS

Non-PFC line printers, other than 533/536, 537, and 585 printers, use a punched tape loop to control carriage advancement. Figure H-1 shows the released tape format for printers using the standard paper size at 6 lines per inch. Figure H-2 shows the tape format for short paper printers.

The format tape shown in figure H-1 can be ordered as customer engineer's part number 44682683.

Each line, or frame, in the tape format represents one line of printer output. Notice, in figure H-1, that frames 132 through 134 are identical to frames 0 through 2. To form a continuous loop, the ends of the tape are fastened together with the first three frames (0 through 2) superimposed over the last three frames (132 through 134). This leaves a tape loop consisting of 131 frames. The tape format in figure H-1 represents two pages of 64-line printer output. The blank lines at frames 64 and 65, and at frames 130 and 131, represent spaces for the creases between pages.

Each channel controls a specific carriage control operation. Channel 1, for example, provides for top of page spacing, channel 3 controls triple spacing of lines, and channel 12 spaces to the bottom of the page.

For postprint operations, when the carriage control character in column 1 of a print line specifies a channel number, the printer checks that format channel, in the current frame, for a punch. If a punch is found, the printer prints the line. If no punch is found, the printer advances the punched tape loop (and, at the same time, the printer carriage) until a punch is found in that channel in a subsequent frame. At that point, the printer prints the line.

As an example of format channel usage, assume that the printer carriage is positioned at the second line of a page (frame 1 of the format tape). Also assume that the second output line contains a carriage control character of 2. The printer checks frame 1 and finds no punch in channel 12, so it advances to the next frame containing a punch in channel 12. The next punch occurs in frame 63 (line 64); therefore, the second output line is printed as the last line on the page.

FORMAT CHANNEL SELECTION ON 533/536, 537, AND 585 PRINTERS

Format channel selection for 533/536 and 585 printers is defined by the EVFU load file and the VFU load procedure respectively. Refer to the NOS 2 Analysis Handbook for a description of the EVFU load file. Refer to the CDCNET Network Configuration and Site Administration Guide for a description of the VFU load procedure. Format channel selection for 537 printers is not supported at level 688.

Frame	Channels											
	12	11	10	9	8	7	6	5	4	3	2	1
0		x	x	x	x	x	x	x	x	x	x	x
1		x										
2		x									x	
3		x		x					x			
4		x						x		x		
5		x						x				
6		x		x					x	x		
7		x				x						
8		x			x				x	x		
9		x		x					x			
10		x	x					x		x		
11		x								x		
12		x		x					x	x	x	
13		x										
14		x				x				x		
15		x		x				x		x		
16		x			x			x		x		
17		x										
18		x		x					x	x		
19		x										
20		x	x					x	x	x		
21		x		x		x			x			
22		x									x	
23		x										
24		x		x	x			x	x	x		
25		x						x				
26		x									x	
27		x		x					x			
28		x				x			x		x	
29		x										
30		x	x	x				x		x	x	
31		x										
32		x			x				x		x	
33		x		x					x			
34		x									x	
35		x				x						
36		x		x				x	x	x		
37		x										
38		x									x	
39		x		x					x			
40		x	x		x			x	x		x	
41		x										
42		x		x		x			x	x		
43		x										
44		x							x		x	
45		x		x				x		x		
46		x									x	
47		x										
48		x		x	x				x	x	x	
49		x				x						
50		x	x					x			x	
51		x		x					x			
52		x							x		x	
53		x										
54		x		x					x	x		
55		x						x		x		
56		x			x	x			x		x	
57		x		x						x		
58		x									x	
59		x										
60		x	x	x				x	x	x	x	
61		x						x	x	x		
62		x								x		
63	x	x		x		x				x		
64												
65												
66		x	x	x	x			x	x	x	x	x
67		x										

Frame	Channels											
	12	11	10	9	8	7	6	5	4	3	2	1
68		x										x
69		x			x							x
70		x									x	x
71		x										
72		x			x						x	x
73		x							x			
74		x							x			
75		x			x						x	x
76		x	x								x	x
77		x										
78		x			x						x	x
79		x										
80		x							x			x
81		x			x					x		x
82		x							x			x
83		x									x	x
84		x			x						x	x
85		x										
86		x	x								x	x
87		x			x				x			x
88		x										x
89		x										
90		x			x				x		x	x
91		x										
92		x									x	x
93		x							x			
94		x										
95		x									x	x
96		x	x								x	x
97		x										
98		x							x			x
99		x									x	
100		x										x
101		x									x	
102		x							x		x	x
103		x										
104		x										x
105		x										
106		x	x								x	x
107		x									x	
108		x										x
109		x										
110		x										x
111		x									x	x
112		x										x
113		x										
114		x									x	x
115		x									x	
116		x	x								x	x
117		x										
118		x										x
119		x										
120		x										x
121		x									x	
122		x									x	x
123		x										
124		x										x
125		x										
126		x	x								x	x
127		x										
128		x										x
129	x	x										
130												
131												
132		x	x	x	x	x	x	x	x	x	x	x
133		x										
134		x										

Figure H-1. Carriage Control Tape Format (Standard Paper Length; 6LPI)

Frame	Channels											
	12	11	10	9	8	7	6	5	4	3	2	1
0		x	x	x	x	x	x	x	x	x	x	x
1		x										
2		x										
3		x										
4		x										
5		x										
6		x										
7		x										
8		x										
9		x										
10		x										
11		x										
12		x										
13		x										
14		x										
15		x										
16		x										
17		x										
18		x										
19		x										
20		x										
21		x										
22		x										
23		x										
24		x										
25		x										
26		x										
27		x										
28		x										
29		x										
30		x										
31		x										
32		x										
33		x										
34		x										
35		x										
36		x										
37		x										
38		x										
39		x										
40		x										
41		x										
42		x										
43		x										
44		x										
45		x										
46		x										
47		x										
48		x										
49		x										
50		x										
51		x										
52		x										
53		x										
54		x										
55		x										
56		x										
57		x										
58		x										
59		x										
60		x										
61		x										
62		x										
63		x										
64		x										
65	x	x										
66		x										
67		x										
68		x										
69		x										

Frame	Channels											
	12	11	10	9	8	7	6	-5	4	3	2	1
70		x										
71		x										
72		x										
73		x										
74		x										
75		x										
76		x										
77		x										
78		x										
79		x										
80		x										
81		x										
82		x										
83		x										
84		x										
85		x										
86		x										
87		x										
88		x										
89		x										
90		x										
91		x										
92		x										
93		x										
94		x										
95		x										
96		x										
97		x										
98		x										
99		x										
100		x										
101		x										
102		x										
103		x										
104		x										
105		x										
106		x										
107		x										
108		x										
109		x										
110		x										
111		x										
112		x										
113		x										
114		x										
115		x										
116		x										
117		x										
118		x										
119		x										
120		x										
121		x										
122		x										
123		x										
124		x										
125		x										
126		x										
127		x										
128		x										
129		x										
130		x										
131		x										
132		x										
133	x	x										
134		x										
135		x										
136		x										
137		x										
138		x										

Figure H-2. Carriage Control Tape Format (Short Paper Length; 8 LPI)

CARRIAGE CONTROL ARRAYS

Line spacing on a printer is controlled by a carriage control array. This array has the following forms for the various printers:

- Programmable format control (PFC) array for 580 printers.
- Carriage control tape for non-PFC 580 printers.
- Electronic vertical format unit (EVFU) for 533/536 printers.
- Vertical format unit (VFU) for 585 printers.

The selection of an array for any printer is dependent on the print density and the paper length to be used. You can specify one of several arrays already defined by the system using the SC parameter on the ROUTE command. These arrays are described in the NOS 2 Analysis Handbook. You can also select a user-specified PFC or VFU array to control print spacing. You must have validation to use a user-specified PFC array (refer to the LIMITS command). A user-specified array remains in effect until another array is encountered or the end-of-file is reached. The print density for printers using standard sized paper is selected using the S or T carriage control characters. 580 printers using short paper always use a print density of 8 lines per inch and ignore S and T.

A user-specified array begins with the control character V. When a V is encountered, the printer ejects the page. For non-PFC 580 printers, the rest of the line is ignored. For PFC 580 printers, if you are not validated to use the V carriage control character, the print operation aborts and you receive an appropriate message in your output file.

ARRAY SYNTAX

An array consists of a string of alphanumeric characters that can be used to redefine some or all of the format channels of the previously specified array. An array for output at 6 lines per inch occupies one line of up to 132 characters plus the terminator for 580 printers, up to 127 characters for 585 printers. An array for 8 lines per inch may occupy one or two lines. For printing on standard length paper, an array for 8 lines per inch can consist of up to 176 characters on 580 printers, 127 characters on 585 printers (88 lines per page). For short paper printers, an array for 8 lines per inch must not exceed 136 characters plus the terminator for 580 printers, 127 characters for 585 printers (68 lines per page).

Column 1 of an array must contain the carriage control character V. The second character of the array determines which array is changed and may be one of the following values.

<u>Value</u>	<u>Description</u>
6	6 lines per inch; the entire array specification occupies one line.
8	8 lines per inch; the entire array specification occupies one line.
C	8 lines per inch; the array specification continues onto a second line. The second line begins in column 3; columns 1 and 2 of the second line are ignored.

The array must be terminated by the letter O. Starting in column 3 and continuing up to the column that precedes the array terminator (letter O), each column in the array represents one line of printer output. A format channel indicator placed in one of these columns in a user-specified array changes the format channel in effect for the corresponding output line. If an array column is left blank, the format channel for the corresponding line remains unchanged. The channel indicators that can be inserted in these columns are as follows.

<u>Character</u>	<u>Significance</u>
A	Specifies top of form code (channel 1); mandatory entry in column 3 of the array (representing output line 1). Delete previous channels.
B through L	Specify format channels 2 through 12, respectively. Delete previous channels.
M	Specifies auto page eject channel. Applies to 585 printers only.
N	Specifies bottom of form channel. Applies to 585 printers only.
O	Array terminator.
X	Remove the channel format previously in effect for the line. Applies to 585 printers only.
Blank	No channel is specified; the channel format previously in effect for the line remains unchanged.

In the 580, each channel number referenced in your PFC array must appear in the PFC array previously in effect. Channel specifications can be repeated within an array.

NOTES

If a PFC array contains an invalid character, the array line is printed, and the printing of the file is terminated. Other arrays with invalid characters are ignored, and the file is printed using the carriage control array previously loaded into the printer.

If a VFU array contains an invalid character, the line containing the V control character is processed according to the undefined format effector action.

Following are some examples of invalid array specifications:

<u>Invalid Array</u>	<u>Error</u>
V6BCD 0	Does not contain an A in column 3.
VBA C DEO	Second character is not 6, 8, or C.
V8ABWCO	Contains an invalid character (W).
V6A 0	Contains a zero for the terminator.

RESERVED FORMAT CHANNELS FOR VFU PROCESSING

For printers that support format channels, three channels are reserved for specific carriage control operations. Channel 1 is always used to position the paper to the top-of-form. Two additional site selected channels are also reserved: one for positioning the paper to the logical bottom-of-form, and one for skipping over the perforations between forms (auto page eject). Each of these channels may appear only once in a VFU array. By default, both the bottom-of-form and auto-page-eject channels will be placed two lines above the physical end of the form. If the operator or user changes form size or print density, the bottom-of-form and auto-page-eject channels are moved to accommodate the change in the number of lines on the form. If the auto-page-eject option is changed, then the auto-page-eject channel is removed and restored in the VFU array.

EXAMPLE

The following is an example of an array used to print a special 8-line form.

Suppose you want to print on the top, fifth, and bottom lines of an 8-line form (at 8 lines per inch print density). Load the following array:

```
Columns: 1  2  3  4  5  6  7  8  9 10 11 12
          V  8  A           B   N           O
```

Since columns 4, 5, 6, 8, 10, and 11 are blank, the format channels for these columns remain unchanged from the array previously in effect. You might want to put an X or a dummy channel indicator (that is, a valid channel indicator that is not otherwise referenced in the file), such as a K, into these columns to ensure that the B and N channel indicators appear only where intended.

After loading the example array above, output lines beginning with an 8 or H (format channel 1) are printed at the top of the form, lines beginning with a 7 or G (format channel 2) are printed at the fifth line, and lines beginning with a 2 (bottom-of-form channel) are printed at the bottom of the form.

OBSOLETE TAPE FORMATS

I

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>Character istics</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 (1000 _g) 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.

NAM/CCP TERMINAL DEFINITION COMMANDS

J

INTRODUCTION

The NAM/CCP network, through which terminals access a host and its applications (like IAF), can support many types of terminals. All supported terminals are grouped by the network into 18 standard terminal classes. Each terminal class has specific operating characteristics, also referred to as terminal characteristics. These terminal characteristics, taken together, make up a terminal definition and are predefined by the network to closely match the operating characteristics of actual terminals. The network accepts changes to any of these terminal definitions through terminal definition commands. These commands specify one or more parameters, each of which may give a new value to a terminal characteristic. Tables J-1 and J-2 list these terminal definitions, the default values, and possible values for each of the 18 classes.

The tables use the following abbreviations when identifying the terminal types for each terminal class:

<u>Abbreviation</u>	<u>Terminal</u>
HASP	Terminals using the Houston Automatic Spooling Program communication protocol.
HAZ	Hazeltine.
POST	HASP terminals that only support postprint format control.
PRE	HASP terminals that only support preprint format control.
Tek	Tektronix (teletypewriter emulation).
TTY	Teletype.
UT	User Terminal.
X.25	Terminals connected to a packet switching network (PSN) using the X.25 communication protocol.
X3.64	Terminals conforming to the ANSI X3.64 standard.

When a terminal logs in to the network, either the network assumes it to be of a certain terminal class or assigns it a terminal class that resembles the terminal's actual characteristics. If you do not know the terminal class in use, you can enter the CH command (described in this appendix) to ascertain the terminal class and some of the characteristics of your terminal. In either case, if the characteristics of your terminal do not match those of its assigned terminal class, you can change the values of the terminal characteristics or even change the terminal class by using the terminal definition commands described in this appendix. You can enter terminal definition commands any time the terminal is connected to the network (refer to login/logout procedures in Volume 1, Introduction to Interactive Usage).

When you use a terminal definition command to change a value, that change remains in effect until the terminal is disconnected from the network or another terminal definition command is used to change the value. Even application switching and logout do not change the values of the terminal characteristics if the terminal has not been disconnected from the network. Disconnecting your terminal from the network (that is, disconnecting the phone and redialing on dial-up terminals or turning a dedicated terminal off and then on again) causes all terminal characteristics to be reset to site-defined default values. The site defines these values using the Network Definition Language (NDL).

You can also reset your terminal's characteristics by entering the TC command. The TC command resets your terminal characteristics to the default values specified by the Communications Control Program (CCP). Note that CCP-defined values may differ from the site-defined NDL values.

It is also possible to change terminal definitions using the TRMDEF command or by using control byte 16 from an interactive job. Section 8 describes the TRMDEF command under Terminal Definition Commands. Refer to Volume 4, Program Interface, for information on control bytes.

TERMINAL DEFINITION COMMAND FORMAT

This section describes the general format of a terminal definition command. The specific format of each command appears later in this appendix.

General formats:

```
ct keyword=value†  
ct keyword1=value1 ct keyword2=value2...ct keywordn=valuen
```

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command. Table J-1 gives the default network control character for your terminal.††
keyword	Represents a mnemonic associated with a terminal characteristic.
value	Specifies a particular value associated with the terminal definition mnemonic. Table J-2 gives the range of values for each terminal definition keyword. On some commands the =value portion of the command is optional and defaults to =Y.
keyword _i	Same as keyword.
value _i	Same as value.

The second format allows you to make multiple terminal definitions in one line. If the line extends beyond character position 54, the results are unpredictable. Also, if you change the network control character ct in a concatenated group of commands, the CT command does not take effect until the remaining commands in the group have been processed.

As with all commands described in this manual, you must press the message transmission key in order for the system to process your command. Appendix K shows the default message transmission keys for the various terminal classes. This appendix describes how you can change the message transmission key (refer to the EB and EL commands).

Example:

To change the page width of a 200 User Terminal, enter these characters

```
%PW=50
```

and press the SEND key. The % character is the network control character, PW is the keyword associated with page width, 50 is the new page width for the terminal, and the SEND key is the message transmission key for a 200 User Terminal.

†The spaces shown are for clarity only. Do not include spaces when you enter a terminal definition command.

††You must press the ATTN key on a 2741-compatible terminal before entering this character.

RESTRICTIONS

You should not enter terminal definition commands while output is in progress. The network may not be able to perform such commands immediately.

The network presets all user-adjustable terminal characteristics any time you establish a connection to the network. The site administrators choose these preset values. Additionally, when a terminal is disconnected from the network and then reconnected, the network resets any terminal characteristics you have changed to their preset values.

To enter a terminal definition command at asynchronous terminals from which you can enter more than one line before transmitting (IN=BK is specified), the terminal definition command must be the last input line in the transmission. If the input is from paper tape (IN=PT is specified), the tape reader must stop. Unpredictable results can occur if more input follows the terminal definition command in the transmission block.

Characters or character codes you specify for the value portion of terminal definition commands are subject to the following restrictions:

- You cannot use the following ASCII characters (or their character codes) in the value portion of the AB, CN, BS, B1, B2, CT, EB, or EL commands:
 - any lowercase or uppercase alphabetic character
 - any digit 0 through 9
 - NUL
 - SOH
 - STX
 - =
 - DEL
 - space
- The ASCII characters or character codes you specify for the AB, BS, B1, B2, CN, CT, EB, and EL terminal definitions must all be different, except that AB and CN can have the same value and EB and EL may have the same value.
- If output flow control is used, do not use DC1s and DC3s in the AB, CN, BS, B1, B2, CT, EB, or EL commands.

TERMINAL DEFINITION COMMANDS

This subsection describes the function and format of each terminal definition command. The headings consist of the command name (a two-character mnemonic) followed by the terminal attribute it defines. Generally, the format descriptions do not show the default values or ranges of values since they vary with terminal class. Tables J-1 and J-2 show this information. If you are unable to find the key on your keyboard that corresponds to a particular character given in table J-1, consult the documentation for your terminal.

This section refers to categories of terminal classes to which commands apply. The categories are as follows:

<u>Category</u>	<u>Terminal Classes</u>
Asynchronous	1 through 8
Mode 4	10, 11, 12, 13, 15
HASP	9, 14
Bisynchronous	16, 17
X.25 PSN	1, 2, 3, 5, 6, 7, 8
3270	18

(AB) - ABORT OUTPUT BLOCK CHARACTER

The AB command specifies the character used to abort an output block. When you enter this character from the terminal as the only character in a line, the current transmission block of output to the terminal is discarded.

Format:

ct AB=ab

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ab	Specifies the new abort output block character. This character must differ from the characters defined for BS, B1, B2, CT, EB, and EL.

This command only applies to asynchronous terminals and is not valid for X.25 PSN terminals.

(AR) - AUTOMATIC CHARACTER RECOGNITION

The AR command allows you to change the terminal's character set or its line speed, in many cases, by means of a switch on the terminal or a removable type ball. You can enter this command any time after the system prompts you for input, even before you enter your user name during login.

Format:

ct AR

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

After entering the AR command, you must select the terminal speed and enter a carriage return. The system acknowledges the carriage return operation and indicates that the line speed recognition is complete by returning two line feeds. If your terminal cannot change its line speed, no line feed responses occur. At this point, enter a closing parenthesis followed by a carriage return so that the system can recognize the terminal's new character set. When the system recognizes the new character set, it issues a line feed and the message AR ACCEPTED.. to indicate you can continue. This is also the auto-recognition procedure used to connect the terminal to the system (refer to the login/logout procedures in Volume 1, Introduction to Interactive Usage).

This command only applies to asynchronous terminals and is not valid for X.25 PSN terminals. X.25 PSN terminals use the ASCII code and character set.

(BF) - BLOCKING FACTOR

The BF command sets the size of intermediate data blocks that will be delivered to the application.

Format:

ct BF=bf

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
bf	Specifies the blocking factor. You can specify the following values for bf:
<u>bf</u>	<u>Description</u>
0	Selects a line feed character and a character count of 100 as data-forwarding signals.
1	Selects a character count of 100 as a data-forwarding signal.
2	Selects a character count of 200 as a data-forwarding signal.

Regardless of the blocking factor, an end-of-line character (usually a carriage return) forwards data to the application.

(BR) - BREAK KEY AS USER BREAK 1

The BR command determines if the BREAK key (interruption key for interactive output) also functions as the interruption sequence (user break 1 sequence) when entered during output and as the cancel character when entered during input.

Format:

ct BR=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Determines if this feature is enabled or disabled. A value of Y enables and a value of N disables this feature.

This command only applies to asynchronous terminals. For X.25 PSN terminals, the BREAK key can only function as the interruption sequence whether entered during input or output.

(BS) - BACKSPACE CHARACTER

The BS command specifies the character used to delete the previous input character.

Format:

ct BS=bs

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
bs	Specifies the new backspace character. This character must differ from the characters currently defined for AB, B1, B2, CN, CT, EB, and EL.

It is possible to backspace only to the beginning of the current physical line; additional backspaces are disregarded. When a page width of 0 is selected, the network assumes a page width of 100 characters for backspacing. If the page width is in the range 20 through 56, backspacing into previous physical lines is sometimes permitted.

This command only applies to asynchronous terminals only.

(B1) - INTERRUPTION OR USER BREAK 1 CHARACTER

The B1 command specifies the character that, when entered as the only character in a logical line (interruption character followed by the message transmission key), causes program interruption.

Format:

ct B1=ubl

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ubl	Specifies the new user break 1 character. This character must differ from the characters currently defined for AB, BS, B2, CN, CT, EB, and EL.

The interruption sequence is discussed in section 16. The BR command determines whether the BREAK key can also function as the B1 character.

This command does not apply to bisynchronous 2780 and 3780 terminals.

(B2) - TERMINATION OR USER BREAK 2 CHARACTER

The B2 command specifies the character that, when entered as the only character in a logical line (termination character followed by the message transmission key), causes program termination.

Format:

ct B2=ub2

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ub2	Specifies the new user break 2 character. This character must differ from the characters currently defined for AB, BS, B1, CN, CT, EB, and EL.

The termination sequence is discussed in section 16.

This command does not apply to bisynchronous 2780 and 3780 terminals.

(CH) - DISPLAY TERMINAL CHARACTERISTICS

The CH command displays some of the terminal's current characteristics.

Format:

ct CH

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

The CH command produces a display in the following format:

TC=tc,BS=bs,CN=cn,AB=ab,B1=ub1,B2=ub2,EL=e1,EB=eb

where the variable portions appear as ASCII characters if printable, mnemonics (for example, STX) if nonprintable, or N/A if the attribute is not applicable.

(CI) - CARRIAGE RETURN IDLE COUNT

The CI command specifies the number of idle characters to be inserted into the output stream after a carriage return.

Format:

ct CI=ci

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ci	Specifies the new carriage return idle count. This value can be from 0 to 127 or the string CA. A value of CA restores the carriage return idle count to the default value.

When the network produces a carriage return, 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.

This command only applies to asynchronous terminals.

(CN) - CANCEL CHARACTER

The CN command specifies the character used to cancel the logical line currently being input.

Format:

ct CN=cn

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
cn	Specifies the new cancel character. This character must differ from the characters currently defined for BS, B1, B2, CT, EB, and EL.

When you enter the cancel character as the last character in a line, the entire logical line in progress is cancelled. 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. The BR command determines whether the BREAK key can also function as the CN character.

This command does not apply to bisynchronous 2780 and 3780 terminals.

(CP) - CURSOR POSITIONING AFTER INPUT

The CP command determines whether or not the system sends the terminal a response to the line feed key, the EL character, and the EB character.

Format:

ct CP=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Indicates whether to send a response to the terminal or not. A value of Y indicates yes and N indicates no.

This command applies only to asynchronous terminals (except 2741s). If CP=Y and IN=KB are selected, the system responds to the line feed key by sending a carriage return and responds to an EL or EB character by sending the cursor positioning response specified by the EL or EB command. If IN=BK is selected or you are at an X.25 PAD terminal, the system responds only to the EB character by sending the cursor positioning response selected by the EB command.

(CT) - NETWORK CONTROL CHARACTER

The CT command specifies a network control character for the terminal. When you enter this character as the first character of a logical line, it signals that what follows is a terminal definition command or a special command that the network forwards to your application with a preemptive status (refer to the interactive status commands in section 8).

Format:

ct CT=char

<u>Parameter</u>	<u>Description</u>
ct	Represents the current network control character for your terminal.
char	Specifies the new network control character. This character must differ from the characters currently defined for AB, BS, B1, B2, CN, EB, and EL.

If you enter the CT command in a line with multiple terminal definitions, the new value char does not take effect until the complete line is processed.

(DL) - DELIMITERS FOR SINGLE-MESSAGE TRANSPARENT INPUT MODE

The DL command specifies the delimiters that terminate single-message transparent input mode. When you initiate transparent input mode (IN=X, IN=XK, or IN=XP), the network reads your input and sends it to the system without translation until it encounters a delimiter. If your terminal's parity is set to even, odd, or zero (PA=E, O, or Z), the parity bit is stripped (set to zero) before the characters are sent to the host, although no translation occurs. If your terminal's parity is set to none or ignore (PA=N or I), the parity bit is not stripped. See the NAM/CCP Terminal Interfaces Reference Manual for more information on transparent input and parity. The first delimiter encountered terminates transparent input mode.

Format:

```
ct DL=Xxx,Ccount,T0
```

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
Xxx	Specifies the 2-digit hexadecimal code (xx) of the character you want as a delimiter. The network does not send the character as part of the input data. When you specify a hexadecimal code above 7F, you must have previously set your terminal's parity to none (PA=N) or ignore (PA=I); otherwise the network cannot detect the code when it occurs in a transparent mode message. If the terminal's parity is set to none, the 8-bit hexadecimal code input from the terminal is matched against the 8-bit hexadecimal code specified as the delimiter (xx). If the terminal's parity is set to ignore, the bottom 7 bits of the terminal input are matched against the bottom 7 bits of the specified delimiter (xx).
Ccount	Specifies a decimal value (count) from 1 to 4095 that functions as a character count delimiter. These values are valid within the network. However, unless changed during installation, the maximum logical input line length in IAF (2500 characters) limits you.
T0	Specifies that a timeout of between 200 and 400 milliseconds will be a transparent mode delimiter.

You can select any of the three types of delimiters: character delimiter (specify Xxx), character count delimiter (specify Ccount), and timeout delimiter (specify T0).

The parameter values for DL are order-independent and optional. However, you must specify at least one of the parameter values. You can omit trailing commas if you do not specify all three types of delimiters.

Entering this command with any number of new delimiters cancels all transparent mode input delimiters already in effect, including those specified with the XL command.

The DL command applies only to asynchronous and mode 4 terminals. Terminal class 4 is configured with the RETURN key as the only transparent mode terminator and mode 4 terminals are configured with the SEND or ETX key as their only transparent mode terminator. For X.25 devices, the PAD forwarding signal always acts as a transparent mode terminator in addition to any other options selected.

(EB) - END-OF-BLOCK CHARACTER

The EB command defines the end-of-block character and defines the cursor-positioning response to that character. The EB character acts as the message transmission key for terminals operating in block mode (refer to the IN command).

Format:

ct EB=eb,cpr

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
eb	Specifies the new end-of-block character. You can specify the following values for eb:
<u>eb</u>	<u>Description</u>
Xxx	Specifies the 2-digit hexadecimal code xx of the character you want as the end-of-block delimiter. This character must differ from the characters currently defined for AB, BS, B1, B2, CN, and CT.
EB	Selects the default end-of-block character or character sequence.
EL	Selects the current logical end-of-line character.
cpr	Specifies the cursor-positioning response to the EB character. You can specify the following values for cpr:
<u>cpr</u>	<u>Description</u>
CR	Send a carriage return to the terminal.
LF	Send a line feed to the terminal.
CL	Send a carriage return and a line feed to the terminal.
NO	Send no response.

The two EB parameters (EB=eb and cpr) are optional and order-independent. If you omit one, the system retains its previous value.

This command is valid only for asynchronous terminals and mode 4 terminals, except for paper tape devices and 2741 terminals. The end-of-block character for mode 4 terminals is ETX and is not changeable. If you specify the char form of the eb parameter, it is only effective if it occurs as the last character in a line.

The CP command enables or disables the cursor-positioning response specified by the cpr parameter.

(EL) - END-OF-LINE CHARACTER

The EL command defines the logical end-of-line character (carriage return) and defines the cursor-positioning response to that character. It also defines the message transmission key for asynchronous terminals not operating in block mode (refer to the IN command).

Format:

ct EL=el, cpr

<u>Parameter</u>	<u>Description</u>										
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.										
el	Specifies the new logical end-of-line character. You can specify the following values for el: <table><thead><tr><th><u>el</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>Xxx</td><td>Specifies the 2-digit hexadecimal code xx of the character you want as the end-of-line delimiter. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, and CT.</td></tr><tr><td>EB</td><td>Selects the current end-of-block character or character sequence.</td></tr><tr><td>EL</td><td>Selects the default logical end-of-line character.</td></tr></tbody></table>	<u>el</u>	<u>Description</u>	Xxx	Specifies the 2-digit hexadecimal code xx of the character you want as the end-of-line delimiter. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, and CT.	EB	Selects the current end-of-block character or character sequence.	EL	Selects the default logical end-of-line character.		
<u>el</u>	<u>Description</u>										
Xxx	Specifies the 2-digit hexadecimal code xx of the character you want as the end-of-line delimiter. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, and CT.										
EB	Selects the current end-of-block character or character sequence.										
EL	Selects the default logical end-of-line character.										
cpr	Specifies the cursor-positioning response to the EL character. You can specify the following values for cpr: <table><thead><tr><th><u>cpr</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td>CR</td><td>Send a carriage return to the terminal.</td></tr><tr><td>LF</td><td>Send a line feed to the terminal.</td></tr><tr><td>CL</td><td>Send a carriage return and a line feed to the terminal.</td></tr><tr><td>NO</td><td>Send no response.</td></tr></tbody></table>	<u>cpr</u>	<u>Description</u>	CR	Send a carriage return to the terminal.	LF	Send a line feed to the terminal.	CL	Send a carriage return and a line feed to the terminal.	NO	Send no response.
<u>cpr</u>	<u>Description</u>										
CR	Send a carriage return to the terminal.										
LF	Send a line feed to the terminal.										
CL	Send a carriage return and a line feed to the terminal.										
NO	Send no response.										

The two EL parameters (el and cpr) are optional and order-independent. If you omit one, the system retains its previous value.

This command does not apply to 2741, HASP, 3270, and bisynchronous 2780 and 3780 terminals; the cpr parameter does not apply to mode 4 or X.25 PSN terminals; and the char specification does not apply to mode 4 terminals.

The CP command enables or disables the cursor-positioning response specified by the cpr parameter.

(EP) - ECHOPLEX MODE

The EP command enables or disables the echoing of input characters back to the terminal.

Format:

ct EP=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) input echoing.

Some terminals can perform their own character echoing, such as terminals with a HALF/FULL duplex switch. If the switch is in the HALF position, set the EP value to N, and the terminal echoes the input. If the switch is in the FULL position, set the EP value to Y and the network echoes the input.

You can use the EP command and full-duplex mode as a security measure. If you are in full-duplex mode and EP is Y during login, the system echoes all of your input except your password. This is not true, however, if you enter your user name and password on the same line.

This command only applies to asynchronous terminals (except 2741s).

(FA) - FULL ASCII INPUT MODE

The FA command enables or disables the full-ASCII input mode.

Format:

ct FA=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) the full-ASCII input mode.

In full-ASCII input, all 128 ASCII characters, including NUL, DEL, STX, LF, and all BLK and message-forwarding signals (which are normally discarded), are sent to the host. Exceptions to this are the following characters:

- NULs, DELs and LFs when encountered as the first character of an input line or as the character following a message-forwarding signal
- CRs that follow LFs when inputting from paper tape
- DC3s that follow the EL character when inputting from paper tape
- DC1s and DC3s when output control (OC=Y) is selected

In full-ASCII mode, the network does not recognize the terminal functions associated with the mnemonics B1, B2, CN, AB, and BS. The network does recognize terminal definition commands and other commands beginning with the CT character.

If SE is in effect when you select FA or if FA is in effect when you select SE, FA takes precedence and SE will not be effective until FA is turned off. If full-ASCII mode is enabled when you select transparent input mode, the network suspends the full-ASCII mode until the transparent input mode terminates.

This command only applies to asynchronous and mode 4 terminals.

(HC) - HOST CONNECTION

The HC command disconnects you from your current host and establishes a new host connection. The new host connection can be selected by an HN command or by the HC command (HC=hostname). The HC command is equivalent to HS or HN followed by TM.

Format:

ct HC=hostname

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
hostname	Specifies the name of the host to which you want your terminal connected.

The host selected is used until you change it or until your terminal is disconnected from the host.

(HD) - DISPLAY OF HOST NODES

The HD command causes the host availability display (HAD) to be displayed at your terminal. The HAD lists all paths you can select to connect your terminal to a host.

Format:

ct HD

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

The HAD has the following format:

```

message          CONTROL CHARACTER=ct
NPU NODE=npu    TERMINAL NAME=terminal

  HOST           NODE           SELECTED/
                   CONNECTED
host1          node1          condition1
host2          node2          condition2
.
.
.
hostn         noden          conditionn
                   statusn
prompt response

```

The variable items in the display have the following descriptions.

<u>Item</u>	<u>Description</u>
message	A message describing the status of your current host connection.
ct	Network control character currently defined for your terminal.
npu	The node number of the Network Processing Unit (NPU) your terminal is connected to.
terminal	The name of your terminal.
host _i	Represents the 1- to 7-character name of a host. The field may be blank.
node _i	Represents a node number (1-31).
condition _i	Indicates that you have selected node _i (S), that you are connected to the host using node _i (C), or that node _i has been automatically selected and the connection to the host is in progress (SA).
status _i	Represents AVAILABLE or NOT AVAILABLE.
prompt message	Action to take in response to a host status message.

(HN) - HOST NODE SELECTION

The HN command sets or changes the host or the path (node) to a host through which your terminal is connected. This command does not disconnect you from your current host connection. You must log out or disconnect with the Secure Login command or TM terminal definition command before you can connect to the new selection.

Format:

ct HN=node

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
node	Specifies a node number. The default is the node the network processing unit is presently using to access its supervisor (CS).

The host availability display (HAD) lists all nodes your terminal can use. Refer to the HD command for a description of the HAD.

(HS) - HOST SELECTION

The HS command selects a host which is connected to your terminal. The host availability display (HAD) shows the available hosts (refer to the HD command).

Format:

ct HS=hostname

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
hostname	Specifies the name of the host to which you want your terminal connected. The valid entries appear in the HAD.

Your terminal uses the path to host with the least traffic. The path selected is used until either you change it or until your terminal is disconnected from the host.

(IC) - FLOW CONTROL FOR INPUT DEVICES

The IC command specifies whether the terminal recognizes DC3s as a stop signal and DC1s as a resume signal for terminal input.

Format:

ct IC=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) input control.

Input control applies to asynchronous terminals, including paper tape devices (except 2741s).

(IN) - INPUT DEVICE AND TRANSMISSION MODE

The IN command identifies the input device and the transmission mode of the device. As the input device, you can specify one of the following:

- A keyboard.
- A paper tape reader.

As the input transmission mode, you can specify either:

- Normalized line mode (one message or logical line per terminal transmission block).
- Normalized block mode (one or more logical lines or messages collected into a block before it is transmitted).
- Transparent mode (input is not edited or translated by the network).

Format:

ct IN=in

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
in	Specifies the input device and transmission mode. You can specify the following values for in:

<u>in</u>	<u>Description</u>
BK	Selects block mode as from a form terminal and normalized input mode.
KB	Selects a keyboard in character or line mode and normalized input mode.
PT	Selects paper tape and normalized input mode.
X	Selects the current input device and transparent input mode.
XK	Selects the keyboard and transparent input mode.
XP	Selects paper tape and transparent input mode.

When the IN value is XK, XP, or X, terminal input is sent to the host in transparent mode blocks. Transparent mode continues until the system encounters a previously defined transparent mode terminator.

When the IN value is XK and transparent mode terminates, the IN value automatically reverts to KB. When the IN value is XP and transparent mode terminates, the IN value automatically reverts to PT. When the IN value is X and transparent mode terminates, the IN value automatically reverts to the value that was previously in effect.

When the IN value is PT or XP, the network sends the X-ON code to start the paper tape reader automatically after the end of a message is output to the terminal and the message empties the terminal's output queue.

The IN command is not applicable to HASP, 3270, or bisynchronous 2780 and 3780 terminals. 2741 terminals can only select IN=KB, IN=X, and IN=XK. Mode 4 terminals, which are always in block mode, and X.25 PSN terminals cannot select PT or XP mode.

(LI) - LINE FEED IDLE COUNT

The LI command specifies the number of idle characters to be inserted into the output stream following a line feed.

Format:

ct LI=value

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
value	Specifies the new line feed idle count. This value can be from 0 to 127 or the string CA. A value of CA restores the line feed idle count to the default value.

The line feed idle count is similar to the carriage return idle count (CI), except that the idle characters are output after a line feed instead of after the carriage return.

This command only applies to asynchronous terminals.

(LK) - LOCKOUT OF UNSOLICITED MESSAGES

The LK command determines whether unsolicited messages from the network operator appear at your terminal.

Format:

ct LK=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to lock out unsolicited messages. A value of Y locks out the messages and a value of N allows the messages to appear.

(MS) - MESSAGE TO NETWORK OPERATOR

The MS command sends a message to the network operator.

Format:

ct MS=message

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
message	Specifies your message to the operator. You can use any string of 50 characters or less.

If you are having problems with the network (such as difficulty connecting to IAF), you can communicate these problems to the network operator using the MS command.

(OC) - FLOW CONTROL FOR OUTPUT DEVICES

The OC command specifies whether the terminal generates DC3s as a stop signal to the network and DC1s as a resume signal for terminal output.

Format:

ct OC=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) output control.

If the network receives a DC3 while output control is enabled, it suspends output until it receives a DC1.

Output control applies to asynchronous terminals, including paper tape devices (except 2741s).

(OP) - OUTPUT DEVICE SELECTION

The OP command specifies the type of output device.

Format:

ct OP=op

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

<u>Parameter</u>	<u>Description</u>								
op	Specifies the output device. You can specify the following values for op:								
	<table border="0"> <thead> <tr> <th><u>op</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>DI</td> <td>Specifies a display device.</td> </tr> <tr> <td>PR</td> <td>Specifies a printer.</td> </tr> <tr> <td>PT</td> <td>Specifies a paper tape punch.</td> </tr> </tbody> </table>	<u>op</u>	<u>Description</u>	DI	Specifies a display device.	PR	Specifies a printer.	PT	Specifies a paper tape punch.
<u>op</u>	<u>Description</u>								
DI	Specifies a display device.								
PR	Specifies a printer.								
PT	Specifies a paper tape punch.								

You can punch a tape in any mode, but the network does not provide the proper X-OFF characters unless OP has a value of PT.

If you select OP=PR, the network does line folding and page bounding for asynchronous and X.25 PSN terminals. If OP=DI, the network assumes the terminal performs these functions itself.

This command only applies to asynchronous terminals (except 2741s). OP=PT does not apply to X.25 PSN terminals.

(PA) - PARITY PROCESSING

The PA command specifies the parity processing to be performed by the network.

Format:

ct PA=pa

<u>Parameter</u>	<u>Description</u>								
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.								
pa	Specifies the type of parity processing. You can specify the following values for pa:								
	<table border="0"> <thead> <tr> <th><u>pa</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>E</td> <td>Set parity bit for even parity during output, and check for even parity (except for X.25 PSN terminals) and set parity bit to zero on input.</td> </tr> <tr> <td>N</td> <td>In transparent mode, do not set or check the parity bit because it is part of the input or output data. In normalized mode, set the parity bit to zero on input and output.</td> </tr> <tr> <td>O</td> <td>Set parity bit for odd parity during output, and check for odd parity (except for X.25 PSN terminals) and set parity bit to zero on input.</td> </tr> </tbody> </table>	<u>pa</u>	<u>Description</u>	E	Set parity bit for even parity during output, and check for even parity (except for X.25 PSN terminals) and set parity bit to zero on input.	N	In transparent mode, do not set or check the parity bit because it is part of the input or output data. In normalized mode, set the parity bit to zero on input and output.	O	Set parity bit for odd parity during output, and check for odd parity (except for X.25 PSN terminals) and set parity bit to zero on input.
<u>pa</u>	<u>Description</u>								
E	Set parity bit for even parity during output, and check for even parity (except for X.25 PSN terminals) and set parity bit to zero on input.								
N	In transparent mode, do not set or check the parity bit because it is part of the input or output data. In normalized mode, set the parity bit to zero on input and output.								
O	Set parity bit for odd parity during output, and check for odd parity (except for X.25 PSN terminals) and set parity bit to zero on input.								

<u>Parameter</u>	<u>Description</u>
<u>pa</u>	<u>Description</u>
Z	Set parity bit to zero on output and check parity bit against zero on input.
I	In transparent mode, do not set or check the parity bit because it is part of the input or output data. Ignore the parity bit when checking for transparent input mode delimiters. In normalized mode, set the parity bit to zero on input and output.

This command only applies to asynchronous terminals (except 2741s). For X.25 PSN terminals, the parity bit is set to 0 and not checked for input.

(PG) - PAGE WAITING

The PG command enables or disables page waiting at the terminal during output. If you enable page waiting, the terminal stops at the end of each output page for your acknowledgement (a null input line) before output continues. (In order for the PG command to be effective, page length (PL) must be non-zero.)

Format:

ct PG=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) page waiting.

When you enter a null input line to get the next page of output, the null input line has no other meaning.

Page waiting does not apply to bisynchronous 2780 or 3780 terminals or to HASP terminals.

(PL) - PAGE LENGTH

The PL command establishes the maximum number of physical lines that can be printed as one page. If the output device type is PR, the network automatically positions the carriage to the top of the form after PL lines have been output.

Format:

ct PL=pl

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
pl	Specifies the page length. You can specify 0 or any value from 8 to 255. PL=0 selects an infinitely long page.

The PL value must be nonzero if you want to enable page waiting, since page waiting only occurs after the number of lines defined by pl-1 are output.

(PW) - PAGE WIDTH

The PW command establishes the maximum number of characters that the terminal prints on one output line. If the output device type is PR, the system automatically starts a new line after pw characters have been output.

Format:

ct PW=pw

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
pw	Specifies the page width. You can specify a value of 0 or any value from 20 to 255. PW=0 selects an infinitely long line.

(RC) - RESET TERMINAL CHARACTERISTICS

The RC command resets your terminal to the NDL-specified default values. Normally you would use this command after disconnecting from a host connection for which you had altered one or more terminal characteristics.

Format:

ct RC

Note that a terminal can also be automatically reconfigured each time the terminal is disconnected from a host if the terminal is on a line for which automatic reconfiguration has been specified by the network administrator.

(SE) - SPECIAL EDITING MODE

The SE command enables or disables the special editing variant of normalized input mode.

Format:

ct SE=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Enables (Y) or disables (N) special editing operations.

When special editing is in effect, any backspace character entered is stored for transmission to the host. Similarly, a line feed entry produces a normal line feed operation at the terminal and the network stores the line feed character code for transmission to the host. A backspace followed by a line feed causes the system to output a bell.

This command is valid only for asynchronous terminals.

(TC) - TERMINAL CLASS COMMAND

The TC command specifies a terminal class for your terminal. The terminal class associates your terminal with a predefined set of terminal characteristics. To effectively use a terminal class, choose a terminal class whose default terminal definition corresponds to the characteristics of your terminal.

Format:

```
ct TC=tc
```

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
tc	Specifies the terminal class. The value tc can be from 1 to 18.

The following list associates terminal class values with terminal types:

<u>tc</u>	<u>Terminal Type</u>	<u>tc</u>	<u>Terminal Type</u>
1	M33, M35, M37, M38 teletypes	10	200 User Terminal
2	CDC 713-10, 722, 751-1, 752, 756	11	CDC 714-30
3	CDC 721, 722-30	12	CDC 711-10
4	IBM 2741	13	CDC 714-10/20
5	M40 teletypes	14	HASP protocol, preprint format control
6	Hazeltine 2000	15	CDC 734, 731-12
7	Terminals conforming to the ANSI X3.64 standard (DEC VT100, TeleVideo 924, 950, 955)	16	IBM 2780
8	Tektronix 4010, 4014	17	IBM 3780
9	HASP protocol, postprint format control	18	IBM 3270

A terminal that is not shown as belonging to a terminal class may still be operational. The terminal can be assigned to a terminal class having similar characteristics and the terminal's definition can be modified as necessary to correctly define the operational characteristics of the terminal.

If your terminal is on an auto-recognition line and its terminal class is other than 4, 10, 13, or 18, you can use the TC command to change its terminal class to ensure proper operation.

If you include a TC command in a sequence of terminal definition commands, enter the TC command first as it resets all other terminal characteristics.

(TM) - TERMINATING A TERMINAL-HOST CONNECTION

The TM command terminates the connection between your terminal and a host.

Format:

ct TM

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT command.

You can use the TM command if your application does not respond to your input. The system responds to the TM command in the following manner:

```
HOST DISCONNECTED CONTROL CHARACTER=ct
ENTER INPUT TO CONNECT TO HOST
```

(XL) - MULTIMESSAGE TRANSPARENT MODE

The network provides a transparent mode that allows you to remain in transparent input mode even after message blocks have been forwarded to the application. The XL command specifies the multimessage-forwarding signal and the terminator for transparent input mode. If your terminal's parity is set to even, odd, or zero (PA=E, O or Z), the parity bit is stripped (set to zero) before the characters are sent to the host, although no translation occurs. If your terminal's parity is set to none or ignore (PA=N or I), the parity bit is not stripped. See the NAM/CCP Terminal Interfaces Reference manual for more information on transparent input and parity.

Format:

ct XL=Xxx,Ccount,TO,Yyy

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
Xxx	Selects the character with the two-digit hexadecimal representation xx as the message-forwarding signal. If you do not specify a value yy, the occurrence of two successive characters defined by xx acts as a terminator for transparent input mode. When you specify a hexadecimal code above 7F, you must have previously set your terminal's parity to none (PA=N) or ignore (PA=I); otherwise the network cannot detect the code when it occurs in a transparent mode message.
Ccount	Selects a decimal value from 1 to 4095 as a character count that functions as a message-forwarding signal. These values (1 to 4095) are valid with the network. However, unless changed during installation, the maximum logical input line length in IAF (2500 characters) limits you.
TO	Selects timeout as a transparent input mode terminator.
Yyy	Selects the character with the two-digit hexadecimal representation yy as a terminator for transparent input mode when it follows xx. When you specify a hexadecimal code above 7F, you must have previously set your terminal's parity to none (PA=N) or ignore (PA=I); otherwise the network cannot detect the code when it occurs in a transparent mode message.

Yyy, if specified, must follow the Xxx parameter; otherwise, it is ignored. The parameters are otherwise order-independent. You must specify at least one of Xxx or TO to select a transparent mode terminator. When it immediately follows the character defined by Xxx on input, the character defined by yy terminates transparent input mode. The characters defined by Xxx and Yyy can be identical or you can omit Yyy. In either case, two successive input characters defined by Xxx are required to terminate transparent mode.

The message-forwarding signals and mode terminators are not sent to the host.

The XL command applies only to asynchronous and mode 4 terminals. Terminal class 4 is configured with the RETURN key as the only transparent mode delimiter and terminator, and mode 4 terminals are configured with the SEND or ETX key as their only transparent mode delimiter and terminator. For devices connected through X.25, the PAD forwarding signal always acts as a transparent mode delimiter in addition to any other options selected.

This command does not apply to HASP and bisynchronous terminals.

Table J-1. Default Terminal Definitions

Mne- monic	Description	Terminal Types																		
		TTY M33	CDC 713-10, 722,751-1, 752,756	CDC 721, 722-30	IBM 2741	TTY M40-2	HAZ 2000	X3.64 DEC VT100	Tek 4000	X.25 any	HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
TC	Terminal class	1	2	3	4	5	6	7	8	(0)	9	10	11	12	13	14	15	16	17	18
AB	Abort output block	CAN	CAN	CAN	(CAN	CAN	CAN	CAN	na	na	na	na	na	na	na	na	na	na	na
BF	Blocking factor	1	1	1	1	1	1	1	1	(1)	na	1	1	1	1	na	na	na	1	1
BS	Backspace character	BS	BS	BS	BS	none	BS	BS	BS	(1)	na	na	na	na	na	na	na	na	na	na
B1	User break 1	DLE	DLE	DLE	:	DLE	DLE	DLE	DLE	(1)	:	:	:	:	:	:	:	na	na	:(6)
B2	User break 2	DC4	DC4	DC4)	DC4	DC4	DC4	DC4	(1))))))))	na	na)(7)
BR	Break as user break 1 (and cancel character)	N	N	N	na	N	N	N	N	(1)	na	na	na	na	na	na	na	na	na	na
CI	Carriage return idles	2	0	0	8	1	0	0	0	(1)	na	na	na	na	na	na	na	na	na	na
CN	Cancel character	CAN	CAN	CAN	(CAN	CAN	CAN	CAN	(1)	(((((((na	na	(
CP	Cursor positioning	Y	Y	Y	na	Y	Y	Y	Y	N	na	na	na	na	na	na	na	na	na	na
CT	Network control character	ESC	ESC	ESC	%	ESC	ESC	%	ESC	(1)	%	%	%	%	%	%	%	%	%	%
DL	Single-message trans- parent input mode																			
	Character delimiter	CR	CR	CR	NL	CR	CR	CR	CR	(1)	na	(4)	(4)	(4)	(4)	na	(4)	na	na	na
	Character count delimiter	2043	2043	2043	na	2043	2043	2043	2043	(1)	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	No	No	No	na	No	No	No	No	(1)	na	na	na	na	na	na	na	na	na	na
EB	End-of-block																			
	Character	EOT	EOT	EOT	na	EOT	EOT	EOT	EOT	(2)	na	(4)	(4)	(4)	(4)	na	(4)	na	na	na
	Cursor positioning	CL	CL	CL	na	CL	CL	CL	CL	No	na	No	No	No	No	na	No	na	na	na
EL	End-of-logical line																			
	Character	CR	CR	CR	NL	CR	CR	CR	CR	(1)	na	(5)	(5)	(5)	(5)	na	(5)	na	na	na
	Cursor positioning	LF	LF	LF	na	LF	LF	LF	LF	No	na	No	No	No	No	na	No	na	na	na
EP	Echoplex mode	N	N	N	na	N	N	Y	N	na	na	na	na	na	na	na	na	na	na	na
FA	Full ASCII input mode	N	N	N	N	N	N	N	N	N	na	N	N	N	N	na	N	na	na	na
IC	Input device flow control	N	N	N	na	N	N	Y	N	na	na	na	na	na	na	na	na	na	na	na
IN	Input device	KB	KB	KB	KB	KB	KB	KB	KB	BK	na	BK	BK	BK	BK	na	BK	na	na	KB
LI	Line feed idles	1	0	0	1	3	3	0	0	(1)	na	na	na	na	na	na	na	na	na	na
LK	Lockout NOP messages	N	N	N	N	N	N	N	N	(1)	No	No	No	No	No	No	No	na	na	N
OC	Output device flow control	N	N	N	na	N	N	Y	N	na	na	na	na	na	na	na	na	na	na	na
OP	Output device	PR	DI	DI	PR	DI	DI	DI	DI	(1)	na	na	na	na	na	na	na	na	na	na
PA	Parity processing	E	E	E	na	E	E	E	E	(3)	na	na	na	na	na	na	na	na	na	na
PG	Page waiting	N	N	N	N	N	N	N	N	(1)	na	Y	Y	Y	Y	na	Y	na	na	Y
PL	Page length	0	24	30	0	24	27	24	35	(1)	0	13	16	16	16	0	13	0	0	24
PW	Page width	72	80	80	132	80	74	80	74	(1)	80	80	80	80	80	80	80	80	120	80
SE	Special editing mode	N	N	N	N	N	N	N	N	na	na	na	na	na	na	na	na	na	na	na
XL	Multimessage transparent input mode	No	No	No	No	No	No	No	No	No	na	No	No	No	No	na	No	na	na	na

(0) Terminals belong to terminal classes 1 through 3 or 5 through 8.
(1) Same as for the terminal if it were not connected via a packet-switching network.
(2) End-of-packet sequence (M bit is reset to zero).
(3) The parity bit is ignored on input and is generated for output as for the basic terminal type.
(4) ETX (resulting from the SEND key).
(5) ESC A (resulting from the CARRIAGE RETURN key).
(6) Program attention key 1 is a valid user break 1.
(7) Program attention key 2 is a valid user break 2.

Table J-2. Parameter Ranges for Terminal Definition Commands (Sheet 1 of 2)

Mnemonic	Description	Terminal Types																		
		TTY M33	CDC 713-10, 722,751-1, 752,756	CDC 721, 722-30	IBM 2741	TTY M40-2	HAZ 2000	X3.64 DEC VT100	Tek 4000	X.25 any	HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
TC	Terminal class	1	2	3	4	5	6	7	8	(0)	9	10	11	12	13	14	15	16	17	18
AB	Abort output block	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	na	na	na	na	na	na	na	na	na
BF	Blocking factor	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	(2)	na	0-2	0-2	0-2	0-2	na	0-2	na	na	0-2
BS	Backspace character	(1)	(1)	(1)	(1)	(1),(7)	(1)	(1)	(1)	(2)	na	na	na	na	na	na	na	na	na	na
B1	User break 1	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
B2	User break 2	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
BR	Break as user break 1 (and cancel character)	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	na	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	(2) na	na	na	na	na	na	na	na	na	na	na
CI	Carriage return idles	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	(2)	na	na	na	na	na	na	na	na	na	na
CN	Cancel character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
CP	Cursor positioning	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	(2)	na	na	na	na	na	na	na	na	na	na
CT	Network control character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
DL	Single-message transparent input mode																			
	Character delimiter	0-FF	0-FF	0-FF	6D	0-FF	0-FF	0-FF	0-FF	(2)	na	03	03	03	03	na	03	na	na	na
	Character count delimiter	1-4095	1-4095	na	na	1-4095	1-4095	1-4095	1-4095	(2)	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	TO	TO	TO	na	TO	TO	TO	TO	(2)	na	na	na	na	na	na	na	na	na	na
EB	End-of-block																			
	Character	(1), EL, EB	(1), EL, EB	(1), EL, EB	na	(1), EL, EB	(1), EL, EB	(1), EL, EB	(1), EL, EB	(2)	na	EB	EB	EB	EB	na	EB	na	na	na
	Cursor positioning	(3)	(3)	(3)	na	(3)	(3)	(3)	(3)	(2)	na	No	No	No	No	na	No	na	na	na
EL	End-of-logical line																			
	Character	(1), EL, EB	(1), EL, EB	(1), EL, EB	na	(1), EL, EB	(1), EL, EB	(1), EL, EB	(1), EL, EB	(2)	na	EB, EL	EB, EL	EB, EL	EB, EL	na	EB, EL	na	na	na
	Cursor positioning	(3)	(3)	(3)	na	(3)	(3)	(3)	(3)	No	na	No	No	No	No	na	No	na	na	na

(0) Terminals belong to terminal classes 1 through 3 or 5 through 8.
(1) Any character from the ASCII 128-character set except any lowercase or uppercase alphabetic character, any digit 0 through 9, NULL, SOH, STX, =, DEL, or space.
(2) Same as for the terminal if it were not connected by a packet-switching network.
(3) CR, LF, CL, NO.
(4) KB, PT, BK, X, XK, XP.
(5) PR, DI, PT.
(6) Same as for the terminal if it were not connected by a packet-switching network. Parity is ignored on input and is generated for output.
(7) ASCII BS character code (8 hexadecimal) cannot be used.

Table J-2. Parameter Ranges for Terminal Definition Commands (Sheet 2 of 2)

Mnemonic	Description	Terminal Types																	
		TTY M33	CDC 713-10, 722,751-1, 752,756	CDC 721, 722-30	IBM 2741	TTY M40-2	HAZ 2000	X3.64 DEC VT100	Tek 4000	X.25 any	HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780
EP	Echoplex mode	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na
FA	Full ASCII input mode	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)	na	Y,N	Y,N	Y,N	Y,N	na	Y,N	na	na
IC	Input device flow control	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na
IN	Input device	(4)	(4)	(4)	KB,XX, X	(4)	(4)	(4)	(4)	X	na	X	X	X	X	na	X	na	na
LI	Line feed idles	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	(2)	na	na	na	na	na	na	na	na	na
LK	Lockout messages	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	na	na
OC	Output device flow control	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na
OP	Output device	(5)	(5)	(5)	PR, DI	(5)	(5)	(5)	(5)	PR, DI	na	na	na	na	na	na	na	na	na
PA	Parity processing	Z,0, E,N	Z,0, E,N	Z,0, E,N	na	Z,0, E,N	Z,0, E,N	Z,0, E,N	Z,0, E,N	(6)	na	na	na	na	na	na	na	na	na
PG	Page waiting	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)	na	Y,N	Y,N	Y,N	Y,N	na	Y,N	na	na
PL	Page length	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	(2)	0	0,8-255	0,8-255	0,8-255	0,8-255	0	0,8-255	0	0
PW	Page width	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	(2)	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255
SE	Special editing mode	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na
TC	Terminal class input	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	4	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	9	10,15	11-13	11-13	11-13	14	10,15	16	17
XL	Multimessage transparent input mode																		
	Character delimiter	0-FF	0-FF	0-FF	2D	0-FF	0-FF	0-FF	0-FF	(2)	na	03	03	03	03	na	03	na	na
	Character terminator	0-FF	0-FF	0-FF	2D	0-FF	0-FF	0-FF	0-FF	(2)	na	na	na	na	na	na	na	na	na
	Timeout delimiter	TO	TO	TO	na	TO	TO	TO	TO	na	na	na	na	na	na	na	na	na	na
	Character count delimiter	1-4095	1-4095	1-4095	na	1-4095	1-4095	1-4095	1-4095	(2)	na	na	na	na	na	na	na	na	na

- (0) Terminals belong to terminal classes 1 through 3 or 5 through 8.
- (1) Any character from the ASCII 128-character set except any lowercase or uppercase alphabetic character, any digit 0 through 9, NULL, SOH, STX, =, DEL, or space.
- (2) Same as for the terminal if it were not connected by a packet-switching network.
- (3) CR, LF, CL, NO.
- (4) KB, PT, BK, X, XK, XP.
- (5) PR, DI, PT.
- (6) Same as for the terminal if it were not connected by a packet-switching network. Parity is ignored on input and is generated for output.
- (7) ASCII BS character code (8 hexadecimal) cannot be used.

You can transmit input to an application in normalized mode (also called character mode) or transparent mode (also called binary mode). Appendix J describes how you delimit and transmit input in transparent mode (refer to the DL and XL commands), however the default input mode for all supported terminal classes is normalized mode. The remainder of this appendix describes how to delimit and transmit terminal input in normalized mode under the default conditions of the various terminal classes. (Appendix J also describes how you can change these defaults).

PHYSICAL END-OF-LINE (LINE FEED)

In terminal classes 1 through 8, you can terminate a physical line of input (block) with a line feed key. The network also terminates a physical line of output whenever the page width of a terminal is reached and OP=PR (refer to the PW and OP command in appendix J). The network may respond to the line feed key by advancing the cursor or carriage to the beginning of the next line (refer to the CP command in appendix J).

LOGICAL END-OF-LINE (CARRIAGE RETURN)

You terminate a logical line of input with a carriage return key. A logical line can consist of one or more physical lines but not vice versa. The carriage return key advances the cursor or carriage to the beginning of the next line.

MESSAGE TRANSMISSION

You transmit your input to the network with the message transmission key. For terminal classes 1 through 8 operating in character or line mode (IN=KB), the message transmission key is the carriage return key. Hence, for these terminal classes, logical lines are transmitted separately. When the delimiter for the logical end-of-line and the transmission key are different (as with block mode devices, including PSN and mode 4 terminals), the terminal stores logical lines until you press the transmission key. The terminal then sends the single transmission consisting of multiple logical lines to the network. The network in turn sends each logical line to the application separately. The message transmission key always acts as a carriage return if not immediately preceded by one.

Table K-1 shows the keys used by the various terminal classes to perform the three functions just described.

Table K-1. Default Message Delimiters and Transmission Keys

Function			
Terminal Class	Physical End-of-line (line feed)	Logical End-of-line (carriage return)	Message Transmission Key†
1	LINE FEED	RETURN	RETURN
2	↓	RETURN	RETURN
3	↓	NEXT	NEXT
4	ATTN	RETURN	RETURN
5	NEW LINE	RETURN	RETURN
6	LF	CR	CR
7	LINE FEED	RETURN	RETURN
8	LF	RETURN	RETURN
9	Varies††	Varies††	Varies††
10	None	RETURN	SEND
11	None	RETURN	SEND
12	None	NEW LINE	ETX
13	None	NEW LINE	ETX
14	Varies††	Varies††	Varies††
15	None	NEW LINE	SEND
16	None	EM	EOT
17	None	EM	EOT
18	None	Enter	Enter

†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.

PASSIVE PROCEDURES

L

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 varieties. Section 4 describes parameter-prompting procedures in detail.

.PROC DIRECTIVE

This format of the .PROC directive does not provide for parameter prompting nor will the system give parameter descriptions or other help.

Format:

.PROC, pname, p1, p2, ..., pn.

<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.
Pi	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=#PRIMARY	(CDC graphics: keyword= ≡ PRIMARY)	FILE1=#PRIMARY
keyword=#FILE	(CDC graphics: keyword= ≡ FILE)	FILE1=#FILE
keyword	A 1- to 10-character keyword.	
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. This option is similar to *K on interactive procedures.	

<u>Parameters</u>	<u>Description</u>
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. This option is similar to *N or *D on interactive procedures.
default1 and default2 could be any 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. If the procedure resides on a library and is called by a name call form of the BEGIN command, keyword is replaced by a null value.
#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).
#PRIMARY	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 refers to the job's primary file (refer to File Types in section 2).

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 L-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 L-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 L-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 L-1. Parameter Substitution In Order-Dependent Mode

		BEGIN Command Parameter Format			
		omitted	keyword	value	\$value\$
Procedure Header Parameter Format	keyword	keyword	keyword	value	value
	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.
IF,FILE(FNAME,AS),PREP1.
REWIND,FNAME.
REVERT.FNAME PREPARED.
ENDIF,PREP1.
ATTACH,FNAME/#M=M,NA.
IF,FILE(FNAME,.NOT.AS),PREP2.
GET,FNAME/NA.
IF,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.IF,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.IF,FILE(TEST,.NOT.AS),PREP2.
08.26.46.GET,TEST/NA.
08.26.47.IF,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 L-1. The resultant dayfile is shown on the right side of the figure.

<pre>GET,PROGRAM1. BEGIN,EXECUTE,PFILE1,PROGRAM1,PRINT. PFILE1 .PROC,EXECUTE,NAME,OUT. FTN5,I=NAME,L=OUT. LGO. IF,EF=0,DROP. BEGIN,LISTING,PFILE2,OUT. ENDIF,DROP. PFILE2 .PROC,LISTING,OUTFILE=OUT. REWIND,OUTFILE. COPYSBF,OUTFILE,OUTPUT.</pre>	<pre>RESULTANT DAYFILE 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.IF,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</pre>
---	---

Figure L-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 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 L-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 L-2 with the following examples to clarify the meaning of the table entries (keyword, default1, default2, value, and null).

Examples of Parameter Matching:

Procedure on Default

<u>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 L-2. Parameter Substitution In Order-Independent Mode

		BEGIN Command Parameter Format				
		omitted	keyword	keyword=	keyword=value	value or \$value\$†
Procedure Header	keyword	keyword	keyword	null	value	error
Parameter Format	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=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 L-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 L-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.IF,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.
IF,\$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.IF,\$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.	
IF,FILE(FNAME,AS),PREP1.	
REWIND,FNAME.	
REVERT. FNAME PREPARED.	
ENDIF,PREP1.	
ATTACH,FNAME/#M=M,NA.	
IF,FILE(FNAME,.NOT.AS),PREP2.	
GET,FNAME/NA.	
IF,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 L-2. Keyword Substitution in Nested Procedures

PAPER TAPE OPERATIONS

M

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>
ⒸR	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 ⒸR LF RO(3)

DATA LINE

? 12.44,18.31,29.08 ⒸR 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 ⒸR 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 J 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=1fn if a local file other than the primary file is to be punched. Do not press Ⓞ.
4. Turn the punch on.
5. Run a leader of rubouts or blank tape.
6. Press Ⓞ. 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 ?.</p>
<pre>? 5 1 2 6 24 120</pre>	<p>You enter a 5 and receive the desired printout.</p>
<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	RO(30)		
NEW, DEMO	Ⓞ LF	CTRL/X-OFF	RO(3)
100 PROGRAM DEMO	Ⓞ LF	CTRL/X-OFF	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)
1111122222333334444455555	Ⓞ 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)
1111122222333334444455555	Ⓞ 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 ← The seven lines of the program are read without interruption.
 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

RUN ← After the command to run the program is read, the reader is turned off.

? 1111122222333334444455555 ← 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 ← These three lines of modification are read in from the tape without interruption.
 137 II(I)=II(I)+4444
 139 1 CONTINUE

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

N

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

Device Code	Device	PRUs	Block Size	
			CM Words	Characters
DB-n	885-42 Disk Storage Subsystem (full-track; 1<n<3)	n*640	n*40960	n*409600
DC-n	895-1/2 Disk Storage Subsystem (full-track; 1<n<2)	n*704	n*45056	n*450560
DD-n	834 Disk Storage Subsystem (full-track; 1<n<8)	n*160	n*10240	n*102400
DE	Extended Memory	16	1024	10240
DF-n	887 Disk Storage Subsystem (full-track; 1<n<3; 4K sector)	n*608	n*38912	n*389120
DG-n	836 Disk Storage Subsystem (full-track; 1<n<3)	n*564	n*36096	n*360960
DH-n	887 Disk Storage Subsystem (full-track; 1<n<2; 16K sector)	n*704	n*45056	n*450560
DI-n	844-21 Disk Storage Subsystem (half-track; 1<n<8)†	n*107	n*6848	n*68480
DJ-n	844-4x Disk Storage Subsystem (half-track; 1<n<8)	n*227	n*14528	n*145280
DK-n	844-21 Disk Storage Subsystem (full-track; 1<n<8)	n*112	n*7168	n*71680
DL-n	844-4x Disk Storage Subsystem (full-track; 1<n<8)	n*227	n*14528	n*145280
DM-n	885 Disk Storage Subsystem (half-track; 1<n<3)	n*640	n*40960	n*409600
DN	9853 Disk Storage Subsystem (full-track; 2K sector)	1120	71680	716800
DP	Distributive Data Path to Extended Memory	16	1024	10240

†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.

<u>Device Code</u>	<u>Device</u>	<u>PRUs</u>	<u>Block Size</u>	
			<u>CM Words</u>	<u>Characters</u>
DQ-n	885-11/12 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.

INTERACTIVE TRANSFER FACILITY (ITF)

○

ITF allows you to connect your interactive terminal to a remote CYBER 200 computer system linked to your host mainframe by a loosely coupled network (LCN). If you want to submit batch jobs or want to retrieve or otherwise manipulate permanent files, you can use the Interactive Facility (IAF) (refer to Submitting Jobs to Remote hosts in section 3).

SELECTING THE ITF APPLICATION

You select the ITF application in the same manner as you would any Network Access Method (NAM) application, such as IAF or RBF. You specify ITF for the application name during login to NOS. If you are logged into IAF, you can switch to ITF by specifying ITF on the BYE, GOODBYE, HELLO, LOGIN, or LOGOUT command.

SELECTING A REMOTE CYBER 200 SYSTEM

Once you select ITF, you receive an ITF banner at your terminal. The following is an example of the two-line ITF banner:

```
ITF 1.0 - psr level
TERMINAL T103, CONNECTION 12
```

Your site may automatically connect you to a remote host. If you are not automatically connected to a remote host, ITF prompts you for the logical identifier (LID) of the remote host you want to access.

The LID is a three-character name your site has associated with that remote host. The prompt has the following form:

```
ENTER LID (OR ?):
```

In response to this prompt, enter one of the following:

<u>Entry</u>	<u>Description</u>
lid	Specifies the three-character LID your site has associated with the remote host you are attempting to reach.
?	Requests a list of acceptable entries.
carriage return	Selects the default LID as defined by your site.

If the connection to the remote host cannot be established (for example, you enter an invalid LID), ITF so informs you. Unless otherwise directed by your site, ITF will then prompt you again for a LID. Your site may automatically switch you to another application, such as IAF, instead of prompting you for another LID.

If ITF establishes a connection to the remote host, it displays the following status message:

ITF, CONNECTED TO HOST pid ON ACN n/TCN m

where pid is the physical identifier (PID) of the remote host and n and m are connection numbers used by the system. Sometimes the PID is identical to the LID you specify.

PROCESSING YOUR JOB ON THE REMOTE CYBER 200

Once your terminal is under the control of the remote host, it gives you the following login prompt:

ENTER CYBER 200 LOGON

You can then log in and proceed with your interactive job. All of your terminal input is transmitted to the remote host and all data from the remote host is displayed at your terminal. You have the following limitations.

- You cannot use transparent input mode (that is, do not enter the IN=X, IN=XK, or IN=XP terminal definition command).
- Input lines longer than 999 characters are truncated without a comment.
- The termination sequence (user break 2) disconnects your terminal from the remote host.

To terminate your CYBER 200 job and disconnect from the remote host, enter the \$BYE command.

TERMINATING YOUR ITF SESSION

Your site may automatically connect you to another application, such as IAF, any time you disconnect from the remote host. If not, your terminal returns to the control of ITF. ITF then prompts you for another LID. To end your ITF session, enter one of the following:

<u>Entry</u>	<u>Description</u>
BYE	Ends your ITF session and logs you out of NOS.
BYE, application	Ends your ITF session and switches you to the specified application. The following are some valid applications:

<u>application</u>	<u>Description</u>
IAF	Interactive Facility
MCS	Message Control System
PLATO	PLATO-NAM Interface
RBF	Remote Batch Facility
TAF	Transaction Facility
TVF	Terminal Verification Facility

SAMPLE ITF SESSION

The following is an example of an ITF terminal session: B2C is the LID of the remote CYBER 200 computer system:

WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198X.

YY/MM/DD. HH.MM.SS. T19A11
CDC NETWORK OPERATING SYSTEM NOS 2
FAMILY: ,usernam,passwd,itf
ITF 1.0 - psr level
Terminal T19A11, Connection 2

ENTER LID (OR ?): b2c
[ITF, connected to host B2C on ACN 1/TCN 2.1
PLEASE ENTER CY200 LOGONlogon 111111 a 99999999
OS 2.1 RSYS21L1 VSYS21L1 I 09837.0412 ACTIVE NONE
NIL
WARNING***ATTACHED POOLSfiles.
NAME D OWNER SUF ACS TYPE DT FC BT RT FO ORI.DATE LEN UNIT
12FILES *LOCAL A XR PD MS S C U S YY/MM/DD 00228 OE
ALL DONE\$bye
BYE
[ITF, terminal connection ended by host B2C.]
ENTER LID (OR ?):bye
ITF, CONNECT TIME 00.01.15.
LOGGED OUT.

FILE TRANSFERS USING XMODEM

P

XMODEM is an industry standard communications protocol used to transfer files between CYBER mainframes and microcomputers. With XMODEM you can upload and download binary files as well as ASCII text files. To use XMODEM, the microcomputer communication software must use the Christensen protocol.

XMODEM COMMAND

The XMODEM command initiates the file transfer between NOS and microcomputers. Certain parameters must be specified on the XMODEM command or XMODEM prompts for them. Other parameters, if not specified, are given default values.

You can define a configuration file (described later in this appendix) to change the default values for many of the command parameters as well as terminal characteristics. Some of the default values normally used by XMODEM can be superseded by values specified in your configuration file. If a value is specified in the command and in the configuration file, the command value supersedes the configuration file value.

Format:

XMODEM, FN=fn, TD=td, FT=ft, LF=lf, SP=sp, EC=ec, FM=fm, CF=cf.

or

XMODEM, fn, td, ft, lf, sp, ec, fm, cf.

<u>Parameter</u>	<u>Description</u>
FN=fn	Name of the file to be transferred. The file to be transferred from the CYBER to the micro may be local or permanent. If the file is not specified, XMODEM prompts for the file name.
TD=td	Transfer direction. If TD=S is specified, the file is sent from the CYBER to the micro. If TD=R is specified, the file is sent from the micro to the CYBER. If this parameter is not specified, XMODEM prompts for it.
FT=ft	File type. If this parameter is not specified, XMODEM prompts for it. A Specifies a 6/12-bit display code text file on the CYBER to or from 8-bit ASCII on the micro. C Specifies a 6-bit display code text file on the CYBER to or from 8-bit ASCII on the micro. E Specifies a 7-bit ASCII code text file on the CYBER to or from 8-bit ASCII code on the micro. T Specifies a text file. XMODEM uses the current character code set mode of your CYBER job.

<u>Parameter</u>	<u>Description</u>
B	Specifies a CYBER binary file.
M	Specifies a micro binary file.
S	Specifies automatic selection (receive only). Special characters in the first block determines the file type.
LF=lf	This parameter determines whether an end-of-line on text files sent to the micro is transmitted as carriage return and line feed (LF=YES) or only as a carriage return (LF=NO). Default is LF=YES. The default value may be changed with the configuration file (refer to Configuration File in this appendix).
SP=sp	This parameter determines whether or not the first block transmitted is a special file type block. The default is SP=NO. If SP=YES is specified, the special block is transmitted. The default may be changed by the configuration file. For more information on the SP parameter, refer to Limitations and Considerations in this appendix.
EC=ec	Error checking (CRC or CHECKSM). The receiver determines the error checking mode. If TD=S is specified, the micro determines whether to use CRC or CHECKSM. Default is EC=CRC.
FM=fm	File markers. FM may be set to NOS, CPM, or MSDOS. FM determines which file markers are recognized or transmitted in text files by the CYBER. This parameter has no effect on the binary files. If FM=NOS is specified, #EOR, #EOF, and #EOI are recognized or transmitted to indicate end-of-record, end-of-file, and end-of-information, respectively. If used, a four-character file marker must appear as the first and only marker on the line. If FM=CPM or FM=MSDOS is specified, end-of-records and end-of-files are ignored on the text files transmitted from a CYBER to a micro. The end-of-information is transmitted as a CTRL/Z character. On files received from a micro, XMODEM recognizes a CTRL/Z character as the end-of-information. #EOR, #EOF, and #EOI are still recognized. FM=NOS is the default. The default may be changed with the configuration file (refer to Configuration File in this appendix).
CF=cf	Configuration file. This optional parameter specifies the name of a configuration file. The specified configuration file restores your terminal characteristics after an XMODEM file transfer. When the CF parameter is used, XMODEM accepts the values for certain command parameters and terminal characteristics from the specified configuration file. The format and contents of the file are discussed in the Configuration File section of this appendix. The configuration file may be local or permanent. Default is CF-PFTFCFG.

USING XMODEM IN PROMPTING MODE

Type in the command:

XMODEM

You will receive the prompt:

HOST TO SEND (S) OR RECEIVE (R) A FILE?

Enter S to send a file from the CYBER to a micro, or R to receive a file from a micro. You may enter the entire words SEND or RECEIVE.

SENDING A FILE FROM A CYBER

If you selected the transfer direction of SEND, the next prompt will be:

PLEASE ENTER THE FILE NAME.

The file you specify must be a local or a permanent disk file. If the file is empty or cannot be read, the file name prompt is repeated.

XMODEM will then ask:

Is the file -

- B - Cyber binary
- M - Micro binary
- T - Text -- Display code (upper case only)
- A - Text -- 6/12 Display code ASCII (upper/lower case)
- E - Text -- 8/12 ASCII

After you select a valid option, XMODEM will print the approximate number of blocks to be transferred and indicate that it is ready to start the transfer. At this point, you should enter the appropriate commands on the micro to initiate the file transfer.

XMODEM on the CYBER will acknowledge an ASCII NAK, a capital C, or an ASCII CAN (CTRL/X). If a NAK is received by the CYBER, the file transfer takes place using the checksum method of error checking. If a capital C is received, the error checking is done using CRC method of error checking. If a CTRL/X is received, the transfer aborts, and you are back at the NOS command prompt.

If 10 characters are received and none are among the expected three, the transfer aborts with the error message:

NO INITIAL NAK RECEIVED.

If a CTRL/X is received at any time during the transfer, the transfer aborts.

RECEIVING A FILE FROM A MICRO

If you elected to receive a file from a micro, XMODEM prompts:

PLEASE ENTER THE FILE NAME.

Enter the local file name under which you wish to have the file stored on NOS. If the file name is invalid or you cannot write to it, XMODEM repeats the prompt.

Next, you are asked:

Is the file -

- B - Cyber binary
- M - Micro binary
- T - Text -- Display code (upper case only)
- A - Text -- 6/12 Display code ASCII (upper/lower case)
- E - Text -- 8/12 ASCII
- S - Auto-select file type

After you select a valid option, you should enter the micro commands necessary to start the transfer. XMODEM on the CYBER sends out an ASCII NAK or capital C (depending on the value set for the command parameter EC) until it either receives a block from the micro or has tried 10 times to initiate the transfer. After 10 tries, XMODEM aborts.

If a CTRL/X (ASCII CAN) is received as the first character in a block, XMODEM aborts, and you are back at the NOS command prompt.

CONFIGURATION FILE

XMODEM can accept the values of some parameters from a configuration file. With this file, you can change the default values for parameters LF, SP, EC, and FM. You can also reset certain terminal characteristics.

These are the rules for the configuration file entries:

- One parameter entry per line is allowed.
- Uppercase and lowercase characters may be used.
- Blanks are not significant.
- Lines beginning with an asterisk (*) are considered as comment lines.
- Each line may optionally be terminated by a period.
- Comments may follow the period on the same line.

The following are the parameters you can specify:

<u>Keyword</u>	<u>Possible values</u>	<u>Description</u>
LF	YES, NO, Y, N	Line feed issued after carriage return. Default is YES.
SP	YES, NO, Y, N	Special file type block issued. Default is NO.
EC	CRC, CHECKSM, CR, CM	Error checking. Maximum parameter length is 7 characters. Default is CRC.
FM	NOS, CPM, MSDOS, N, C, M	File marker type. Default is NOS.
PA	NONE, EVEN, ODD, ZERO, N, E, O, Z	Value to which parity will be reset upon termination of XMODEM file transfer. Default is NONE.
EP	YES, NO, Y, N	Echoplex value set at termination of file transfer. Default is NO.
IC	YES, NO, Y, N	Input flow control reset value. Default is YES.
OC	YES, NO, Y, N	Output flow control reset value. Default is YES.
AC	decimal number	Attention character (affects CDCNET terminals only). Refer to the CDCNET Terminal Interface Usage Manual.

Example:

```
EC = CRC.      Set error checking.  
SP = NO.  
LF = Y.  
EP = N.  
OC = Y.
```

FILE TRANSFER NOTES

There are three types of files that can be transferred: text files, CYBER binary files, and micro binary files. The processing of file markers and carriage control indicators (line feed, for example) depends on the type of file being transferred and the direction of transfer.

TEXT FILES

Text files have either an ASCII carriage return or a carriage return and a line feed appended to the end of each line when transferred from a CYBER to a micro.

When a file is transferred from a micro to a CYBER, carriage returns are translated into zero byte terminators, and an ASCII line feed is ignored if it immediately follows a carriage return. If the FM=CPM or FM=MSDOS parameter is specified, XMODEM recognizes a CTRL/Z character as the end-of-information and also recognizes #EOR, #EOF, and #EOI file markers. If FM=NOS is specified, XMODEM recognizes or transmits #EORs, #EOFs, and #EOIs. To be recognized, a 4-character file marker must appear as the first and only characters on a line.

If a file received by the CYBER is transferred as a display code text file (uppercase only), lowercase alphabetic characters are translated as uppercase. Untranslatable characters are set to asterisks (*), except for the ASCII NULL character, which is ignored.

CYBER BINARY

A CYBER binary file is transferred eight bits at a time. The file is blocked during transmission, with the first character being the length of the block or an EOR/EOF/EOI marker. If an EOR/EOF marker is found, the next character is again a block length or an EOR, EOF, or EOI. The maximum block size is 240 (decimal). A value of 253 indicates an EOR, 254 indicates an EOF, and 251 indicates an EOI. Only files that are transferred from a CYBER to a micro as CYBER binary files can be transferred from a micro to a CYBER as CYBER binary files, otherwise unpredictable results can occur.

Any data received by the CYBER after the EOI marker is received is acknowledged but discarded.

MICRO BINARY

A micro binary file received by the CYBER is stored as 8/12-bit ASCII. Each 8-bit ASCII character received is stored on the CYBER as a 12-bit ASCII character with the high order bit set. When the file is transmitted back to the micro, any zero character is ignored. Thus, a NULL character must be represented as 4000B. The high order bit is optional on all other characters. A file processed by FCOPY meets these requirements, except that end of lines (since they are zero) are ignored. This method is necessary to guarantee that the file is sent back to the micro exactly as it was received.

LIMITATIONS AND CONSIDERATIONS

The following are general tips on using XMODEM:

- Since the XMODEM protocol is an 8-bit protocol, parity must be set to NONE. XMODEM internally issues a TRMDEF,PA=N. However, data networks such as local area networks may force a specific parity. Such networks must be configured for transparent data transmission. If they are configured to force parity other than NONE, XMODEM will not work. The communications path must be capable of transmitting all 256 ASCII characters.
- Echoplex must be set to OFF (EP=N). If EP=Y is set, each character sent to the CYBER is echoed back, and the micro cannot distinguish between echoed data and real data transmitted by the CYBER. XMODEM internally turns echoplex off. As there is no way for XMODEM to determine the current setting of echoplex, it normally leaves it off after the transmission. This can be altered by using the configuration file (described earlier). Some data networks may echo data themselves, rather than having the CYBER front end echo the data. XMODEM cannot disable this echoing. You must disable the echoing.
- Some micros, such as the APPLE,[†] require the high order bit in each character to be set for some files. This is also true of some word processors, such as WORDSTAR. If the file is transferred as a 6/12-bit display code text file to the CYBER, the high order bit is lost because these character sets do not support all 256 ASCII characters. To get around this problem, you can either write a program on the micro to restore the high order bit, or you can transfer the file to the CYBER as an 8/12-bit ASCII file, or as a micro binary file.
- If the SP parameter is set to YES, the first block transmitted from the CYBER to the micro is of this form:

```
SOH+200B/filetype/255D-filetype/123D/zeros
```

Where SOH is an ASCII SOH, and file type is

```
00 - Text  
04 - Micro binary  
08 - Cyber binary
```

The 123 is included to ensure that the checksum does not match so that micros which do not recognize this block will respond with a NAK (Negative Acknowledge). This block is recognized by ASCII Express^{††} for the Apple. If the special block is not supported by your micro's transfer program, it will be considered an invalid block and will be NAKed. If this block is NAKed three times, the transfer will proceed with the data.

If the first block received by the CYBER is of this form and the file type was set to auto-select, the file type in the block is used to determine the file type used for the transfer. If an invalid file type is detected, micro binary file is used. If the auto-select mode is set and no special file type block is received by the CYBER, a file type of TEXT with 6/12-bit display code is used.

[†]A registered trademark of Apple Computer, Inc.

^{††}A registered trademark of United Software Industries, Inc.

- Although the CYBER implementation supports most common extensions to the XMODEM protocol, it does not support batch mode. If you try to transfer files using batch mode, the transfer terminates.
- As input timeout has not been implemented in NOS, XMODEM does not timeout after 10 seconds and does retransmit the last block. The protocol requires the receiver to timeout, but the sender may not always timeout. This causes problems only when the micro is the sender and the micro does not timeout. Most XMODEM packages available for micros timeout in both directions; so this limitation will not normally cause any problems.

INDEX

- AAM (See Advanced Access Method)
AB command J-5
Abbreviated job status command 8-17
Abort C-1
Abort output character 16-5; J-5
Aborting output 16-5
ABS record type 15-3,6
Absolute binary card output F-7
ACCESS command 8-12
Access level C-1
Access level, equipment C-4
Access subsystem 8-12,20.1,37
Account block 3-11; 7-4,59,63; C-1
Account dayfile 3-18; 7-4
Accounting for system resources 3-8; 7-4
Accumulated central processor time
3-18; 7-10,17
ACF record type 15-3
Adder activity 3-18
Address out of range error 6-23
Advanced Access Method (AAM) 1-6
AEMP (See Matrix Array Processor (MAP),
resource usage)
AESR (See System Resource Unit (SRU))
AFTER directive 15-27
Allocatable device C-1
Alphanumeric C-1
ALTER command 8-42
Alternate storage 2-15,16
ANSI labeled tapes 2-8; 12-9,13,19; G-1
ANSI X3.64 standard terminal 16-1; J-25
APPEND command 10-7
APPEND mode 10-8
Appending files 10-7
Application C-1
Application, resource usage 3-18
Application switching 3-5; 8-18,19,21;
C-1; J-2
APPSW command 8-18,19
AR command J-5
ARG= entry point 15-43
Argument C-1
ASCFL code 9-23
ASCII C-1
ASCII code set 9-23; A-1,12
ASCII command 8-2; A-3
ASCII graphic character sets A-1
ASCII graphic 63-character set A-1,2
ASCII graphic 63/64-character set 9-46; A-1
ASCII graphic 64-character set 9-48;
A-1,2,3
ASCII graphic 95-character set 9-46,48;
A-1,2,3
ASCII mode 8-2,4; A-3; E-1
ASCII 128-character set A-1,3
ASCII, 7-bit code 9-23; A-1
ASCII63 code 9-23
ASCII64 code 9-23
ASCII8 code 9-23
ASCII88 code 9-23
Assembler language 1-5
ASSIGN command 2-15; 9-2; 12-5
Asynchronous terminal output 16-5
Asynchronous terminals J-4
ATTACH command 10-8
ATTN key, IBM 2741 terminal 16-8
AUTO command 8-3
Auto-drop status 9-56
Auto eject mode H-2
Auto mode 8-3
Automatic line spacing H-2
Auxiliary device 2-15,16; 10-4,35; C-1

B format tapes 9-59,64; I-1
Backspace character 16-3; J-7
BAM (See Basic Access Method)
Banner page 9-54,55; D-2
BASIC C-1
Basic Access Method (BAM) 1-6
BASIC command 8-12
BASIC subsystem 8-12
BATCH command 8-13
Batch job 3-1
Batch subsystem 8-13
BCD code A-12
BEGIN command 4-41,53; 6-16.3
Beginning-of-information (BOI) 2-3,4; C-2
BF command J-6
Binary card F-1,5
Binary dump 11-1
Binary file 15-14
Binary input mode K-1
Binary mode 2-7; 9-7
Bisynchronous terminals J-4
Bit C-2
BKSP command 9-4
BLANK command 12-9
Blank label, magnetic tape 12-9
Block C-2
BLOCK command 7-2

Block count G-8,9
 Block mode J-18
 Block size 12-13; N-1
 Block size, terminal input J-6
 Blocking factor J-6
 BOI (See Beginning-of-information)
 BR command J-6
 BREAK key 16-5; J-6
 BRIEF command 8-4
 BS command J-7
 BUFFER OUT statement, FORTRAN 2-7
 BYE command 3-4; 8-19
 Byte C-2
 B1 command J-7
 B2 command J-8

Cancel character J-9
 Cancel line character 16-4,5
 CAP record type 15-3,15
 Card, binary F-1,5,6
 Card, coded F-1,3
 Card deck, physical file structure 2-5,6
 Card file 2-6
 Card file data conversion F-1
 Card formats F-2
 Card punch 1-4
 Card, punch format F-1
 Card reader 1-4
 Cards punched, record of 3-19
 Carriage control array H-7
 Carriage control characters H-2,3,4,5
 Carriage control tape H-4.2
 Carriage return J-12,13; K-1
 Carriage return function 16-3
 Catalog C-2
 CATALOG command 15-3,5
 Catalog track 2-7
 CATLIST command 8-28; 10-3,10-10
 .CC directive 4-33
 CCP (See Communications Control Program)
 CCCCCO file 13-1
 CDC graphic 63/64-character set 9-47
 CDC 711-10 terminal 16-1; J-25
 CDC 713-10 terminal 16-1; J-25
 CDC 714-10/20 terminal 16-1; J-25
 CDC 714-30 terminal 16-1; J-25
 CDC 721 terminal 16-1; J-25
 CDC 722-30 terminal 16-1; J-25
 CDC 722 terminal 16-1; J-25
 CDC 731-12 terminal 16-1; J-25
 CDC 732-12 terminal 16-1; J-25
 CDC 734 terminal 16-1; J-25
 CDC 751-1 terminal 16-1; J-25
 CDC 752 terminal 16-1; J-25
 CDC 756 terminal 16-1; J-25

CDCNET 3-2; 8-9; 16-1
 CDCNET network 8-10
 CEJ/MEJ (See Central/Monitor exchange jump)
 Central library directory 5-5
 Central memory allocation 1-3
 Central memory (CM) 1-2
 Central memory dump 11-1,5,7,9,10
 Central memory, presetting 7-55
 Central memory resident (CMR) 1-3
 Central memory time slice 3-12
 Central processor time, accumulated 3-18; D-4
 Central processor unit (CPU) 1-2; 7-74
 CFO command 7-3
 CH command J-8
 CHANGE command 10-15
 Character C-2
 Character code set conversion 9-23; A-1
 Character conversion E-1
 Character echoing J-14
 Character input mode K-1
 Character set
 Anomalies A-2
 ANSI standard A-1
 ASCII A-1
 ASCII graphic 63-character set A-1
 ASCII graphic 64-character set 9-48; A-1
 ASCII graphic 95-character set 9-48; A-1
 ASCII 128-character set A-1
 Batch job A-2,3,7
 CDC graphic 63-character set A-1,3
 CDC graphic 64-character set A-1,3
 Conversion for magnetic tapes 9-23; A-12
 Definition A-1
 Interactive job 8-2,6.2; A-2,3,5
 NOS-supported A-1
 Recognition J-5
 Character set mode 8-4
 CHARGE command 7-4,69
 Charge number 7-4
 Checkpoint 9-38; 13-1; C-2
 Checkpoint dump 13-1,3
 Checkpoint file 9-38; 12-8,14; 13-1; C-2
 Christensen protocol P-1
 CHVAL command 7-5
 CI command J-9
 CKP command 13-1
 CLASS command 7-7
 CLEAR command 9-5
 Clock cycle count 7-22.1
 CM (See Central memory)
 CMR (See Central memory resident)
 CN command J-9
 COBOL, RECORDING MODE IS BINARY statement 2-7

COBOL, RECORDING MODE IS CODED statement 2-7
 COBOL, sign overpunch convention A-2
 Code set
 ASCII A-1
 Conversion 9-23; A-1
 Definition A-1
 Display code A-1
 NOS-supported A-1
 6-bit display code 8-2,6.2; A-1
 6/12-bit display code 8-2,6.2; A-1
 Code set conversion 9-23; A-10
 Coded card conversion F-4
 Coded card output F-6
 Coded file 9-13,51
 Coded line 2-7
 Coded mode 2-7; 9-7
 Colon, code set representation A-2
 Colon representation F-1
 Command
 Continuation line 5-2
 Format 5-2,3
 Interactive status 8-16
 Multiple entry 7-21
 Prefix character 5-2
 Processing 5-1
 Syntax 5-2
 Terminator 5-2
 Terminator character 5-3
 Command processing
 Dollar (\$) prefix 5-2
 Operating system format 5-2
 Product set format 5-2
 Separator characters 5-2
 Slant (/) option 5-2
 Terminator character 5-2
 Command processing flow 5-5
 Command record
 Job 3-1; C-2
 Procedure 4-3,4,5
 Commands
 File maintenance 14-1
 Global library file 5-1
 Iterative G-31
 Load/dump memory 3-13
 Local file 5-1
 Primary file editing 5-2; 8-39
 Product set 5-1
 Protected 3-13
 Skipping 6-17,18,19,30
 Subsystem selection 8-11
 System 5-1
 Tape management 12-1
 Terminal definition J-1
 COMMENT command 7-9
 Comment (*) directive 4-40.1
 Comment (*) directive 15-20.1,31
 COMMON command 10-16
 Communications Control Program (CCP) 3-2; 16-1; J-1,2
 COMPASS 1-5; 7-39
 COMPASS compressed output 9-26
 Compile C-2
 Compiler language 1-5
 Compressed output, COMPASS 9-26
 Connect time 3-4
 Connection status 6-7
 Constants 6-5
 Continuation line, command 5-2
 Control byte 16-9
 Control character C-2
 Control character, character set A-1
 Control point C-3
 Control registers 6-8
 Control registers, flow control command 6-25; 7-16
 Control statement C-3
 Conversion, code set 9-23; A-10,12
 Conversion mode F-3
 Conversion mode, 9-track tape 12-7,14
 COPY command 2-7; 9-5
 COPYBF command 9-11
 COPYBR command 9-12
 COPYCF command 9-13
 COPYCR command 9-15
 COPYEI command 9-16
 Copying files
 Binary mode 9-7
 Coded files 9-13,15,51
 Coded mode 9-7
 Multifile files 9-10,11
 Multirecord files 9-12
 Octal copy 9-62
 Processing options 9-10,62
 Record replacement 15-7
 Tape files 9-59
 Termination 9-7,10
 Verification 9-10
 Copying records 15-7
 COPYL command 15-3,7
 COPYLM command 15-3,7
 COPYSBF command 9-17; F-2
 COPYX command 9-18
 .CORRECT directive 4-23
 Correspondence code 8-2
 CP command J-10
 CPU mode error 6-22
 CPU priority 7-64
 CPU (See Central processor unit)
 CPU time, accumulated 7-10,17; D-4
 CPU time limit 7-64
 CPUMTR 1-3
 CRM (See CYBER Record Manager)
 CSET command 8-4
 CT command J-10
 ct D command 8-16

ct E command 8-17
 ct S command 8-17
 CTIME command 5-5; 7-10
 CTRL/F character 16-6
 CTRL/P character 16-6
 CTRL/T character 16-6
 Cursor positioning J-10
 CYBER Control Language level 6-7,16
 CYBER loader 1-6; 7-60; 15-32,33; C-3
 CYBER Record Manager (CRM) 1-6; 2-2,3;
 9-1; 10-8; 15-1; C-3
 CYBER 200 Computer System 3-6

 Data blocks, network J-6
 .DATA directive 4-3,4,14,34.2
 #DATA value 4-14,34.2, L-1
 Database file 2-15
 Date 6-7
 Day of the week 6-7
 Dayfile 7-10; C-3; D-4
 Account 3-18
 Job 3-18
 DAYFILE command 7-10
 DC1 J-18,21
 DC3 J-18,21
 DDP (See Distributive data path)
 Deadlock 7-50,52
 Deadstart C-3
 Debug mode 15-17
 DEC VT100 terminal 16-1; J-25
 Default C-3
 Deferred batch jobs 7-68
 Deferred routing 9-35,44,45,46
 DEFINE command 10-17
 DELETE command 8-44
 Deleting input lines 16-4
 Delimiters for input K-1
 Detach command 8-16
 Detached interactive job 2-12; 8-16,30; C-3
 Device C-3
 Assignment 7-57
 Auxiliary 2-16; 10-29
 Family 2-15
 Mass storage 2-15
 Type 6-13,14,15; 7-51; 9-2; 10-4.1; N-1
 DIAL command 8-20.1
 Direct access file 2-11,14; 10-7,17; C-3
 Directory
 Library 5-5
 Random access 15-24
 DIS code 9-23
 Disk, auxiliary 7-51,52,57
 Disk physical file structure 2-5
 Disk storage subsystems 1-4; N-1
 Display code C-3
 Display code, 6-bit 8-2; 9-23; A-1

Display code, 6/12-bit 9-23; A-1
 DISPLAY command 6-16.4
 Disposition code C-3
 Disposition code, ROUTE command 9-44
 Distributive data path (DDP) 1-4; N-1
 DIS63 code 9-23
 DIS64 code 9-23
 DL command J-11
 DMB command 11-1
 DMB extended memory error 11-4
 DMB output 11-3
 DMD command 11-5
 DMDECS command 11-6
 DMP command 11-6.1
 DMPECS command 11-7
 DOCUMENT command 9-20
 DROP command 7-12.1
 DROPDS command 10-20
 Dropping jobs 7-12
 DT function 6-14
 Dual processors 3-18; 7-64
 Dual-state, NOS/VE 3-5
 Dump 9-62; 10-33
 Central memory 11-1,5,6.1,9,10
 Checkpoint 13-1,3
 Exchange package 11-1,5,6.1
 Extended memory 11-1,6,7
 Field length 11-1
 Dump memory commands 11-1
 DUP command 8-45

 E format tapes 9-59,64; I-2
 EB command J-12
 EBCDIC code A-12; C-3
 EBCFL code 9-23
 Echoing J-14
 Echoplex mode J-14
 ECS (Extended Core Storage; See Extended
 EDIT command 14-1
 EEA (See Error exit address)
 EFFECT command 8-4
 EL command J-13
 ELSE command 6-17
 .ELSE directive 4-4,38
 Empty PRU/record C-4
 End-of-block character J-12
 End-of-file (EOF) 2-3,4; 9-51; C-4; F-2
 End-of-file label G-1,8,10
 End-of-information (EOI) 2-3,4; C-4; F-2
 End-of-line F-2; H-1; K-1
 End-of-line byte A-2; F-1
 End-of-line character J-13
 End-of-record (EOR) 2-3,4; 9-51; C-4; F-2
 End-of-reel I-3
 End-of-tape (EOT) 12-4,5; C-4; G-1; I-3
 End-of-volume label G-1,9,10

.ENDHELP directive 4-24
 ENDIF command 6-18
 .ENDIF directive 4-4,39
 ENDW command 6-19
 ENQUIRE command 2-12; 7-13; 9-50
 ENTER command 7-21
 .ENTER directive 4-24
 Entry point, definition C-4
 Entry point, library directory 15-33
 Entry point listing 15-5,15
 .EOF directive 4-38
 EOF (See End-of-file)
 EOFn label G-1,10
 EOF1 label 2-8,9; 12-9,11; G-8
 EOI (See End-of-information)
 .EOR directive 4-38
 EOR (See End-of-record)
 EOT (See End-of-tape)
 EOvn label G-1,10
 EOv1 label 2-8,9; G-9
 EP command J-14
 Epilogue C-4
 Epilogue, project C-11
 Equipment access level C-4
 Equipment status table (EST) 3-19
 ERRMSG command 7-22
 Error codes, system 6-9,10
 Error exit mode 6-22
 Error flag 6-8,25; C-4
 Error messages B-1
 Error processing 3-12; 5-7; 6-23
 EST ordinal C-4; D-4
 EST (See Equipment status table)
 EVFU load file H-2,4.2
 EVICT command 9-22
 .EX directive 4-9,39
 Exchange package 3-12; C-4
 Exchange package dump 11-1,5,6.1
 Executable object code C-5
 EXECUTE command 8-13
 EXECUTE mode 10-8
 Execute subsystem 8-13
 Execution time 1-2
 EXIT command 3-12; 5-7; 6-19
 Exit mode 6-8
 Exit processing 5-7
 .EXPAND directive 4-6,33,34.2
 EXPLAIN command 8-20.1
 Extended instruction stack purging C-5
 Extended memory
 Description 9; 1-3; C-5
 Direct access files N-1
 Dump 11-1,6,7
 Error 11-4
 Field length 3-9; 7-25,39
 Library directory 15-33
 Preservation 7-44
 Extended stack purging 7-39
 External references, library directory 15-33
 F tape format 2-10; 9-10
 FA command J-14
 Family 2-15; D-2
 Family device 2-15; C-5
 Family name 6-8.1; 7-74; C-5
 FCI 12-4; C-5
 FCOPY command 2-7; 9-23; A-12
 FET (See File environment table)
 Field length, extended memory (FLE) 3-9
 Field length (FL) 1-2; 3-9; 7-24,39,57; C-5
 Load/Dump 11-1
 Field length, maximum (MAXFL) 3-9; 7-39
 Field length, running (RFL) 3-9; 7-40
 File
 Append operations 8-51; 10-7
 Assignment to device 9-2
 Backspacing 9-4
 Backup copy 10-3
 Copying 8-51; 9-5,16,17,62; 15-7
 Copying lines 8-49,52
 Creation 2-12,13
 Creation date 10-13; G-8,9
 Definition 2-1; C-5
 Device residence 2-7,15,16
 Editing 14-1,14,18
 Execution 8-35
 External characteristics 9-46
 External documentation 9-20
 Generation number G-8,9
 Information 15-5
 Internal characteristics 9-48
 Internal documentation 9-20
 Listing record information 15-14
 Locked 9-26,65
 Logical 2-1,2
 Maintenance 15-7,11
 Organization 2-3
 Overwriting data 9-35
 Packing 9-36
 Physical structure, by device type 2-5
 Positioning 9-4,43,56,57,58
 Printed output formatting 2-12
 Printing 9-46
 Purging 10-31,32
 Reformatting 9-27
 Releasing 9-5,22,42,56,64
 Replacement 10-45
 Resequencing 9-39
 Retrieval 10-7,20,21,28
 Rewinding 9-43
 Routing 7-36; 9-31,35,44
 Simultaneous user access 10-8,9
 Sorting 9-40,58
 Status 7-15,27; 9-56
 Storing 10-45,46
 Structure 2-1
 Structure, CRM logical 2-3

File
 Structure reporting 9-51
 Subdivisions 2-1,2,3
 Subsystem association 8-12; 10-5
 Terminology 2-3
 Transfer between mainframes
 9-31,49,50; 10-21
 Types 2-11
 Verification 15-35
 Verifying 9-65
 File access category 3-14; C-5
 File access level 3-14; C-6
 File access mode 2-14; 10-8,10
 File accessibility character, tape 12-10,15
 File attribute 6-12
 File, binary 15-14
 File, card 2-6
 File catalogue 10-10
 File, checkpoint 9-38; 13-1
 File, coded 9-13,51
 FILE command, CRM 2-7
 File, database 2-15
 File, direct access 2-13,14; 10-7,17
 File directives 4-34.1
 File directory 15-1
 File environment table (FET) C-6
 File expiration date G-8,9
 FILE function 6-12,13
 File header label G-1,4,10
 File, indirect access 2-14; 10-7,20; C-7
 File information table (FIT) C-6
 File marks 2-5; 9-36,51,68
 File, mass storage 2-6,15
 File, multifile 2-3,4
 File name 2-1
 File name table (FNT) 5-5
 File password 10-4.1
 File, permanent 2-13,14; 10-10,15,45,46
 File permission mode 10-5,30
 File permit category 10-3
 File, primary 2-13; 9-37
 File, private 10-3
 File, program library 2-17
 File, public 10-3
 File, queued 7-45; 9-31,44
 File residence 2-15; 10-4.1,29
 File, semi-private 10-3
 File sequence number, tape 12-15
 File set C-6
 File, SYSTEM 10-16
 File, temporary 2-12
 File transfers
 Mainframe-mainframe 10-23
 Mainframe-micro P1
 File types 2-11
 File, user library 2-17
 #FILE value 4-14; L-1
 FIT (See File information table)
 FL (See Central memory field length)
 Flag C-6

FLE (See Extended memory field length)
 Flow control command
 Command descriptions 6-16.3
 Continuation lines 6-2
 Control registers 7-16
 Expressions 6-2
 Format 6-2
 Functions 6-12
 Operands 6-2
 Operators 6-2
 Symbolic names 6-7,25
 Syntax 6-2
 .Fn directive 4-24.1
 FNT (See File name table)
 Format channel selection H-4.2
 Format controls, 580 line printer H-1
 Format effectors 8-4
 Forms code 9-47
 FORTRAN C-6
 FORTRAN, BUFFER OUT statement 2-7
 FORTRAN command 8-14
 FORTRAN subsystem 8-14
 Frame C-6
 FSE command 14-2
 FTNTS command 8-15
 FTNTS subsystem 8-15
 FTP Command 10-20.1
 FTP Directives 10-20.1
 APPEND_FILE Directive 10-20.2
 CHANGE_AUTO_LOGIN_MODE
 Directive 10-20.3
 CHANGE_BELL_MODE Directive 10-20.3
 CHANGE_DEBUG_MODE Directive 10-20.4
 CHANGE_FILE_NAME Directive 10-20.4
 CHANGE_INPUT_FILE Directive 10-20.4
 CHANGE_OUTPUT_FILE
 Directive 10-20.5
 CHANGE_PORT_COMMAND_MODE
 Directive 10-20.5
 CHANGE_TRANSFER_TYPE
 Directive 10-20.5
 CHANGE_VERBOSE_MODE
 Directive 10-20.6
 CREATE_HOST_CONNECTION
 Directive 10-20.6
 DEFINE_AUTO_LOGIN
 Directive 10-20.7
 DELETE_FILE Directive 10-20.7
 DELETE_HOST_CONNECTION
 Directive 10-20.8
 DISPLAY_COMMAND_INFORMATION
 Directive 10-20.8
 DISPLAY_DIRECTORY
 Directive 10-20.8
 DISPLAY_FILE_NAMES
 Directive 10-20.9
 DISPLAY_FTP_OPTIONS
 Directive 10-20.9
 DISPLAY_LOCAL_HELP
 Directive 10-20.9

DISPLAY_REMOTE_HELP
 Directive 10-20.10
 GET_FILE Directive 10-20.10
 LOGIN_USER Directive 10-20.12
 QUIT Directive 10-20.13
 REPLACE_FILE Directive 10-20.14
 SEND_FTP_COMMAND Directive 10-20.15
 Full-ASCII mode 8-2; J-14
 Full-duplex mode J-14
 Full Screen Editor (FSE) 14-2
 Batch mode 14-2
 Function keys 4-24.1
 .F7 directive 4-24.1

GE tape density 7-54
 Generation C-6
 GET command 10-21
 GID (See Record group identifier)
 Global error flag 6-8,25
 Global library set 5-5; 6-13; 15-43; C-6
 GO command 7-22
 GOODBYE command 3-4; 8-21
 Graphic C-6
 Graphic character A-1
 GTR command 15-3,11
 GTR directives 15-12

HAD (See Host availability display)
 Half-duplex mode J-14
 Hardware, System 1-1
 HASP terminals 16-1; J-1,4,25
 HAZ terminal J-1
 Hazeltine 2000 terminal 16-1; J-1,25
 HC command J-15
 HD command J-15
 HD tape density 7-54
 HDRn label G-1,10
 HDR1 label 2-8,9; 12-9,11,15; G-4
 HELLO command 3-5; 8-21
 HELLO7 command 3-5,6.1; 8-22
 HELP command 8-26
 .HELP directive 4-25
 Help, online 8-26
 HELPME command 8-26
 HHA (See Highest high address)
 Highest high address (HHA) 3-9
 HN command J-17
 Hollerith punch code F-3
 Hollerith punch output F-5
 Host availability display (HAD)
 3-5; J-15,16
 Host node selection J-15,18
 HTIME command 5-5; 7-22.1

I format tapes 2-10; 9-10,59,64
 IAF (See Interactive Facility)
 IBM 2741 terminal 16-1,8; J-25
 IBM 2780 16-2; J-25
 IBM 3780 16-2; J-25
 IC command J-18
 .IC directive 4-34
 Idle characters J-9
 IF command 6-20
 IFE command (See IF command)
 .IF directive 4-4,40
 .IFE directive (See .IF directive)
 IGNORE directive 15-27
 Immediate job status command 8-17
 Implicit remote routing 9-48
 Implicit routing 9-50
 IN command J-18
 Indefinite operand error 6-23
 Indirect access file 2-14; 10-7,20; C-7
 Input data E-1
 Input device selection J-18
 Input disposition 2-11; 9-44
 Input file 2-11; 3-11; C-7
 INPUT, file 3-11
 Input flow control J-18
 Input line
 Correction 16-3
 Deletion 16-4
 Entry before prompt 16-4
 Length 16-3
 Termination 16-3
 Input prompt 16-4
 Input queue 2-11
 Input transmission mode J-18
 INSERT directive 15-27
 Instruction stack purging 7-39; C-7
 Interactive command names 5-5
 Interactive Facility (IAF) 1-1; 3-2,4,6;
 7-32; 8-4,18; 16-1; C-7; J-1,25
 Interactive job 3-1
 Interactive job origin 3-6.2
 Interactive job subsystem 6-11
 Interactive status commands 8-1,16; 16-4
 Interactive Transfer Facility (ITF) 3-6;
 7-32; 8-18
 Interrecord gap C-7
 Interruption character 16-6; J-7
 Interruption sequence 16-4,6,7; C-7; J-6,7
 ITEMIZE command 15-3,14
 Iterative command 6-31
 ITF (See Interactive Transfer Facility)

Job C-7
 Accumulated CPU time 7-10
 Checkpoint 13-1
 Completion 3-17
 Control 3-9

Dropping 7-12
 Entry 3-1; 7-25,68
 Error processing 6-23
 Exit processing 5-7
 Field length 1-2; 3-9; 7-24,35,57
 Initiation 3-1
 Output 3-19; D-1
 Output processing 2-12; 7-62
 Remote routing 3-6; 9-31,44; 10-21
 Resource demand 7-14,51
 Resource usage 3-18; 7-17,59,64,67
 Restart 13-1
 Scheduling 3-9
 Scheduling priority 3-6.2
 Security 3-13
 Status 7-13,21,73; 8-17,31
 Summation of activity 3-18
 Suspension 7-62,63,65
 Termination 2-11; 3-1; 5-7; 7-62
 Time limit 7-24
 Timeout period 3-4
 Job access category set C-7
 Job access level C-7
 Job card 7-23
 Job command 3-6.1; 7-23
 Job communication area 5-4
 Job control
 Command limit 3-11
 Error 3-12
 Input file 3-11
 Rollout 3-12
 SRU limit 3-11
 Time limit 3-11
 Job file 2-11
 Job input file 3-11
 Job, interactive 3-1
 Job name 3-7
 Job origin type 3-6.2; 6-7; D-2
 Job priority 2-12; 3-7,9; 7-23,24,64
 Job processing commands 8-18
 Job processing parameters 7-23
 Job, remote batch 9-50
 Job, rolled out 3-12
 Job scheduler 7-58
 Job sequence name (JSN) 3-6.2; 7-18; C-7;
 D-2
 Job service class 3-6.2
 Job step C-7
 Interruption sequence 16-6
 SRU limit 3-11
 Suspension 16-7
 Time limit 3-11; 7-64
 Job termination 2-11
 JSN (See Job sequence name)

KEY command 8-5
 Keypunch conversion mode 2-6; F-3
 KRONREF command 14-4

L format tapes 2-10; 9-10; 12-2
 L tape format 2-7
 Label C-7
 ANSI G-1
 Magnetic tape 2-8; 12-11,19; G-1
 Optional G-1,10
 Required G-1
 User G-11
 LABEL command 2-7; 12-11
 Label standard level 12-10
 Labeled tape 12-7,13
 LBC command 11-8
 LCN (See Loosely coupled network)
 LDI command 7-26
 LEN function 6-16
 LENGTH command 7-27
 Level number C-8
 LI command J-20
 LIB command 8-26
 LIBEDIT command 15-3,17
 LIBEDIT directives 15-20
 LIBEDIT output 15-31
 LIBGEN command 15-28
 Library 2-17; 15-1; C-8
 Creation 15-17,34
 Global library set 15-43
 Maintenance 15-17,34
 Processing examples 15-37
 Program 14-5,8,14
 System 15-43
 User 15-33,34,43
 LIBRARY command 15-43
 Library directory
 Entry points 15-33
 External references 15-33
 Library file 2-13; C-8
 Library file type 2-11
 Library maintenance 15-1
 Library record type 15-2,3
 Library search order 15-44
 Library, user 15-1,33
 Structure 15-33
 LIBRARY, user name 2-17
 LID (See Logical identifier of mainframe)
 LIMITS command 7-28
 Line C-8
 Length 16-3

Logical 16-3
 Physical 16-3
 LINE command 8-6.1
 Line feed J-12,13; K-1
 Line feed function 16-3
 Line feed idles J-20
 Line mode 8-6.1; C-8
 Line numbers 8-3,39; 9-40,58
 Line printer 1-4
 Auto eject mode H-2
 Carriage control characters H-2,3
 Character sets 9-47; A-3
 Paper size H-1
 Print trains A-3
 Punched tape loop H-4
 533/536 9-48; A-4; H-1
 580 H-1,4
 585 9-48
 Line printer carriage control H-1
 Line sequence number 9-40
 Line speed recognition J-5
 Lines printed, record of 3-19
 Lines printed, total 3-19
 Linked host 7-36
 LIST command 8-28,46
 LIST directive 15-28
 Listing record information 15-14
 Listing tape labels 12-19
 LISTLB command 12-19
 LISTLID command 7-36
 LIST80 command 9-26
 Literal card input F-3
 Literal string 5-3; 6-5
 LK command J-20
 Load/dump memory commands 3-13; 11-1
 Load point C-8
 Loader information 7-16
 Loading
 Binary data 11-8
 Central memory 11-9
 Octal lines 11-8
 Loading (also see CYBER Loader) Central
 memory 1-6
 LOC command 11-8
 Local applications 7-32
 Local batch job 3-6.2
 Local file 2-12; C-8
 Creation 2-12,13
 Local file name C-8
 Local file name table (FTN) C-8
 Local file type 2-11
 LOCK command 9-26
 Locked file C-8
 Lockout of messages J-20
 Logical device 2-6; C-8
 Logical identifier of mainframe (LID)
 7-36; C-8
 Logical line 16-3; K-1
 Logical record C-8
 Logical track 2-6
 Login 3-2,5; 8-28; C-8
 LOGIN command 8-28
 Logout 3-4,5; 8-21,29; C-8
 LOGOUT command 3-4; 8-29
 Logout display 3-4; 8-20
 Loopback 7-38
 Loosely coupled network (LCN) C-8
 LO72 command 9-27
 LVL function 6-16
 MA (See Monitor address)
 MACHINE command 7-39
 Machine identifier 3-19; 6-8.1; D-4
 Macro C-8
 Magnetic tape 1-4
 ANSI labels 12-2,9
 ANSI standard labeled 2-8,9
 Assignment 12-1,5,21
 Blank label 12-9
 Block size 12-13
 Block terminator 2-10
 Checkpoint dump 13-2
 Checkpoint file 12-8,14
 Code set A-12
 Command rules 12-2
 Conversion mode for 9-track tapes 12-7,14
 Copying 2-7
 Creating files 12-5
 Data formats 2-5,9; I-1
 Density 2-8; 7-54; 12-6,13
 Density specification 12-2
 Device type 12-6
 Error processing 12-3
 Error recovery 2-10
 Expiration date 12-16
 File accessibility character 12-10,15
 File positioning 12-11
 File sequence number 12-15
 File set configurations 2-8
 Generation number 12-15
 Generation version number 12-16
 Label 2-8; 12-9,11; G-1
 Label standard level 12-10
 Label verification 12-11
 Labeled 9-59,64; 12-7,13,21
 Load point 12-2
 Maximum block size 9-62
 Multifile set 2-9; 12-11,15
 Multivolume set 12-25
 Noise block 9-62
 Noise size 12-8,14
 Nonstandard labeled 2-8
 Obsolete tape formats I-1
 Overwriting labels 12-11

Parity 2-7,8,10
 Parity Error 12-3
 Physical file structure 2-5
 Processing option 12-3
 Recording mode 2-7
 Section number 12-15
 Set identifier 12-15
 Tape mark 2-8
 Trailer sequence 12-4
 Unlabeled 9-4; 12-7,13
 Unloading 12-5
 Verification 12-11
 Volume name 12-25
 Write ring 12-4
 7-track 2-7,8
 9-track 2-7,8
 Magnetic tape activity 3-18
 Magnetic tape file 9-43
 Manual references 11
 MAP (See Matrix Array Processor)
 Mass storage C-9
 Mass storage activity 3-18; D-4
 Mass storage, Alternate 2-15,16
 Mass storage device 1-4; 2-15; N-1
 Mass storage file 2-6
 Master device C-9
 Master user 14-11,12
 Matrix Array Processor (MAP) 3-8
 Resource usage 3-18
 MAXFL (See Field length, maximum)
 Memory)
 MERGE file 14-16
 Message Control System (MCS) 3-6; 7-32;
 8-18
 Message delimiters K-2
 Message lockout, operator J-20
 Message to network operator J-21
 Message transmission key J-12,13; K-1,2
 MFL command 7-39
 MFLINK
 Directives 10-23
 Interactive use 10-27
 MFLINK command 3-6,6.1; 10-22
 MFQUEUE
 Directive prefix character 9-31
 Routing directive 9-33
 MFQUEUE command 3-6,6.1; 9-31
 Microcomputer file transfers P-1
 MODE command 6-22
 Mode 4 terminals J-4
 Modification symbols 7-50.1
 MODIFY command 14-5
 MODIFY mode 10-8
 MODIFY utility 14-8
 MOVE command 8-48
 MS command J-21
 Multifile file 2-3,4; C-9
 Multifile set C-9
 Multitmainframe configuration 1-3; 2-15
 M33 teletype 16-1; J-25
 M35 teletype 16-1; J-25
 M37 teletype 16-1; J-25
 M38 teletype 16-1; J-25
 M40 terminal 16-1; J-25

 NAM (See Network Access Method)
 NAM/CCP network commands 8-26
 NAME directive 15-31
 NDL (See Network Definition Language)
 Network 16-1; C-9; J-1
 Applications 3-6; 7-32
 Configurations 7-36
 Processing modes 1-1
 Network access device 1-5
 Network Access Method (NAM) 1-1; 7-38
 Network control character J-10,21
 Network Definition Language (NDL) J-2,24
 Network Job Entry Facility (NJF) 7-32; 8-18
 Network operator (NOP) C-9
 Network Operator Utility (NETOU) 7-32; 8-18
 Network processing unit (NPU) 1-5
 Network type 6-8.1
 NETOU (See Network Operator Utility)
 NEW command 9-34
 NEW directive 15-28
 NEWPL file 14-16
 NJF (See Network Job Entry Facility)
 No abort option 10-3
 No-auto-drop status (ND) 8-27; 9-56
 .NOCLR directive 4-31
 NOEXIT command 3-12; 5-7; 6-24
 NOINS directive 15-28
 Noise C-9
 Noise size 12-8,14
 Nonallocatable device C-9
 NOP (See Network operator)
 NOREP directive 15-29
 NORERUN command 7-40
 NOREW directive 15-29
 Normal mode 8-2,4,7; A-3; E-1
 NORMAL mode 8-4; A-3
 Normalized input mode J-18; K-1
 NOS/VE dual-state system 3-6
 NOS/VE Interactive Facility (VEIAF) 7-32;
 8-18
 NOSORT command 8-29
 NOSTEXT file 14-4
 NOTE command 7-40
 .NOTE directive 4-31
 NPC= entry point 5-2; 15-43
 NULL command 8-15
 Numeric string 6-5
 NUM function 6-16.1
 Object code C-9

OC command J-21
 OFFSW command 7-41
 OLD command 10-28
 OLD directive 15-29
 OLDPL file 14-17
 ONEXIT command 5-7; 6-24
 Online manuals 8-20.1,26
 ONSW command 7-42
 OP command J-21
 Operand overflow error 6-23
 Operand underflow error 6-23
 Operands 6-5
 Operating system level 6-7
 Operating system value 6-11
 Operating system version D-2
 OPL file 14-4
 OPL record type 15-3,6,15
 OPLC record type 15-3,6,15
 OPLD directory 15-24,33
 OPLD record type 15-3,6,13
 OPLEDIT command 14-8
 Origin type C-9
 OUT command 9-35
 Output
 Cards punched 3-19
 Data E-2
 Device selection J-21
 Flow control J-21
 Lines printed 3-19
 Listing identification 2-2
 Print file code conversion A-3
 Printed H-1
 Processing 2-12; 7-62
 Queued job 2-12
 Remote batch 2-12
 Suppression of headers 8-4
 Suspension 16-5,7
 OUTPUT file 2-11; 3-19; 7-68; 9-44,45
 Output line
 Aborting 16-5
 Length 16-3
 OVCAP record type 15-4
 OVL record type 15-3
 OWRITE command 9-35

 PA command J-22
 PACK command 9-36
 Pack name 10-3,29
 Packet-switching network (PSN) 16-1; C-9
 PACKNAM command 10-29
 Page boundary 16-5
 Page density 6-7
 .PAGE directive 4-32
 Page length 6-7; J-23
 Page size 6-7
 Page wait 16-5; J-23

 Page width 6-7; J-24
 Panel compilation 14-10
 Panel definition file 14-10
 Panel, full screen C-9
 PANELIB 8-36
 Paper length, line printer H-1
 Paper size, line printer H-1
 Paper tape
 Control characters M-2
 Corrections M-4
 Input lines M-2
 Operation M-1
 Punch M-1
 Punching off-line M-3
 Punching online M-4
 Reader M-1
 Parameter C-9
 Keyword 5-4
 Order-dependent 5-4
 Order-independent 5-4
 Parity C-10
 Magnetic tape 2-7,10
 Parity processing J-22
 Passive procedure 4-1; C-10; L-1
 PASSWOR command 7-42
 Password C-10
 Batch 7-42
 Changing 7-42
 Entry 7-74
 Interactive 7-42
 Pause bit 7-22,44
 PAUSE command 7-44
 PBC command 11-9
 PDU command 14-10
 PE tape density 7-54
 Peripheral equipment 1-4,5
 Peripheral processor library 5-5
 Peripheral processor (PP) 1-4; C-10
 Permanent file 2-13,14; 10-10,15,45,46;
 C-10
 Permanent file activity 3-18; D-4
 Permanent file catalog 2-12
 Permanent file device C-10
 Permanent file family C-10
 Permit category 10-3
 PERMIT command 10-30
 Personal identifier 3-3,13
 PFC array, 580 line printer 9-48.2; H-1,6
 PFC printer H-1
 PFC (See Programmable format control)
 PG command J-23
 Physical line 16-3; K-1
 Physical mainframe identifier (PID) 7-36
 Physical record unit (PRU) 2-4; C-10; N-1
 PID (See Physical identifier of mainframe)
 PL command J-23
 PLATO - NAM Interface (PLATO) 7-32; 8-18
 Plot disposition 2-11; 9-45

Post radix C-10
 POST terminal J-1
 PP record type 15-3
 PP (See Peripheral processor)
 PPU record type 15-3
 Prefix character, command 5-2
 \$ Prefix, command 5-2
 Prefix table 15-25
 Prefix table, record 15-2,35
 PRIMARY command 9-37
 Primary file 2-13; C-10
 Changing lines 8-42
 Creation 9-34
 Editing 8-1,39
 Inserting lines 8-42
 Line number overlap 8-41
 Type 2-11
 #PRIMARY value 4-14; L-1
 Print density H-1,7
 Print disposition 2-11; 9-44
 Print trains A-3
 Printer carriage control H-1
 Printer Support Utility (PSU) 7-32; 8-18
 Printing files 9-46
 Priority, job 3-9
 Priority, scheduling 3-7
 Private auxiliary device C-10
 Private file 10-3
 Privileged network applications 7-32
 .PROC directive 4-2
 Interactive format 4-11
 Menu format 4-22
 Passive format L-1
 PROC record type 15-3
 Procedure C-10
 Branching directives 4-38
 Call 4-41
 Checklist entries 4-13
 Checklist patterns 4-14,15,16
 Command record 4-4
 Command section 4-3
 Concatenation character 4-8
 Data section 4-3,6,34.2
 #DATA value 4-14
 Default termination sequence 4-51
 Directive syntax 4-9
 Directives 4-2
 Expansion control directives 4-33
 Expansion processing 4-4,8
 File directives 4-34.1
 File search order 4-42
 #FILE value 4-14
 Formatting and Help directives 4-23
 Formatting section 4-2
 Help section 4-2
 Inhibit character 4-6,7
 Interactive help processing 4-25
 Interactive parameter entry 4-4
 Interactive parameter format 4-12
 Menu 4-22
 Menu help processing 4-25
 Nesting level 4-1,14; 6-9
 Noninteractive 4-1; L-1
 Order-dependent parameter processing 4-53
 Order-independent parameter processing 4-53
 Parameter matching modes 4-53; L-2
 Parameter substitution L-2
 Passive L-1
 #PRIMARY value 4-14
 Processing 4-3
 Programmable function keys 4-24.2
 Prologue (See Prologue)
 Requesting help 4-46
 Special parameter values 4-14
 Structure 4-2
 Termination 4-41,47,51
 Procedure file C-11
 Residence 4-2
 Search order 4-42
 Procedure header directive 4-2
 Interactive format 4-11
 Interactive parameter format 4-12
 Menu format 4-22
 PROFILC file 14-11
 PROFILE command 14-11
 PROFILE directives 14-12
 Program library 2-17; 14-5,14
 Programmable format control (PFC) 9-48.2; H-1
 Programmable function keys 4-24.1
 Project number 7-4
 Project profile file 14-11
 Prologue 7-4,30,31,73; C-11
 Prologue, project C-11
 .PROMPT directive 4-32
 PROTECT command 7-44
 Protected command 3-13
 PRU device C-11
 PRU (See Physical record unit)
 PSN (See Packet-switching network)
 PSR level 6-16
 PSU (See Printer Support Utility)
 Public auxiliary device C-11
 Public file 10-4
 Punch disposition 2-11; 9-45
 PUNCH file 2-11; 3-19; 9-45; F-5,6
 PUNCHB file 2-11; 3-19; 9-45; 11-9; F-6
 PURGALL command 10-31
 PURGE command 10-32.1
 PW command J-24
 P8 file 2-11; 3-19; 9-45; F-7

 QGET command 7-45
 QTF (See Queued File Transfer Facility)
 Queue priority C-11
 Queued file 2-11; 7-45; 9-31,35,44; C-11

Queued File Transfer Facility (QTF) 3-6
 Queued job output 2-12
 QUEUE7 command 7-46

RA (See Reference address)
 Random access 15-1,2; C-11
 Random access directory 15-11,17
 Random access file C-11
 RBF (See Remote Batch Facility)
 RBR command 11-9
 RC command J-24
 READ command 8-49
 READ mode 10-8.1
 Read-only permission C-11
 READAP mode 10-8
 READMD mode 10-8
 READUP mode 10-8
 Real-time clock 7-58
 RECLAIM command 10-33
 Database 10-33
 Directive options 10-39
 Directives 10-36
 Format 10-35
 Output listing 10-44
 Record C-11
 Checksum 15-6,15
 Comments field 15-6,15
 Copying 15-7
 Creation date 15-6,15
 File subdivision 2-3
 Length 15-6,15
 Name 15-15
 Prefix table 15-2,25
 S-type 15-1
 Type 2-3; 15-3,15,16
 U-type 15-1
 Record group identifier (GID) 15-21
 Record splitting 9-9,10
 Recording mode
 Binary 2-7
 Coded 2-7
 RECORDING MODE IS BINARY statement, COBOL 2-7
 RECORDING MODE IS CODED statement, COBOL 2-7
 Records, relocatable 15-7
 RECOVER command 8-30
 Recoverable jobs 3-3; 8-30
 Recovery C-11
 Recovery processing 8-30,32
 REDO command 7-50
 Reel number G-2
 Reference address (RA) 1-2
 Reference address (RA and RAE) C-11
 Reference record identifier (RID) 15-21
 Reformatting directives 7-69

REL record type 15-3,6,15
 Related publications 11
 Releasing file space 9-22
 Releasing files 9-5,42,56,64
 Relocatable records 15-7
 Remote Batch Facility (RBF) 1-1; 3-1,6; 7-32; 8-18
 Remote batch job 3-6.2; 9-50
 Remote host 7-36; 9-31; C-12
 Remote Host Facility (RHF) 7-38; C-12
 Removable packs 7-57
 RENAME command 9-37
 RENAME directive 15-29
 REPLACE command 10-45
 REPLACE directive 15-30
 Replacing files 10-45
 REQUEST command 9-38; 12-21
 RERUN command 7-50.1
 RESEQ command 9-40
 Reservation block 2-7
 Reserved format channels for VFU processing H-9
 RESOURC command 7-50.1,55
 Resource demand 7-14,51,54,56; 9-42
 Resource limits 7-29
 Resource overcommitment 7-55
 Resource usage 7-16.1,59,64,67; D-4
 Resource usage, record of 3-18
 Restart 13-1,2
 RESTART command 13-2
 RETURN command 7-54; 9-42
 REVERT command 4-41,51
 REWIND command 9-43
 REWIND directive 15-30
 Rewinding files 9-43
 RFL command 7-40,57
 RFL= entry point 15-43
 RFL (See Running field length)
 RID (See Reference record identifier)
 Rolled out C-12
 ROLLOUT command 7-58
 Rollout control 3-12
 ROUTE command 3-6,6.1; 9-44
 Routing files 9-31,44
 RTIME command 5-5; 7-58
 Rubout M-2
 Rubout characters C-12
 RUN command 8-35
 Running field length (RFL) 3-9; 7-40

S tape format 2-7,10; 9-10; 12-2
 S-type record 15-1
 Satellite coupler 1-5
 SAVE command 10-46
 Scheduler 3-9; 7-58
 Scheduling priority 2-12; 3-6.2,7,9; C-12

SCOPE 2 Operating System 3-7; 7-46; 8-22
SCOPE 2 Station Facility (SSF) 7-38
SCOPY command 9-51
Scratch files 2-1
SCREEN command 8-7
Screen formatting 8-36; 14-10
Screen mode 8-7; C-12
Scrolling mode 8-6.1
SDM= entry point 15-43
SE command J-24
SECHDR command 9-54
Secure login command 8-36
Secured mode 3-13
Secured system 9-55; 10-48,49; C-12
Security access categories 10-48
Security access level 7-33,61; 9-3,55;
10-2,49; C-1
Security administrator C-12
Security count C-12
Security features 3-13; 8-36; 9-54,55;
10-48,49
Security permissions 7-33
Security unlock status C-12
Semi-private file 10-4
Sense switch 6-12; 7-41,42,73
Separator, command 5-2
Sequence number C-12; G-4,8,9
Sequential access 15-1; C-12
Sequential (SQ) file C-12
Service class 2-12; 3-6.2,7;
7-5,7,18,23,33; C-13; D-2
SET command 6-25
.SET directive 4-34
Set identification G-4,8,9
Set identifier 12-15
SETASL command 7-59
SETCORE command 7-60
SETFAL command 9-55
SETFS command 9-56
SETJAL command 7-61
SETJOB command 7-62
SETJSL command 7-63
SETPFAC command 10-48
SETPFAL command 10-49
SETPR command 7-64
SETTL command 7-64
SHELL command 7-66
SHELL permissions 7-32
Shell program 7-66
Short PRU C-13
SHOW command 8-36
SI format tapes 2-10; 9-10,59,64
SKIP command 6-30
SKIPEI command 9-56
SKIPF command 9-57
SKIPFB command 9-57
Skipping commands 6-17,18,19
SKIPR command 9-58
Software, system 1-5
SORT command 9-58
Sort flag 8-29
Sorting files 8-29; 9-40,58
Source code C-13
Special editing mode 8-2; C-13; J-24
SRU (See System resource unit)
Stack purging 7-39
Standard labeled tape C-13
STIME command 5-5; 7-67
Store and forward 7-38
Storing files 10-45,46
STR function 6-16.1
STRB function 6-16.2
STRD function 6-16.3
String C-13
SUBMIT command 7-68
 Directives 7-68
 Transparent submit mode 7-68
Submit file 7-68
Subsystem 6-7; 8-11; 10-5
Subsystem, default 8-15
Subsystem flag C-13
Subsystem selection commands 8-1,11
SWITCH command 7-73
Symbolic names, Flow control command
6-7,25
Synchronous terminal output 16-5
SYSEDIT, SC directive 5-2
System access category set C-13
System access level C-13
System code G-4,8,9
System error codes 6-9,10
System file C-13
SYSTEM file 10-16
System library 5-2; 10-16; 15-43
System monitor 1-3
System origin job 3-6.2
System prologue 7-30
System resource unit (SRU) 3-8;
7-59,64,67; C-13; D-4
 Limit 3-11
 Resource usage 3-18
System resource usage 7-16.1
System resources, accounting of 3-8; 7-4
System security features 3-13; 8-36
System symbols, cross reference listing 14-4
System text 14-4
SYSTEXT file 14-5
TAF (See Transaction Facility)
Tape assignment 12-5,21
Tape density 7-54
Tape density identifier 7-54
Tape format C-13
Tape formats 2-7,10; 12-13; I-1

Tape formats, obsolete I-1
 Tape management 12-1
 Tape mark C-13
 Tape marks 2-8; 12-2
 Tape owner identification G-2
 Tape (See also Magnetic tape)
 Tape Verification Facility (TVF) 3-6
 TAPEn file 11-10
 TC command J-2,25
 TCOPI command 2-7; 9-59
 TDU command 8-8.1
 TDUMP command 9-62
 Tek terminal J-1
 Tektronix 4010/4014 terminal 16-1; J-25
 Teletypewriter (TTY) 16-1; J-1
 Televideo terminals 8-6.2,8; 16-1
 Temporary file C-13
 Terminal
 Categories J-4
 Character conversion E-1
 Characteristics 8-9; J-1
 Classes 16-1; J-1,24,25
 Control 16-9
 Definition commands 16-2
 Display of characteristics J-8
 Input/output conventions 16-2
 Name C-13
 Output 16-5
 Session 3-3
 Terminal attributes 8-9
 Terminal character set-mode 6-6
 Terminal characteristics 3-5; J-1
 Terminal characteristics, CCP default J-2
 Terminal characteristics, NDL default J-2,24
 Terminal classes J-1
 Terminal control commands 8-2
 Terminal definition commands 16-4; J-1
 Terminal definition file 8-6.1,7,8.1
 Terminal display mode 6-8.1
 Terminal identifier (TID) 2-12
 Terminal messages 7-40
 Terminal output
 Asynchronous 16-5
 Synchronous 16-5
 Terminal Verification Facility (TVF) 7-32; 8-18
 Terminating a host-terminal connection J-26
 Terminating input lines 16-3
 Termination character 16-6; J-8
 Termination sequence 16-4,6; C-14; J-8
 Terminator, command 5-2,3
 TERMLIB 8-6.1,7,8.1
 TEXT command 8-37
 Text editor facility 14-1
 Text editors 14-1,18
 Text mode 8-37
 TEXT record type 15-3
 TID (See Terminal identifier)
 Tieline Facility (TLF) 7-32; 8-18
 Time of the week 6-7
 Time slice C-14
 Timed/Event rollout C-14
 TIMEOUT command 8-9
 Timeout period 3-5
 Timeout status 8-9
 TLF (See Tieline Facility)
 TM command J-26
 Tracks 2-7
 Transaction Facility (TAF) 1-1; 3-6; 7-32; 8-18
 Transferring files between mainframes 9-31,49,50; 10-22
 Transmission key K-1
 Transmitting input K-1
 Transparent input mode C-14; J-11,18,27; K-1
 Transparent submit mode 7-68,70; C-14
 TRMDEF command 8-9; 16-2; J-2
 TTY (See Teletypewriter)
 TVF (See Tape verification facility)
 TVF (See Terminal Verification Facility)
 TYPE directive 15-30
 Type of network 6-8.1
 U-type record 15-1
 UCF record type 15-4
 UCLP (See Output, Lines printed)
 UCLV (See Output, Lines printed)
 UCPC (See Output, Cards punched)
 UEAC (See Application, Job usage)
 UEAD (See Adder activity)
 UECP (See Accumulated central processor time)
 UEEMS (See Mass storage activity)
 UEMT (See Magnetic tape activity)
 UEPF (See Permanent file activity)
 UHLA G-11
 UHLA label G-1
 UJN (See User job name)
 ULIB command 15-34
 ULIB directory 15-32,33
 ULIB record type 15-3,6
 Unlabeled tape 12-7,13
 UNLOAD command 7-54; 9-64
 UNLOCK command 9-65
 UPDATE command 14-14
 UPDATE mode 10-8
 UPL record type 15-15
 UPLx record type 15-4
 UPROC command 7-73
 US code A-3
 USECPU command 7-74
 User break processing 16-4
 User break 1 16-6; C-14; J-6,7

User break 2 16-6; C-14; J-8
 USER command 7-69,74
 User header label G-1
 User index 7-30; C-14
 User index hash 3-8; 7-14; C-14; D-2
 User job name (UJN) 3-8; 7-18,23,62;
 C-14; D-2
 User label G-11
 User library 2-17; 15-1,32,33,34,43
 User name 2-14; C-14
 User name, alternate 10-5
 User name LIBRARY 8-22,26
 User number C-14
 User permissions 7-31
 User program 1-5
 User prologue (See Prologue)
 User trailer label G-1
 User validation 3-8; 7-28,74
 User validation limits 7-5
 User volume label G-1
 Using XMODEM P-1
 UT terminal J-1
 UTLA G-11
 UTLA label G-1
 UVLA G-11
 UVLA label G-1

 Validation file C-14
 Validation limits 7-28
 VEIAF (See NOS/VE Interactive Facility)
 VERIFY command 9-65
 Verifying files 9-65; 15-35
 VFU load procedure H-2,4.2
 VFYLIB command 15-3,35
 VFYLIB directive 15-31
 Volume C-14
 Volume accessibility character, tape 12-10.1
 Volume header label G-1,2
 Volume serial number (VSN) 12-1,6,11,25;
 C-15; G-2
 VOL1 label 2-8,9; 12-1,9; G-1,2
 VSN command 12-25
 VSN (See Volume serial number)

 Wait disposition 2-11,12; 9-45
 Wait-if-busy option 10-6
 WBR command 11-10
 Weekday 6-7
 WHATJSN command 8-37
 WHILE command 6-31
 Word C-15
 WRITE command 8-51
 Write interlock 2-15; 9-26,65; C-15

 WRITE mode 10-8.1
 Write ring C-15

WRITEF command 9-68
 WRITEN command 8-52
 WRITER command 9-68

 X command 8-38
 X format I-3
 X format tapes 9-59,64
 X.25 Public data network 7-36,38
 X.25 terminals J-4
 XEDIT command 14-18
 XL command J-27
 XMODEM command 8-38; P-1
 X3.64 terminal J-1

 YANK status 15-6,15

 Zero byte terminator C-15
 Zero length PRU C-15
 ZZZDUMP file 3-12; 11-5,6,7
 ZZZZDMB file 11-1,3
 ZZZZTRM 8-6.1,7
 ZZZZZLD file 13-1
 ZZZZZxx files 13-1

 026 punch code A-3; F-3
 029 punch code A-3; F-3

 200 User Terminal 16-1; J-25
 2741 terminal, ATTN key 16-8
 5/7/9 multipunch F-3,5
 533/536 printer 9-48; A-4; H-1,2,4
 580 line printer
 Carriage control array H-7
 Format controls H-1
 PFC array H-1
 Spacing code 9-48.2
 585 line printer 9-48; H-1,2
 5870 Non Impact Printer 9-44; H-1
 6-bit display code 8-2; 9-23
 6/12-bit display code 8-2; 9-23
 6/7/8/9 multipunch 2-6; C-15; F-2
 6/7/9 multipunch 2-6; C-15; F-2
 6PP record type 15-4
 63-character set A-1
 64-character set A-1
 7-bit ASCII code 9-23; A-1
 7/8/9 multipunch 2-6; 3-1; C-15; F-2
 7/9 multipunch F-5
 7PP record type 15-4
 77 table (see Prefix table, record) 15-2
 7700 table (see Prefix table, record) 15-2
 9-54,55; 10-48,49
 95-character set A-1

ase fold on dotted line;
il edges with tape only.

FOLD

LD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA

Technical Publications

ARH219

4201 N. Lexington Avenue

Arden Hills, MN 55126-9983



COMMENT SHEET

MANUAL TITLE: CDC NOS Version 2 Reference Set,
Volume 3, System Commands
PUBLICATION NO.: 60459680 REVISION: L

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

REV. 5/86 PRINTED IN U.S.A.

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

FOLD ON DOTTED LINES AND TAPE

