60435700

CONTROL DATA CORPORATION

NOS VERSION 1 INSTALLATION HANDBOOK

CONTROL DATA®
CYBER 170 SERIES
MODELS 172, 173, 174, 175
CYBER 70 SERIES
MODELS 72, 73, 74
6000 SERIES
COMPUTER SYSTEMS

REVISION RECORD

REVISION	DESCRIPTION
A (6-6-75)	Manual released.
B (10-17-75)	Corrects various technical and typographical errors, documents changes for ALGOL 4.1, APL 1.2, COBOL 4.5, Sort/Merge 4.3, and FORTRAN Extended 4.5; adds information concerning SIFT, FORM 1.0, CYBER Database Control System 1.0, Database Utilities 1.0, Data Description Language 2.0, and Query Update 3.0, and also describes the operating system at corrective code level 411/410.
C (3-08-76)	Corrects various technical or typographical errors; documents changes for Export Import 1.2, Time-Sharing Module 1.2, TRANEX 1.3, FORTRAN Extended 4.6, Sort/Merge 4.4, CYBER Record Manager 1.4, CYBER Loader 1.1, and COMPASS 3.3; adds information concerning Multimainframe 1.0, COBOL 5.0, and Conversion Aids, COBOL 4 to COBOL 5 1.0; includes SHARE, LINK, PRESET, and AUTOLOAD CMRDECK entries; notes validation file and user accounting file changes; shows support of the 844-41 disk storage subsystem; and describes the NOS 1.1 operating system at corrective code level 419/420.
D (7-1-76)	Corrects various technical and typographical errors; documents changes for CYBER Loader 1.2; provides additional COBOL 5.0 information; adds information concerning order of installation, archive file preassignment, and 32K installation; and describes the NOS 1.1 operating system at corrective code level 430/428.
Publication No. 60435700	

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

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 Lexington Avenue North
St. Paul, Minnesota 55112
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	SFC †	REV
Cover		-
Title Page		-
ii		D
iii		D
iv		D
v/vi		D
vii		D
viii		D
ix/x		D
xi		D
xii		D
xiii		D
xiv		D
xv		D
xvi		D
xvii		D
xviii		D
xix		D
I-1-1		D
I-1-2		D
I-2-1		D
I-2-2		D
I-2-3		D
I-2-4		D
I-2-5		D
I-2-6		D
I-2-7		D
I-2-8		D
I-2-9		D
I-3-1		C
I-3-2		C
II-1-1		C
II-1-2		B
II-1-3		C
II-1-4		C
II-1-5		C
II-1-6		C
II-1-7		C
II-1-8		C
II-1-9		C
II-2-1		A
II-2-2		C
II-2-3		B
II-2-4		B
II-2-5		B
II-2-6		C
II-2-7		C
II-2-8		C
II-2-9		D
II-2-10		C
II-2-11		C
II-2-12		C
II-2-13		C
II-2-14		C
II-2-15		C
II-2-16		C
II-3-1		A
II-3-2		B
II-3-3		A

PAGE	SFC †	REV
II-3-4		C
II-3-5		A
II-3-6		A
II-3-7		A
II-3-8		A
II-3-9		A
II-3-10		A
II-3-11		A
II-3-12		B
II-3-13		C
II-3-14		C
II-3-15		C
II-4-1		C
II-4-2		A
II-4-3		C
II-4-4		C
II-4-5		C
II-4-6		A
II-4-7		A
II-4-8		C
II-4-9		C
II-4-10		A
II-4-11		D
II-4-12		C
II-4-13		A
II-4-14		C
II-4-15		C
II-4-16		C
II-4-17		A
II-4-18		C
II-4-19		C
II-4-20		C
II-4-21		C
II-4-22		C
II-4-23		C
II-4-24		C
II-4-25		C
II-4-26		C
II-4-27		C
II-4-28		C
II-4-29		C
II-4-30		C
II-4-31		C
II-4-32		C
II-4-33		C
II-4-34		C
II-4-35		C
II-4-36		C
II-4-37		C
II-4-38		C
II-4-39		C
II-4-40		C
II-4-41		C
II-4-42		C
II-4-43		C
II-4-44		C
II-4-45		C
II-4-46		C
II-4-47		C

PAGE	SFC †	REV
II-4-48		C
II-4-49		C
II-5-1		C
II-5-2		C
II-5-3		D
II-5-4		C
II-5-5		C
II-5-6		C
II-5-7		C
II-5-8		C
II-5-9		C
II-5-10		C
II-5-11		C
II-5-12		C
II-5-13		C
II-5-14		C
II-5-15		C
II-5-16		C
II-5-17		C
II-5-18		C
II-5-19		C
II-5-20		C
II-5-21		C
II-5-22		C
II-5-23		C
II-5-24		C
III-1-1		C
III-1-2		C
III-1-3		C
III-1-4		D
III-1-5		D
III-1-6		D
III-1-7		D
III-1-8		D
III-1-9		D
III-1-10		D
III-1-11		D
III-1-12		D
III-1-13		D
III-1-14		D
III-1-15		D
III-2-1		D
III-2-2		D
III-2-3		D
III-2-4		D
III-2-5		D
III-2-6		D
III-2-7		D
III-2-8		D
III-2-9		D
III-2-10		D
III-2-11		C
III-2-12		C
III-2-13		C
III-2-14		C
III-2-15		C
III-2-16		C
III-2-17		C
III-2-18		C

PAGE	SFC †	REV
III-2-19		C
III-2-20		C
III-2-21		C
III-2-22		C
III-2-23		C
III-2-24		C
III-2-25		C
III-2-26		C
III-2-27		A
III-3-1		C
III-3-2		C
III-4-1		D
III-4-2		C
III-4-3		D
III-4-4		C
III-4-5		A
III-4-6		C
III-4-7		C
III-4-8		B
III-4-9		C
III-4-10		A
III-4-11		D
III-4-12		D
III-4-13		C
III-4-14		C
III-4-15		A
III-4-16		C
III-4-17		A
III-4-18		B
III-4-19		C
III-4-20		D
III-4-21		C
III-4-22		C
III-4-23		C
III-4-24		B
III-4-25		D
III-4-26		C
III-4-27		C
III-4-28		D
III-4-29		D
III-4-30		D
IV-1-1		C
IV-1-2		C
IV-1-3		A
IV-1-4		D
IV-1-5		C
IV-1-6		C
IV-1-7		C
IV-1-8		C
IV-1-9		C
IV-1-10		C
IV-1-11		C
IV-1-12		C
IV-1-13		C
IV-1-14		A
IV-1-15		A
IV-1-16		B
IV-1-17		C
IV-1-18		C

†SFC Software Feature Change

PAGE	SFC†	REV
IV-1-19		C
IV-1-20		C
IV-1-21		C
IV-1-22		A
IV-1-23		A
IV-1-24		D
IV-1-25		C
IV-1-26		C
IV-1-27		C
IV-1-28		C
IV-1-29		D
IV-1-30		D
IV-1-31		D
IV-1-32		C
IV-1-33		D
IV-1-34		D
IV-1-35		C
IV-1-36		A
IV-1-37		A
IV-1-38		A
IV-1-39		A
IV-1-40		C
IV-1-41		C
IV-1-42		C
IV-1-43		C
IV-1-44		C
IV-1-45		C
IV-1-46		C
IV-1-47		C
IV-1-48		C
IV-1-49		A
IV-1-50		C
IV-1-51		C
IV-1-52		C
IV-1-53		D
IV-1-54		A
IV-1-55		C
IV-1-56		C
IV-1-57		C
IV-1-58		B
IV-1-59		A
IV-1-60		A
IV-1-61		C
IV-1-62		C
IV-1-63		C
IV-1-64		C
IV-1-65		A
IV-1-66		C
IV-1-67		A
IV-1-68		A
IV-1-69		A
IV-1-70		C
IV-1-71		A
IV-1-72		C
IV-1-73		A
IV-1-74		A
IV-1-75		A
IV-1-76		C
IV-1-77		A

PAGE	SFC†	REV
IV-1-78		A
IV-1-79		C
IV-1-80		A
IV-1-81		A
IV-1-82		A
IV-1-83		A
IV-1-84		A
IV-1-85		A
IV-1-86		C
IV-1-87		C
IV-1-88		C
IV-1-89		A
IV-1-90		A
IV-1-91		C
IV-1-92		C
IV-1-93		C
IV-2-1		A
IV-2-2		A
IV-2-3		C
IV-2-4		A
IV-2-5		A
IV-2-6		A
IV-2-7		A
IV-2-8		A
IV-2-9		C
IV-2-10		A
IV-2-11		A
IV-2-12		A
IV-2-13		A
IV-2-14		A
IV-2-15		A
IV-2-16		A
IV-2-17		A
IV-2-18		C
IV-2-19		A
IV-2-20		A
IV-2-21		A
IV-2-22		A
IV-2-23		C
IV-2-24		C
IV-2-25		D
IV-2-26		C
IV-2-27		C
IV-2-28		C
IV-2-29		C
IV-2-30		C
IV-2-31		C
IV-2-32		C
IV-2-33		D
IV-2-34		D
IV-2-34.1/2		D
IV-2-35		C
IV-2-36		C
IV-2-37		C
IV-2-38		C
IV-2-39		C
IV-2-40		C
IV-2-41		C

PAGE	SFC†	REV
IV-2-42		C
IV-2-43		C
IV-2-44		C
IV-2-45		C
IV-2-46		C
IV-2-47		C
IV-2-48		C
IV-2-49		C
IV-2-50		C
IV-2-51		D
IV-2-52		C
IV-2-53		C
IV-2-54		C
IV-2-55		C
IV-2-56		C
IV-2-57		C
IV-2-58		C
IV-2-59		C
IV-2-60		C
IV-2-61		C
IV-2-62		C
IV-3-1		C
IV-3-2		C
IV-3-3		C
IV-3-4		C
IV-3-5		C
IV-3-6		C
IV-3-7		C
IV-3-8		C
IV-4-1		A
IV-4-2		A
IV-4-3		B
IV-4-4		A
IV-4-5		A
IV-4-6		A
IV-4-7		A
IV-4-8		A
IV-4-9		A
IV-4-10		A
IV-4-11		A
IV-4-12		A
IV-4-13		A
IV-4-14		A
IV-4-15		A
IV-4-16		A
IV-4-17		B
IV-4-18		B
IV-4-19		A
IV-4-20		A
IV-4-21		A
IV-4-22		A
IV-4-23		A
IV-4-24		D
IV-4-25		A
IV-4-26		A
IV-4-27		A
IV-4-28		C
IV-4-29		A
IV-4-30		B

PAGE	SFC†	REV
IV-4-31		C
IV-4-32		C
IV-4-33		C
IV-4-34		C
IV-4-35		C
IV-4-36		C
IV-4-37		C
IV-4-38		C
IV-4-39		C
IV-5-1		C
IV-5-2		C
IV-5-3		C
IV-5-4		A
IV-5-5		C
IV-5-6		C
IV-5-7		C
IV-5-8		D
IV-5-9		C
IV-5-10		C
IV-5-11		C
IV-6-1		A
IV-6-2		A
IV-6-3		A
IV-6-4		A
IV-6-5		A
IV-6-6		A
IV-6-7		A
IV-6-8		A
IV-6-9		A
IV-7-1		C
IV-7-2		C
IV-7-3		C
IV-7-4		C
IV-7-5		C
IV-7-6		C
IV-7-7		C
IV-7-8		C
IV-7-9		C
IV-7-10		C
IV-7-11		C
IV-7-12		C
IV-7-13		C
IV-8-1		D
IV-8-2		C
IV-8-3		C
IV-8-4		C
IV-8-5		D
IV-8-6		C
IV-8-7		C
IV-8-8		C
A-1		D
A-2		D
A-3		D
Index-1		D
Index-2		D
Index-3		D
Index-4		D
Index-5		D
Index-6		D
Index-7		D

†SFC Software Feature Change

PAGE	SFC †	REV	PAGE	SFC †	REV	PAGE	SFC †	REV	PAGE	SFC †	REV
Index-8		D									
Index-9		D									
Index-10		D									
Index-11		D									
Index-12		D									
Index-13		D									
Cmt Sheet		D									
Return Env		-									
Cover		-									

†SFC Software Feature Change

AA 5935

PREFACE

This handbook provides an analyst with the information needed to install the Network Operating System (NOS). The analyst who makes this installation should have at least 6 months experience with a CONTROL DATA® 6000, CYBER 70, or CYBER 170 computer system.

To meet these needs, the NOS 1.1 Installation Handbook has the following design.

Part I	Lists and describes all tapes released with the base NOS 1.1 system and those optional products that can be ordered separately.
Part II	Contains the procedure for deadstarting the system.
Part III	Contains general and specific installation information for the operating system and each product, such as the standard installation parameters and examples of jobs necessary to modify the operating system and product set.
Part IV	Contains information necessary to maintain a system once it is installed; it covers special system files, VALIDUs, PROFILa (used for a user validation), NETWORK (used to describe the communications network), permanent file utilities, Time-Sharing Stimulator, Deadstart Dump Interpreter (DSDI), Status Control Register Simulator, 881 Pack Formatting, and multimainframe operation.

The following manuals contain additional information on the NOS Version 1 supported software and hardware.

NOS PRODUCT MANUALS

Control Data Publication	Publication Number
NOS General Information Manual	60435900
NOS Reference Manual Volume 1	60435400
NOS Reference Manual Volume 2	60445300
NOS Applications Programmer's Instant Manual	60436000
NOS System Programmer's Instant Manual	60449200
NOS Operator's Guide	60435600
Common Utilities Reference Manual	60495600
COMPASS 3 Reference Manual	60492600
COMPASS 3 Instant	60497900
COMPASS 3 Instruction Card	60493000
Loader Reference Manual	60429800
Loader Instant	60449800
Modify Reference Manual	60450100
Modify Instant	60450200

Control Data Publication	Publication Number
NOS Export/Import Reference Manual	60436200
Record Manager 1 Reference Manual	60495700
NOS Text Editor Reference Manual	60436100
NOS Time-Sharing User's Reference Manual	60435500
NOS Time-Sharing User's Guide	60436400
NOS Terminal User's Instant	60435800
On-Line Maintenance Software Reference Manual	60436600
SIFT Programming Systems Bulletin	60496500
Update Reference Manual	60449900
Update Instant	60450000
Application Installation Handbook	76071100
8-Bit Subroutines 1 Reference Manual	60495500

OPTIONAL PRODUCT MANUALS

Control Data Publication	Publication Number
ALGOL 4 Reference Manual	60496600
APL *CYBER Reference Manual	19980400
BASIC 3 Reference Manual	19983900
CDC CYBER Database Control System 1 Reference Manual	60498700
COBOL 4 Reference Manual	60496800
COBOL 4 Instant	60497000
Database Utilities 1 Reference Manual	60498800
DDL 1 Reference Manual	60359000
DDL 2 Reference Manual Volume 1 Schema Definition	60498400
DDL 2 Reference Manual Volume 2 COBOL Sub-schema Definition	60498600
DDL2 Reference Manual Volume 3 QU Sub-schema Definition	60498500
FORTRAN 2.3 Reference Manual	60174900
FORTRAN 2.3 Instant	60189500
FORTRAN Extended 4 Reference Manual	60497800
FORTRAN Extended 4 Instant	60497900
FORTRAN Extended Debug User's Guide	60498000
Query Update 2 Reference Manual	60384900
Query Update 3 Reference Manual	60498300
SIMULA 1 Reference Manual	60234800
SIMULA 1 Instant	60235100
Sort/Merge 4 Reference Manual	60497500
Sort/Merge 4 Instant	60497600
SYMPL 1 Reference Manual	60496400
TRANEX Reference Manual	60407900

HARDWARE MANUALS

Control Data Publication	Publication Number
CYBER 170 Computer System Reference Manual	60420000
CYBER 70/Model 72 Computer System Reference Manual	60347000
CYBER 70/Model 73 Computer System Reference Manual	60347200
CYBER 70/Model 74 Computer System Reference Manual	60347400
6400/6500/6600 Computer Systems Reference Manual	60100000

NOS 1.1 and its product set are intended to be used only as described in this document. Control Data is not responsible for the proper functioning of undescribed features or undefined parameters.

CONTENTS

PART I
INTRODUCTION

PART II
DEADSTART

PART III
INSTALLATION OF CONFIGURED SYSTEM AND PRODUCT SET

PART IV
SYSTEM MAINTENANCE

APPENDIX

FIGURES

TABLES

SYSTEM AND MANUAL DESCRIPTION 1

1.1 RELEASE DESCRIPTION

1.1.1 LIST OF PRODUCTS

NOS VERSION 1 PRODUCTS

The following products are on the NOS deadstart tape. Description of all materials released with the NOS base package is in part I, section 2.1.

NOS 1.1
COMPASS 3.3
Modify 1.0
CYBER Record Manager 1.4
CYBER Loader 1.2
Product texts, COMDECKS, and programs
Text Editor
8-Bit Subroutines 1.0
Update 1.2 and CYBER utilities

The maintenance package is also available for use with the base package, but it must be ordered separately. The maintenance package consists of the maintenance tools (STIMULA, 1TS, DFSORT, FTNTRAN, PSAMP, SMP, DSDI, SIFT 1.0, games, and diagnostic programs), SYMPL 1.1, CEDIAG 1.1, and 881/883 Pack Formatting.

OPTIONAL PRODUCTS

The following products can be ordered separately. The release materials for these products are defined in part I, sections 2.2 through 2.19.

ALGOL 4.1
APL 1.2
BASIC 3.1
COBOL 4.5
COBOL 5.0
Conversion Aids, COBOL 4 to COBOL 5 1.0
Conversion Aids System 1.2
CYBER Database Control System 1.0
Database Utilities 1.0
Data Description Language 1.0
Data Description Language 2.0
Export/Import 1.2
FORM 1.0
FORTRAN Extended 4.6
Multimainframe 1.0
Query Update 2.1
Query Update 3.0
SIMULA 1.1
Sort/Merge 4.4
Time-Sharing Module 1.2
TRANEX 1.3

Additional application products are also available with NOS. Refer to the Application Installation Handbook for products and release materials available.

RELEASE MATERIALS DESCRIPTION 2

This section lists the tapes released with the operating system package and each optional product. All tapes are released in 64-character set mode and in internal (F=I) format.

2.1 NOS VERSION 1

The tapes released as part of NOS include the following products.

Deadstart tape	Binaries for NOS	section 2.1.1
REL0	Installation decks program library, system and product set modification and PSRs, and notes and cautions, if any	section 2.1.1
REL1A	Program library for NOS	section 2.1.1
REL1E	Binary and program library for Product texts and CYBER Loader	section 2.1.1
REL3A	Binary and program libraries for COMPASS, Update, and CYBER utilities	section 2.1.2
REL3B	Binary and program library for CYBER Record Manager	section 2.1.3
REL3D	Binary and program library for FORM	section 2.1.4

The maintenance package, which must be ordered separately from the NOS package, is also available. The maintenance package includes the following products.

REL2A	Binaries and program libraries for SIFT and the various maintenance tools whose program libraries are in Modify format	section 2.2.1
REL2B	Binary and program library for CEDIAG	section 2.2.2
REL2C	Binary and program library for 881/883 Pack Formatting	section 2.2.3
REL2E	Binary and program library for SYMPL	section 2.2.4

2.1.1 OPERATING SYSTEM

Deadstart tape — The NOS deadstart tape contains binaries of:

NOS
COMPASS
CYBER Loader
Modify
CYBER Record Manager
Product texts
Text Editor
Update and CYBER utilities

The deadstart tape has the following characteristics: unlabeled, 7- or 9-track, 800 bpi, binary recording mode, one file.

REL0	REL0 is used in the installation and modification of the operating system and product set. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, INSTALL*NOS430428 as file id in HDR1 label, six files.

File 1	Procedure to install the REL0 files, binary of NOTE, and installation decks program library
File 2	System modifications in Modify format
File 3	Product set modifications in Update format
File 4	PSR data base
File 5	Product set notes and cautions, if any
File 6	Operating system notes and cautions, if any

REL1A

The NOS system OPL is a program library in Modify format and includes the following.

NOS
Modify
Text Editor

It has the following characteristics: labeled, 7- or 9-track, Modify format, 800 bpi, binary recording mode, one file, NOS1P1*NOS430428 as file id in HDR1 label.

REL1E

REL1E is a program library in Update format that contains the Product texts, Product texts I/O, and CYBER Loader.

REL1E has the following characteristics: labeled, 7- or 9-track, Update format, 800 bpi, binary recording mode, LDR1P2*NOS430428 as file id in HDR1 label, nine files.

File 1	Program library for the following decks: CPCTEXT, IPTEXT, PPTEXT, SCPTEXT, and CPUTEXT
File 2	Empty file
File 3	Binaries of CPCTEXT, IPTEXT, PPTEXT, SCPTEXT, and CPUTEXT
File 4	Program library for the following decks: PFMTEXT, CPC, IORANDM, IO, CHEKPT, and RECOVR
File 5	Binary of PFMTEXT
File 6	Binaries of SYS.RM, CHEKPT, CPC, RECOVR, IORANDM, and IO that are placed on the library file SYSIO
File 7	Program library for CYBER Loader
File 8	Binary of CYBER Loader
File 9	Binaries of UCLOAD and TRAPPER

2.1.2 COMPASS 3.3, UPDATE 1.2, AND CYBER UTILITIES

REL3A

REL3A contains program libraries in Update format for COMPASS, Update, and CYBER utilities COPYL and ITEMIZE. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, CMP*UPD*NOS430428 as file id in HDR1 label, six files.

File 1	Program library for COMPASS
File 2	Absolute binary code of COMPASS
File 3	Empty file
File 4	Program library for Update and CYBER utilities
File 5	Absolute binary code of Update and CYBER utilities
File 6	Empty file

2.1.3 CYBER RECORD MANAGER (CRM) 1.4, 8-BIT SUBROUTINES 1.0

REL3B — REL3B is a tape which contains the program libraries in Update format for CRM and 8-Bit Subroutines. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, CRM1P4*NOS430428 as file id in HDR1 label, nine files.

File 1	Program library for Basic Access Module
File 2	Binary code of overlay files for Basic Access Module
File 3	I/O modules binary code for Basic Access Module
File 4	Program library for Advanced Access Module
File 5	Binary code of overlay files for Advanced Access Module
File 6	I/O modules binary code for Advanced Access Module
File 7	Program library for 8-Bit Subroutines
File 8	Binary code of overlay programs for 8-Bit Subroutines
File 9	I/O modules binary code for 8-Bit Subroutines

2.1.4 FORM 1.0

REL3D — REL3D is a program library in update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, FORM1P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for FORM
File 2	Relocatable routines for FORM
File 3	Library routines for FORM

2.2 MAINTENANCE PACKAGE

2.2.1 MAINTENANCE TOOLS

REL2A — REL2A contains program libraries in Update and Modify format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, TOOLS2A*NOS430428 as file id in HDR1 label, six files.

File 1	Program library in Modify format for STIMULA, 1TS, DFSORT, DSDI, PSAMP, SMP, all games, and all diagnostics excluding deadstart diagnostics
File 2	Binaries of file 1
File 3	Empty file
File 4	Program library in Update format for SIFT
File 5	Binaries of SIFT
File 6	Empty file

2.2.2 CEDIAG 1.1

REL2B — REL2B is a program library in update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, TOOLS2B*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for CEDIAG
File 2	Absolute binary code of CEDIAG
File 3	Empty file

2.2.3 881/883 PACK FORMATTING

REL2C — REL2C is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, TOOLS2C*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for 881/883 Pack Formatting
File 2	Absolute binary code of file 1
File 3	Empty file

2.2.4 SYMPL 1.1

REL2E — REL2E is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, binary recording mode, 800 bpi, TOOLS2E*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for SYMPL
File 2	Absolute binary code of SYMPL compiler overlays
File 3	Relocatable binary code of SYMPL object library

2.3 OPERATING SYSTEM MODULES

2.3.1 TIME-SHARING MODULE 1.2

REL1B — REL1B is a program library in Modify format for the optional product Time-Sharing Module 1.2. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, TLX1P2*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for Time-Sharing Module
File 2	Time-Sharing Module binaries
File 3	Empty file

2.3.2 EXPORT/IMPORT 1.2

REL1C

REL1C is a program library in Modify format for the optional product Export/Import 1.1. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, EI21P2*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for Export/Import
File 2	Export/Import binaries
File 3	Empty file

2.3.3 TRANEX 1.3

REL1D

REL1D is a program library in Modify format for the optional product TRANEX. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, Modify format, binary recording mode, TRX1P3*NOS430428 as file id in HDR1 label, three files.

File 1	Program library of TRANEX
File 2	Absolute binary code of transaction subsystem and user libraries
File 3	Relocatable binary code of library routines

2.3.4 MULTIMAINFRAME 1.0

REL1F

REL1F is a program library in Modify format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, MMF1P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for Multimainframe
File 2	Multimainframe binaries
File 3	Empty file

2.4 FORTRAN EXTENDED 4.6

REL4A

REL4A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, FTN4P6*NOS430428 as file id in HDR1 label, six files.

File 1	Program library for FORTRAN Extended
File 2	Absolute binary code of compiler overlays
File 3	Empty file
File 4	Program library for FORTRAN Common Library
File 5	Empty file
File 6	Relocatable binary code of library routines

2.4.1 FORTRAN EXTENDED 4.6 WITH INTERACTIVE OPTION

REL4B

REL4B is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, FTNI4P6*NOS430428 as file id in HDR1 label, six files.

File 1	Program library for FORTRAN Extended with Interactive Option
File 2	Absolute binary code of compiler overlays
File 3	Empty file
File 4	Program library for FORTRAN Common Library
File 5	Empty file
File 6	Relocatable binary code for library routines

2.5 COBOL 4.5

REL5A — REL5A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, COB4P5*NOS430428 as file id in HDR1 label, three files.

File 1	Program library of COBOL
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines

2.6 CONVERSION AIDS SYSTEM 1.2

REL5B — REL5B is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, CAS1P2*NOS430428 as file id in HDR1 label, 16 files

File 1	Program library for LCAS
File 2	Absolute load module for LCAS
File 3	Binary (FORTRAN) syntax file for LCAS
File 4	Program library for FCAS (Update)
File 5	Binary data file for FORTRAN FCP verification
File 6	Binary data file for COBOL FCP verification
File 7	Binary data file for COSY-to-Update FCP verification
File 8	Binary data file for COSY-to-Update FCP verification (file 2 of 2)
File 9	Absolute load module CBLFCP1 for COBOL FCP
File 10	Absolute load module CBLFCP2 for COBOL FCP
File 11	Absolute load module FTNFCP1 for FORTRAN FCP
File 12	Absolute load module FTNFCP2 for FORTRAN FCP
File 13	Absolute load module COUP for COSY-to-Update FCP
File 14	Absolute load module COUP for COBOL-COPY-LIBRARY FCP
File 15	Binary (FORTRAN) syntax file CBLFCPM for COBOL FCP
File 16	Binary (FORTRAN) syntax file FTNFCPM for FORTRAN FCP

2.7 COBOL 5.0

REL5C — REL5C is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, COB5P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for COBOL
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines

2.8 CONVERSION AIDS, COBOL 4 TO COBOL 5 1.0

REL5D — REL5D is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, C4C51P0*NOS430428 as file id in HDR1 label, six files.

File 1	Program library for Conversion Aids
File 2	Absolute binary of the Language Conversion Processor

File 3	Absolute binary of the COPY utility
File 4	Binary syntax file for the Language Conversion Processor
File 5	Empty file
File 6	Empty file

2.9 SORT/MERGE 4.4

REL6A

REL6A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SORT4P4*NOS430428 as file id in HDR1 label, three files.

File 1	Program library of Sort/Merge
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines

2.10 ALGOL 4.1

REL7A

REL7A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, ALG4P1*NOS430428 as file id in HDR1 label, six files.

File 1	Program library for ALGOL
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines
File 4	Program library for ALGEDIT
File 5	Binaries for ALGEDIT
File 6	Empty file

2.11 BASIC 3.1

REL8A

REL8A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, BAS3P1*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for BASIC
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines

2.12 APL 1.2

REL9A

REL9A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, APL1P2*NOS430428 as file id in HDR1 label, six files.

File 1	Program library for APL
File 2	Absolute binary code of overlay loader
File 3	Absolute binary code for terminal processing
File 4	Absolute binary code for batch processing
File 5	Absolute binary code for terminal file processing
File 6	Absolute binary code for batch processing

2.13 SIMULA 1.1

REL10A

REL10A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SIMU1P1*NOS430428 as file id in HDR1 label, three files.

File 1	Program library of SIMULA
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines

2.14 QUERY UPDATE 2.1

REL11A

REL11A is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, QU2P1*NOS430428 as file id in HDR1 label, three files.

File 1	Program library of Query Update
File 2	Absolute binary code of Query Update
File 3	Library routines for SYSMISC

2.15 DATA DESCRIPTION LANGUAGE 1.0

REL11B

REL11B is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, DDL1P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library of Data Description Language
File 2	Absolute binary code of Data Description Language
File 3	Library routines for SYSMISC

2.16 CYBER DATABASE CONTROL SYSTEM 1.0

REL11C

REL11C is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, CDCS1P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for CDCS
File 2	Absolute binary code of CDCS
File 3	Binary code of library routines for CDCS

2.17 DATABASE UTILITIES 1.0

REL11D

REL11D is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, DBU1P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for DBU
File 2	Absolute binary code of DBU
File 3	Binary code of library routines for DBU

2.18 QUERY UPDATE 3.0

REL11E

REL11E is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, QU3P0*NOS430428 as file id in HDR1 label, three files.

File 1	Program library for QU 3
File 2	Absolute binary code of QU 3
File 3	Library routines for SYSMISC

2.19 DATA DESCRIPTION LANGUAGE 2.0

REL11F

REL11F is a program library in Update format. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, DDL2P0*NOS430428 as file id in HDR1 label, three lines.

File 1	Program library for DDL
File 2	Absolute binary code of DDL
File 3	Binary code of library routines for DDL

INSTALLATION OF A CONFIGURED SYSTEM AND PRODUCT SET 3

To obtain a running system that is configured according to the requirements of a particular installation site, use the following general procedure along with the references to specific sections for detailed procedures.

1. Deadstart using the released version of the deadstart tape.	Tape description in part I, section 2.1 Deadstart procedure in part II, section 1
2. Create VALIDUs† (and PROFILa† and NETWid†† files, if needed). To use the system without VALIDUs, turn off user validation. Type DISABLE, VALIDATION. Press CR **NOTE** The system must be in UNLOCK mode to do this.	Part IV, section 1
3. Create a new deadstart tape to include customer requirements.	
a. Possible customer requirements for the operating system are· changing installation parameter settings; creating new or modified CMRDECK, IPRDECK, and LIBDECK; and updating the system to the current PSR level.	Modification information in part III, section 2 Procedure in part III, section 1
b. Possible customer requirements for the product set are adding the optional products to the deadstart tape; changing installation parameter settings; and updating products to the current PSR level.	Modification information for each product in part III, sections 3 and 4 Procedure in part III, section 1

† In this manual, user validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

†† The network description file is referred to as NETWid, where id is the 2-character machine identification.

4. Deadstart using the configured version of the deadstart tape. — Part II, section 1

5. If only a batch environment is required, it is recommended that automatic job rollout be disabled.

   Type DISABLE, AUTOROLL.
   Press CR

ADDITIONAL MANUALS 4

The following manuals contain additional information on the NOS Version 1 supported software and hardware.

4.1 MANUALS FOR PRODUCTS IN NOS PACKAGE

Control Data Publication	Publication Number
NOS General Information Manual	60435900
NOS Reference Manual Volume 1	60435400
NOS Reference Manual Volume 2	60445300
NOS Applications Programmers Instant Manual	60436000
NOS System Programmers Instant Manual	60449200
NOS Operator's Guide	60435600
Common Utilities Reference Manual	60493300
COMPASS 3 Reference Manual	60492600
COMPASS 3 General Information Manual	
COMPASS 3 Instant	60492800
COMPASS 3 Instruction Card	60493000
Loader Reference Manual	60429800
Loader Instant	60449800
Modify Reference Manual	60281700
Modify Instant	60283000
NOS Export/Import Reference Manual	60436200
Record Manager 1 Reference Manual	60495700
NOS Text Editor Reference Manual	60436100
NOS Time-Sharing User's Reference Manual	60435500
NOS Terminal User's Instant	60435800
On-Line Maintenance Software Reference Manual	60436600
SIFT Programming Systems Bulletin	60496500
Update Reference Manual	60449900
Update Instant	60450000
Application Installation Handbook	76071100
8-Bit Subrotuines 1 Reference Manual	60495500

4.2 MANUALS FOR OPTIONAL PRODUCTS

Control Data Publication	Publication Number
ALGOL 4 Reference Manual	60496600
APL *CYBER Reference Manual	19980400
BASIC 3 Reference Manual	19983900
CDC CYBER Database Control System 1 Reference Manual	60498700
COBOL 4 Reference Manual	60496800
COBOL 4 Instant	60497000
Database Utilities 1 Reference Manual	60498800
DDL 1 Reference Manual	60359000
DDL 2 Reference Manual Volume 1 Schema Definition	60498400
DDL 2 Reference Manual Volume 2 COBOL Sub-schema Definition	60498600
DDL2 Reference Manual Volume 3 QU Sub-schema Definition	60498500
FORTRAN 2.3 Reference Manual	60174900
FORTRAN 2.3 Instant	60189500
FORTRAN Extended 4 Reference Manual	60497800
FORTRAN Extended 4 Instant	60497900
FORTRAN Extended Debug User's Guide	60498000
Query Update 2 Reference Manual	60384900
Query Update 3 Reference Manual	60498300
SIMULA 1 Reference Manual	60234800
SIMULA 1 Instant	60235100
Sort/Merge 4 Reference Manual	60497500
Sort/Merge 4 Instant	60497600
SYMPL 1 Reference Manual	60496400
TRANEX Reference Manual	60407900

4.3 HARDWARE MANUALS

Control Data Publication	Publication Number
CYBER 170 Computer System Reference Manual	60420000
CYBER 70/Model 72 Computer System Reference Manual	60347000
CYBER 70/Model 73 Computer System Reference Manual	60347200
CYBER 70/Model 74 Computer System Reference Manual	60347400
6400/6500/6600 Computer Systems Reference Manual	60100000

SUMMARY OF DEADSTART 1

Deadstart is the process of loading the current version of the NOS deadstart tape into core memory and mass storage from a 657, 659, 667, or 669 magnetic tape unit. The deadstart tape consists of a single binary file of programs recorded as logical records. These logical records are divided into groups that are separated by zero-length records. Refer to part III, section 2.10 to obtain a system catalog.

If the deadstart tape is mounted on a 667 or 669 magnetic tape unit, special procedures (coldstart or warmstart) must be followed. The procedures for coldstart and warmstart differ from the deadstart procedures only until the first record (PRL) is read from the deadstart tape. Then, deadstart, coldstart, and warmstart are equivalent.

1.1 DEADSTART PROCEDURE

The deadstart process includes the following actions if the deadstart tape will be mounted on a 657 or 659 magnetic tape unit.

- Mounting the deadstart tape
- Setting the deadstart panel to indicate:

  The appropriate CMRDECK

  The deadstart function (maintenance deadstart, automatic system deadstart, or system deadstart with options)

  The system devices if the CMRDECK and IPRDECK will not be modified
- Activating the DEADSTART switch
- Initializing the system
- Initiating job processing

NOTE

The controller for the tape unit must be on a channel without a PPU (that is, channel 12, 13, 32, or 33).

The following outline lists the procedure necessary to deadstart. It references sections containing detailed instructions.

Mount Deadstart Tape

1.	Ensure that required mass storage devices have packs mounted and/or are available. Ensure that all line printers are equipped with a carriage control tape, if required.	Refer to appendix A of the operator's guide for carriage control tape formats.
2.	Mount current deadstart tape on tape unit to be specified on deadstart panel (words 1 through 11 specify tape unit, controller, and channel number used to access deadstart tape): ready the unit.	Refer to appendix D of the operator's guide for instructions for mounting the deadstart tape.

Set Deadstart Panel

3. Set deadstart panel for deadstart tape loading and select (in word 13) the LIBDECK, CMRDECK, and deadstart function (system or maintenance deadstart). If system deadstart is selected, also specify (in word 14) the level of deadstart, CPU options, and system devices (if CMRDECK and IPRDECK are not to be modified).

   Part II, section 2 describes deadstart panel selections.

   a. Set MODE switch to LOAD

   b. Set toggle switches of deadstart panel

4. Momentarily activate either the DEADSTART switch on the deadstart panel or the switch on the system console.

5. The information displayed at the system console depends upon the deadstart function chosen in word 13, yyy.

   Maintenance Deadstart (yyy=010 or 011)

   Note specific information for 010 (PPU 0 memory display) and 011 (load deadstart dump program).

   Specific information is in part II, section 2.3 under yyy=010 and 011.

   System Deadstart With Deadstart Options Displayed (yyy=001)

   Select one or more of the following.

   The deadstart option display is described in part II, section 3.1.

   a. Select dump or test option. Continue with step 4 after any of these options.

      These options are defined in part II, sections 3.3 through 3.4 and 3.7 through 3.9.

   b. Select one or more options to change the deadstart panel settings. Then, continue with the CR option.

      These options are defined in part II, sections 3.6 and 3.10 through 3.14.

   c. Press CR to select the load system option

      Refer to part II, section 3.2.

   Deadstart continues according to the same dependencies as noted in Automatic System Deadstart.

   Automatic System Deadstart (yyy=000)

   Deadstart continues as follows:

   a. If any of the bits 0 through 5 of word 14 are currently set, the system library is loaded onto the system devices after memory confidence checking is complete. Continue with step 9; there are no further options before initializing the system.

      Refer to part II, section 3.2.1. Memory confidence checking is described in section 2.3 and in the operator's guide.

   b. If bits 0 through 5 of word 14 are currently 000 000, CMRINST is displayed. Continue with step 6.

      Refer to part II, section 3.2.2.

6. CMRDECK modification

CMRINST appears on the system console display; alternate between the CMRINST and the CMRDECK displays by pressing the right blank key. If either the CMRDECK or CMRINST overflows two screens, the display can be advanced by pressing the + key.

CMRDECK is described in part II, section 4.

Enter any CMRDECK information (when CMRINST or CMRDECK are displayed) or continue with step 7.

7. IPRDECK modification

To load the system library without displaying IPRINST or IPRDECK, continue with step 8.

IPRDECK is described in part II, section 5.

To display and modify the current IPRDECK (while CMRDECK or CMRINST is displayed):

Type NEXT.

Press CR

IPRINST is displayed; alternate between the IPRINST and the current IPRDECK displays by pressing the right blank key. (The current IPRDECK is specified in the current CMRDECK.) If either the IPRDECK or IPRINST overflows two screens, the display can be advanced by pressing the + key.

Enter any IPRDECK information and/or continue with step 8.

8. To indicate that all modifications to the CMRDECK and IPRDECK are completed:

Type GO.

Press CR

9. If the system deadstart level is 0 or 2, the system is loaded on the current system devices; if no system devices are specified, the system is automatically loaded on the EST ordinal 0 device.

If level 1, the system library is recovered from the last checkpoint; if level 3, the system library is recovered from the current system device and RPL is recovered from the current central memory residence.

If a deadstart error occurs, a message appears on the right screen. If the system is loaded from the deadstart tape, the decks being loaded are displayed on the right screen as deadstart status messages. If the screen is blank, check the deadstart panel for correct settings.

Initialize the System

10. The stop on PP memory parity error control bit is set in the status/control register if the machine is a CDC CYBER 170. Detection of a PP memory parity error during system operation (on a CDC CYBER 170) results in the display of an error message

    PP MEMORY PARITY ERROR.

    at the system control point.

11. DSD requests on the left screen that the date and time be entered.

    It is possible to recover the date and time from the previous deadstart if this is a level 3 deadstart. However, it is recommended to enter the correct date and time with each deadstart because the system enters the date and time into the system dayfile, error log dayfile, and account dayfile along with the appropriate messages.

    DSD requests the date:

        ENTER DATE YY/MM/DD.

12. Type yy/mm/dd.
    Press CR

yy	Year; 00 through 99
mm	Month; 01 through 12
dd	Day; 01 through n; n is number of days in month

    To recover previous date on a level 3 deadstart, press CR. For other levels, pressing CR causes the system to set the date to when the deadstart tape was created.

13. When DSD accepts the date entry, it requests the time:

        ENTER TIME HH.MM.SS.

14. Type hh.mm.ss.
    Press CR

hh	Hour; 00 through 23
mm	Minute; 00 through 59
ss	Second; 00 through 59

    To recover the time from the previous deadstart (time entered at original deadstart plus time accumulated until deadstart button is pressed, which causes the system to cease operation) on a level 3 deadstart, press CR. On other levels, pressing CR causes the system to set the time to 00.00.00.

Initiate Job Processing

15. If the DSD commands, AUTO, MAINTENANCE, and ISF, are part of the IPRDECK as is recommended, then if loading from a deadstart tape, the first of these commands appears on the left screen after the date and time are entered. If the tape load is not complete, these DSD commands may be bypassed by pressing the left blank key until the flashing entry is clear; continue with step 18.

    Refer to operator's guide.

    If these commands are not part of the IPRDECK, then to continue, initiate job processing by assigning control points to the standard packages:

    Type AUTO.

    Press CR

    Also, to activate system files VALIDUs† (user validation), PROFILa† (user accounting), RSXDid and RSXVid (resource management control files, where id is the machine id):

    Type X.ISF.

    Press CR

16. When the deadstart tape is rewound to load point, deadstart is complete. The deadstart tape is not required again during system operation unless another deadstart is necessary. It is possible to clear, unload, and remove the deadstart tape to use the tape unit for other operations.

17. If the entries in the IPRDECK do not suit the local environment, alter the job control parameters by entering DSD commands.

    Refer to operator's guide.

18. If this deadstart procedure used the deadstart tape released with the system, then continue with the general procedure to construct a deadstart tape that is modified to represent the customer's requirements.

    Part I, section 3 describes how to create a modified and/or updated deadstart tape.

    If this deadstart procedure used a deadstart tape already modified to represent the customer's requirements, the system is now ready for use.

19. The system provides a default selection of all CPU hardware and program error exit modes prior to initiation of each job [that is, MODE(7,7)] .

    Refer to volume 1 of the system reference manual for a description of the MODE control statement.

† In this manual, user validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

1.2 COLDSTART PROCEDURE

Coldstart is a method of deadstart used when the deadstart tape is mounted on a 667 or 669 magnetic tape unit. A coldstart procedure is necessary to deadstart and load the 667/669 controlware and then load the system. This process includes its own deadstart panel setting and a program (ABC) to read the controlware deck from the card reader, load it into the controller, and initiate the system deadstart. Refer to part II, section 2 for panel settings.

The coldstart process includes the following actions.

- Mounting the deadstart tape
- Setting the deadstart panel to indicate:

  The appropriate CMRDECK and LIBDECK

  The deadstart function (maintenance deadstart, automatic system deadstart, or system deadstart with options)

  The system devices if the CMRDECK and IPRDECK are not to be modified
- Preparing the card reader
- Activating the deadstart switch
- Activating the card reader
- Initializing the system
- Initiating job processing

NOTE

The card reader and the 667/669 tape unit must be on different channels when performing a coldstart. In addition, they must be on channels without PPUs (that is, channel 12, 13, 32, or 33).

The following outline lists the procedure necessary to coldstart. It references sections containing detailed instructions.

Mount Deadstart Tape

1. Ensure that required mass storage devices have packs mounted and/or are available.

2. Mount current deadstart tape on a 667 or 669 magnetic tape unit to be specified on the deadstart panel (words 5 and 7 specify tape unit and channel number used to access deadstart tape); ready the unit.

   Refer to appendix D of the operator's guide for instructions for mounting the deadstart tape.

Set Deadstart Panel

3. Set deadstart panel for deadstart tape loading and select (in word 13) the LIBDECK, CMRDECK, and deadstart function (system or maintenance deadstart).

   If system deadstart is selected, also specify (in word 14) the level of deadstart, CPU options, and system devices (if CMRDECK and IPRDECK are not to be modified).

   a. Set MODE switch to LOAD

   b. Set toggle switches of deadstart panel

   Part II, section 2 describes deadstart panel selections.

Prepare Card Reader

4. Insert coldstart deck in card reader (words 1, 2, 3, 6, 10, and 11 of the deadstart panel specify the controller and channel number used to access the card reader).

   Refer to appendix D of the operator's guide for instructions for operating the card reader.

Activate Deadstart Switch

5. Momentarily activate either the DEADSTART switch on the deadstart panel or the switch on the system console.

Activate Card Reader

6. Read the coldstart deck into the card reader.

   Coldstart now proceeds as described in step 5, section 1.1. (References to deadstart after and including step 5 also apply to coldstart.)

   Refer to appendix D of the operator's guide for instructions for operating the card reader.

1.3 WARMSTART PROCEDURE

When the deadstart tape is mounted on a 667 or 669 magnetic tape unit, a warmstart may be performed if the controlware has already been loaded into the controller. Although the deadstart panel setting is different for warmstart (refer to part II, section 2 for panel settings), the warmstart procedure is the same as the procedure for deadstart, section 1.1. Thus, this procedure is used when it is necessary to perform a warmstart.

If the controlware has already been loaded into the controller, either a coldstart or a warmstart may be performed. However, warmstart procedures are sufficient in this case. If the controlware has not been loaded, a coldstart must be performed.

NOTE

If either channel access to a dual access controller for a 667 or 669 is reserved when warmstart is required, it may be necessary to clear that channel in order to perform a warmstart from a 667 or 669.

For example, if a channel accessing the tape unit to be used for deadstart (specified on the deadstart panel) is reserved due to system malfunction requiring warmstart, the channel must be cleared manually before warmstarting. To do this, perform the following operations on the affected side of the controller (that is, if the tape unit can be accessed from channels 13 and 33 and channel 13 is reserved, perform the operations on the channel 13 side of the controller).

Press the STOP switch.

Press the MASTER CLEAR switch.

Press the GO switch.

1.4 POTENTIAL DEADSTART PROBLEMS

If errors are encountered during deadstart, a descriptive message is displayed on the right console screen and deadstart halts. Refer to the error processing information in section 2 of the operator's guide.

The following are other problems that may occur during deadstart.

Problem	Cause
Tape moves but stops before any display is activated.	• There is a parity error on one of the first records of the deadstart tape. • The deadstart tape is not an I-mode unlabeled tape. • A 7-track tape is mounted on a 9-track drive, or vice-versa. • The tape unit is set to the wrong density. • Word 12 on the deadstart panel is set incorrectly. • A channel parity error on a CDC CYBER 170 has been detected by the tape controller. • During a coldstart, the card reader and/or the tape unit containing the deadstart tape are on a channel with a PPU.
Tape does not move.	• During a coldstart, the card reader and the 667/669 tape unit are on the same channel. • If the unit light of the deadstart tape unit is lighted, the correct unit is selected but the deadstart panel is set incorrectly. Often it is the load address in word 12 that is not set correctly. • If the unit light does not light, check the channel, controller, and unit selections on the deadstart panel. • Two or more units may have the same physical unit number. • Unit is not ready. • A channel parity error on a CDC CYBER 170 has been detected by the tape controller. • During a coldstart, the card reader and/or the tape unit containing the deadstart tape are on a channel with a PPU.

DEADSTART PANEL SETTINGS 2

2.1 GENERAL DESCRIPTION

Words 0001 through 0014 of the deadstart program, program for coldstart, or program for warmstart are transferred from the deadstart panel to locations 1 through 14_8 in PPU 0 memory to execute when the DEADSTART switch is activated. The CDC CYBER 170 deadstart panel also contains words 0015 through 0020. Loading of CDC CYBER 170 deadstart panel instructions to PPU 0 ends with the transmission of word 0020. Words 0016 through 0020 are reserved for future system use and their settings are ignored. Deadstart programs for coldstart and warmstart are used only when the deadstart tape is mounted on a 667 or 669 magnetic tape unit.

Each of the deadstart programs:

- Specifies and activates the equipment to read the deadstart tape (words 1 through 12)
- Selects the LIBDECK to be used in building the system (word 13, www)
- Selects the CMRDECK to be used to deadstart (word 13, xxx xxx)
- Specifies whether the deadstart is to be a system or a maintenance deadstart (word 13, yyy)
- Specifies the following system deadstart variables:
    - Devices on which system library is to reside (word 14, sss sss)
    - Central processor options (word 14, ppp)
    - Level of system deadstart (word 14, rrr)

Figures II-2-1, II-2-2, and II-2-3 illustrate the deadstart panel switch position for the NOS deadstart, coldstart, and warmstart programs.

The switch positions indicated by a 1 (switch in up position) or a 0 (switch in down position) are mandatory settings. However, the switch positions for fields represented by alphabetic characters are determined by each installation. Each of these fields is described in the information in the following sections. (Words 0016 through 0020 of the CDC CYBER 170 deadstart panel are reserved for future system use and are not shown in Figures II-2-1, II-2-2, and II-2-3. Word 0015 must be set to 0 for NOS).

WORD	SWITCH (BIT) POSITIONS											
	11	10	9	8	7	6	5	4	3	2	1	0
0001	1	1	1	1	0	1	t	t	t	t	t	t
0002	1	1	1	1	1	1	t	t	t	t	t	t
0003	e	e	e	0	0	0	0	0	u	u	u	u
0004	1	1	1	1	1	1	t	t	t	t	t	t
0005	0	0	0	0	0	0	0	0	1	0	0	0
0006	1	1	1	1	1	1	t	t	t	t	t	t
0007	0	0	1	1	0	0	0	0	0	0	0	0
0010	1	1	1	1	0	0	t	t	t	t	t	t
0011	1	1	1	0	0	1	t	t	t	t	t	t
0012	1	1	0	1	0	0	0	0	0	0	0	0
0013	w	w	w	x	x	x	x	x	x	y	y	y
0014	r	r	r	p	p	p	s	s	s	s	s	s
†0015	0	0	0	0	0	0	0	0	0	0	0	0

Figure II-2-1. NOS Deadstart Program

† Words 0015 through 0020 apply only to CDC CYBER 170 machines.

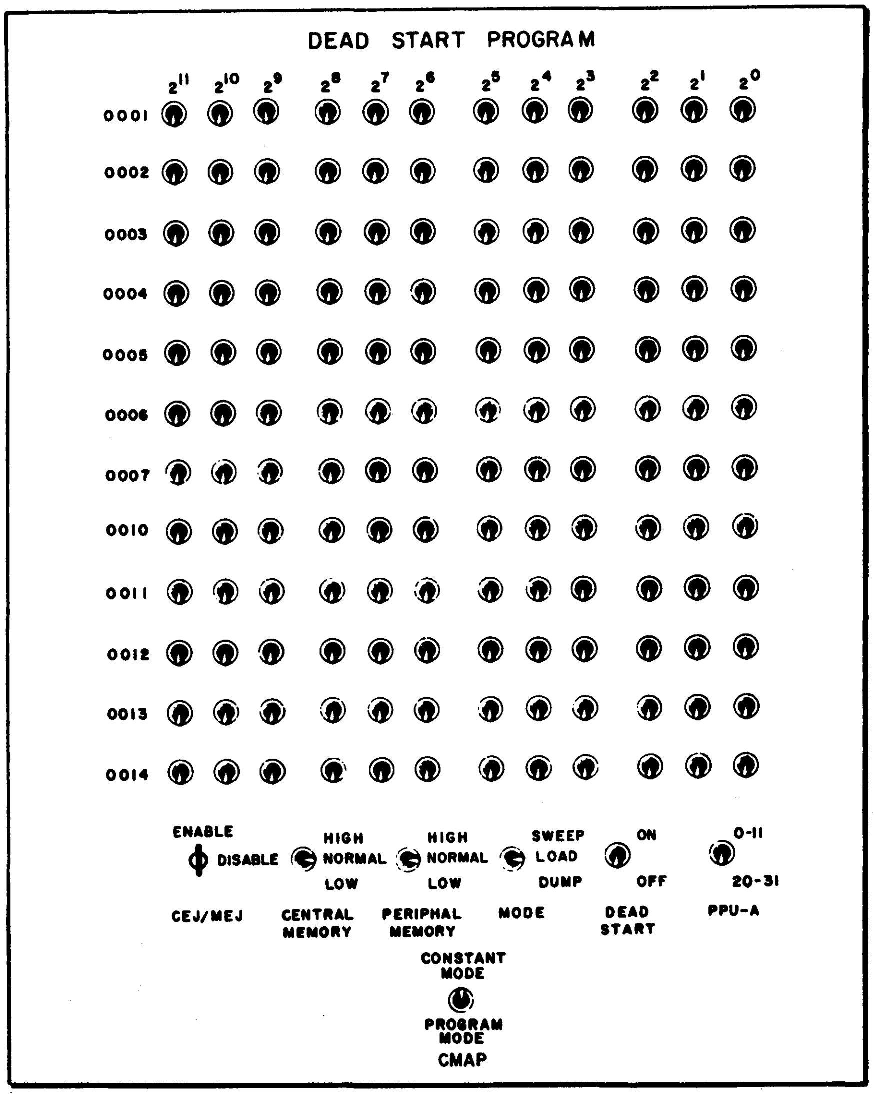


Figure II-2-4. CDC CYBER 70/6000 Deadstart Panel

In reference to Figure II-2-4, the following switches are also present on the CDC CYBER 70/6000 deadstart panel.

CEJ/MEJ	This key is used to enable or disable the central exchange jump/monitor exchange feature. To enable CEJ/MEJ, turn the key fully counterclockwise. To disable CEJ/MEJ, turn the key fully clockwise. If it is necessary to disable CEJ/MEJ for maintenance purposes, it is recommended that this key be used rather than the option available during deadstart.
CENTRAL MEMORY, PERIPHAL MEMORY, and MODE	These switches are used to perform hardware maintenance functions. For normal system operation the CENTRAL MEMORY and PERIPHAL MEMORY switches should be placed in the NORMAL position. The MODE switch should be placed in the LOAD position. For further information, refer to the appropriate hardware reference manual.
DEADSTART	The DEADSTART switch should be activated by briefly holding it in the ON position.
PPU-A	This switch determines whether the internal PPU chassis is numbered 0 through 11 or 20 through 31.
CMAP	The CMAP (central memory access priority) switch is available as a standard option. When this switch is in the CONSTANT MODE position, all PPUs may interrupt ECS transfers. When this switch is in the PROGRAM MODE position, only priority PPUs are able to interrupt ECS transfers. Refer to the appropriate hardware reference manual for information on assigning priority status to a PPU. It is recommended that NOS be run with the CMAP switch in the CONSTANT MODE position.

For all other switches, move the switch up to set the 1 or ON position; move the switch down to set the 0 or OFF position.

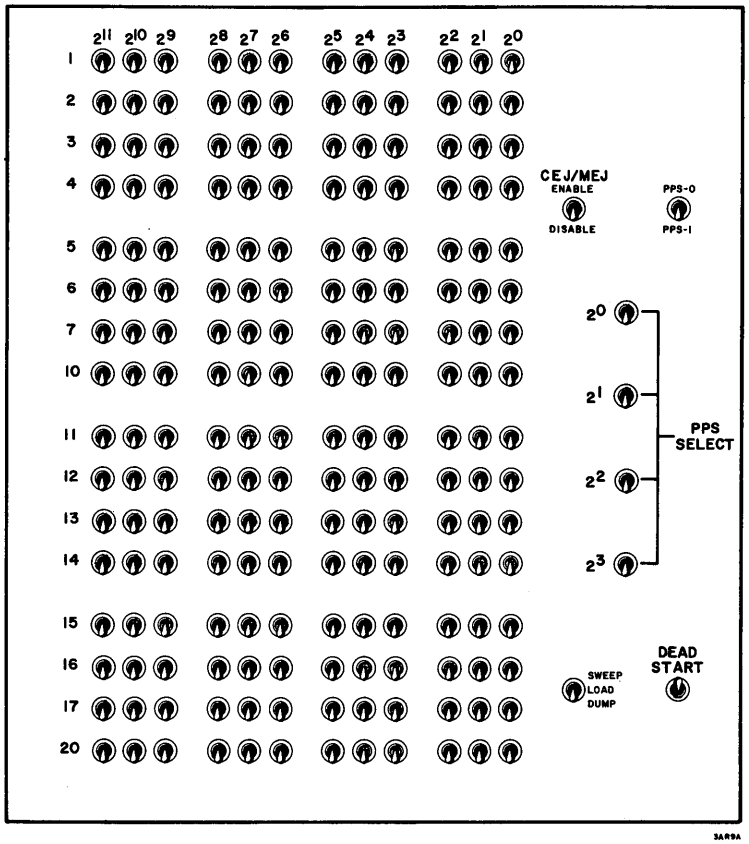


Figure II-2-5. CDC CYBER 170 Deadstart Panel

In reference to Figure II-2-5, the following switches are also present on the CDC CYBER 170 deadstart panel.

CEJ/MEJ	This switch is used to enable or disable the central exchange jump/monitor exchange jump feature. If it is necessary to disable this feature for maintenance purposes, it is recommended that this switch be used, rather than the option available during deadstart.
PPS-0, PPS-1	This switch determines wheter the internal PPU chassis is numbered 0 through 11 (PPS-0 position) or 20 through 31 (PPS-1 position).
PPS SELECT	These switches indicate which peripheral processor is to contain PPU 0. This selection is made by enabling the switches to form the binary number of the desired peripheral processor.
SWEEP, LOAD, DUMP	This switch is used when performing hardware maintenance functions. For normal system operation, it should be placed in the LOAD position. For further information, refer to the appropriate hardware reference manual.
DEADSTART	The DEADSTART switch should be activated by briefly holding it in the ON position.

For all other switches, move the switch up to set the 1 or ON position; move the switch down to set the 0 or OFF position.

2.2 WORDS 0001 THROUGH 0012

Set the switches for words 0001 through 0012 to identify the magnetic tape unit on which the deadstart tape is mounted, its associated controller, and the channel used to access this equipment. When setting switches for coldstart, also indicate the controller associated with the card reader and channel used to access this equipment.

0001	111 101 ttt ttt
0002	111 111 ttt ttt
0003	eee 000 00u uuu
0004	111 111 ttt ttt
0005	000 000 001 000
0006	111 111 ttt ttt
0007	001 100 000 000
0010	111 100 ttt ttt
0011	111 001 ttt ttt
0012	110 100 000 100

Deadstart Program

0001	111 101 ccc ccc
0002	111 111 ccc ccc
0003	eee 000 000 000
0004	010 100 000 000
0005	010 100 ttt ttt
0006	111 111 ccc ccc
0007	001 100 uuu 000
0010	111 100 ccc ccc
0011	111 001 ccc ccc
0012	111 110 110 100

Deadstart Program for Coldstart

0001	111 101 ttt ttt
0002	011 110 001 101
0003	001 000 001 100
0004	010 100 000 000†
0005	010 100 000 000†
0006	111 111 ttt ttt
0007	000 010 110 uuu
0010	111 100 ttt ttt
0011	111 001 ttt ttt
0012	110 100 000 000

Deadstart Program for Warmstart

ccc ccc — Channel used to access the card reader

ttt ttt — Channel used to access the magnetic tape equipment

eee — Controller associated with magnetic tape unit uuuu
For coldstart, controller associated with the card reader

uuuu — Magnetic tape unit on which the deadstart tape is mounted

---

†

0004	111 111 ttt ttt
0005	010 001 000 000

if a 6681 or a 6684 is on the channel

When the deadstart panel is set and the deadstart switch is activated, the hardware automatically:

- Master clears all data channels and sets them to the active and empty condition
- Assigns all PPUs to their corresponding data channels (that is, PPU 1 to channel 1, etc.)
- Sets each PPU to the IAM instruction, so that each PPU is waiting for data to become available on its associated channel
- Loads contents of deadstart panel into PPU 0 memory, disconnects the deadstart panel, disconnects channel 0, and initiates execution of the program loaded from the deadstart panel. This program (deadstart, coldstart, or warmstart) reads and executes the first record on the deadstart tape (PRL). PRL displays and processes any maintenance options that are selected and loads the proper records from tape that are selected. (For example, if PP0 or DDD options are selected, PRL loads these records.) In addition, during coldstart, the deadstart program for coldstart reads the coldstart deck from the card reader before PRL is read and executed. The program on the coldstart deck (ABC) is executed, causing the magnetic tape controlware to be loaded and then PRL is read and executed. (Refer to part III, section 2.7.2 for a description of the coldstart deck.)

2.3 WORD 0013

Word 13 specifies the deadstart function, the CMRDECK to be used at deadstart, and the LIBDECK to be used in generating the system.

0013	www xxx xxx yyy

yyy — These switches specify the deadstart function that occurs when the deadstart switch is activated. These functions are system deadstart or maintenance deadstart.

The maintenance deadstart function allows the system to be used to perform hardware or software maintenance tasks. Only the maintenance operation selected can be performed; processing of user jobs is not possible.

yyy	Deadstart Function
000	Automatic system deadstart PRL automatically loads the system without displaying deadstart options, and bypasses PP0, DDD, DDS, and EDD. The CCM (check computer memory) portion of PRL is called, presetting all PPU memory and central memory (PPU memory only on a level 3 deadstart). It then checks the data written and the S/C (status/control) register (available only on a CDC CYBER 170). If any error occurs in the data or if an error status bit is found to be set in an S/C register of a CDC CYBER 170, the error is displayed and operator intervention is required to continue the deadstart sequence. (Refer to Display S/C Register, section 3.8 and section 2 of the operator's guide.)

yyy	Deadstart Function
	If no errors are found during this memory confidence checking, PRL automatically activates SET (process CMRDECK entries). The first display to appear at the system display after the deadstart switch is activated and memory confidence checking is complete is determined by the setting of word 14, bits 0 through 5. Generally, this automatic process consists of the following. 1. Validates labels on all mass storage devices. This is done to ensure that the configuration matches that specified in the CMRDECK being used. 2. Builds central memory tables that reflect information contained in the device labels (level zero deadstart only). If a recovery deadstart is being performed, the central memory tables can be recovered from checkpoint information on mass storage (levels 1 and 2) or verified against information in device labels if central memory is found to be intact (level 3). 3. Loads base operating system (core system) programs into central memory. The amount of loading to be performed depends upon the level of deadstart selected in word 14.
001	System deadstart with deadstart options displayed The deadstart options are displayed on the left console screen when the DEADSTART switch is activated. Deadstart proceeds according to the options selected. The display and entries that can be made at the system console are described in part II, section 3. The remaining field in word 13 and all fields of word 14 are optional when yyy=001 because the options display allows the operator to specify values for these fields from the console keyboard. However, it is recommended that these fields be set on the deadstart panel in order to establish default values. In this case, the options display can be used when it is required to alter the default values established on the deadstart panel. The display that occurs after the deadstart options display depends upon the current setting of word 14, sss sss. (The deadstart panel setting of word 14 can be overridden with the W deadstart option described in part II, section 3.11.)
010	Maintenance deadstart that displays the contents of PPU 0 PRL loads PP0 (the memory display program) into PPU memory. PP0 displays the contents of PPU 0 on both system console display screens. This function is generally used by engineers to enter sample PPU programs (for example, to test peripheral devices on the system).

yyy	Deadstart Function
	To perform another deadstart function, it is necessary to reset this field (yyy) on the deadstart panel and activate the DEADSTART switch.
	PPU 0 memory can also be displayed when the deadstart options are displayed (part II, section 3.7).
011	Maintenance deadstart to load DDD
	PRL loads DDD, the deadstart dump program, into PPU memory when the DEADSTART switch is activated. It allows the contents of PPU memory and/or central memory to be dumped to a specified line printer.
	Refer to part II, section 3.3 for dump procedures. To continue after dump processing, reset this field (yyy) on the deadstart panel choosing another deadstart function and activate the DEADSTART switch.
	Other deadstarts or dumps may be selected during a system deadstart with options displayed (yyy=001). These are DDS, SCR (CDC CYBER 170 only), and EDD.

xxx xxx — These switches specify which CMRDECK on the deadstart tape is to be used to deadstart. CMRDECK is described in detail in part II, section 4.

There can be up to 64_{10} CMRDECKs on the deadstart tape. If it is necessary to perform a level 1, 2, or 3 (recovery) deadstart, the CMRDECK selected during the most recent level 0 deadstart must be used. Levels of deadstart are discussed under word 14.

It is not necessary to specify the CMRDECK on the deadstart panel if system deadstart with options displayed is selected in word 13 (yyy=001). In this case, the options displayed allow the operator to specify the CMRDECK to be used from the console keyboard. Additionally, values entered via the options display have precedence over those specified on the deadstart panel. For example, bits 3 through 8 of word 13 on the deadstart panel (xxx xxx) could be set to select the CMRDECK most frequently used by an installation. Another CMRDECK could then be selected when necessary via the options display during a level 0 deadstart.

www — These switches specify which LIBDECK on the deadstart tape is to be used to regulate the system load. LIBDECK is discussed in detail in part III, section 2.6.

There can be up to eight LIBDECKs on the deadstart tape. The records are named LIBDECK and LIBDCK1 through LIBDCK7. They are selected by setting www = 000 through 111, respectively.

2.4 WORD 0014

Word 14 is significant only for a system deadstart (word 13, yyy). It specifies the system devices, the central processor options, and the level of system deadstart.

0014	rrr ppp sss sss

sss sss — These switches specify whether the CMRDECK and IPRDECK are to be displayed before loading the system library or whether deadstart is to load the system library without further displays.

It is not necessary to enter word 14 on the deadstart panel if system deadstart with options displayed is selected in word 13 (yyy=001). In this case, the options provided allow the operator to enter all values for word 14 from the console keyboard. The values entered through the options display have precedence over those specified on the deadstart panel.

sss sss	Action
set	These switches (bits 0 through 5) correspond to the mass storage devices defined in the first six entries (ordinals 0 through 5) of the equipment status table (EST). Each switch that is set to the ON position specifies that the system library will reside on the corresponding mass storage device. If more than one device is specified, system efficiency can be greatly improved because more than one system file can then be accessed at the same time. All devices specified must be of the same equipment type. For example, if sss sss is set to 000 101, when the DEADSTART switch is activated the system library from the deadstart tape is placed on mass storage equipment with EST ordinals 0 and 2. The equipment assigned to EST ordinals 0 and 2 must be defined in the CMRDECK as the same type of mass storage equipment. The next display when sss sss is set is the request for date and time. There is no opportunity to alter the CMRDECK or IPRDECK.
000 000	Deadstart displays CMRINST when this setting is selected for sss sss; the CMRDECK selected in word 13 or the IPRDECK specified in the CMRDECK can be altered. Therefore, when it is necessary to make modifications to the CMRDECK during deadstart, no system device can be specified on the deadstart panel. In this case, one of the CMRDECK entries is the SYSTEM entry which allows specifying any of the mass storage devices assigned to the first 17_8 EST entries as system devices (whereas only mass storage devices assigned to EST ordinals 0 through 5 can be specified via the deadstart panel switches). Refer to section 4 for CMRDECK modifications and to section 5 for IPRDECK modifications.

sss sss	Action

In summary, there are three opportunities to select the system device(s).

1. Place one or more of the switches identified by sss sss in the ON position.
2. Select option to enter word 14 via the options display.
3. Specify the SYSTEM entry when making CMRDECK modifications.

If a system device is never specified, the system library will reside on the mass storage device assigned to the first entry of the EST by default.

ppp These switches designate the central processor control settings.

ppp	Central Processor Control Option
bit 6	CPU 0 option

0 Indicates that CPU 0 is available in the system. This switch is significant only for dual CPU systems and is normally left in this position.

1 Indicates that CPU 0 is not available in the system. On dual CPU systems, this allows use of the system when CPU 0 is down. CPU 0 and CPU 1 should not be disabled simultaneously (bits 6 and 7 both in up position).

NOTE

If both CPU 0 and CPU 1 are set to 0 (available), the system determines if both CPUs are available.

This option also indicates that the compare move unit (CMU) is unavailable, if present. In a dual CPU system, if either CPU is disabled, detection of CMU is also disabled.

bit 7 CPU 1 option

0 Indicates that CPU 1 is available in the system. This switch is significant only for dual CPU systems and is normally left in this position.

1 Indicates that CPU 1 is not available in the system. On dual systems, this allows use of the system when CPU 1 is down. CPU 0 and CPU 1 should not be disabled simultaneously.

NOTE

If both CPU 0 and CPU 1 are set to 0 (available), the system determines if both CPUs are available.

This option also indicates that the CMU is unavailable, if present. In a dual CPU system, if either CPU is disabled, detection of CMU is also disabled.

ppp	Central Processor Control Option
bit 8	CEJ/MEJ option

0 Indicates that CEJ/MEJ option is enabled

1 Disables CEJ/MEJ option. This is not recommended because the system automatically determines if the CEJ/MEJ option is present and initializes accordingly. Therefore, this switch should always be set to the down position. System failure may result if the CEJ/MEJ option is present in the system and is disabled in this manner.

If it is necessary to disable the CEJ/MEJ option for maintenance purposes, use the keylock labeled CEJ/MEJ on the deadstart panel of a CDC CYBER 70 or the switch labeled CEJ/MEJ on the deadstart panel of a CDC CYBER 170. Refer to Figures II-2-4 and II-2-5 in this section.

rrr These switches designate the level of system deadstart. For additional information concerning levels of deadstart, refer to section 2 of the operator's guide.

rrr	Level of System Deadstart
000	Indicates an initial or level 0 system deadstart in which the system is loaded from the deadstart tape. This is not considered to be a recovery deadstart although permanent files are recovered automatically. Permanent files are recovered on all levels of system deadstart. Level 0 is specified for the first deadstart following a period in which the system has been inoperative or has been used for other than NOS operations. If it becomes necessary to redeadstart the system (because of system malfunction, for example), it is recommended that a level 3 recovery deadstart be attempted. If level 0 is selected, all jobs in the system (excluding system origin type and time-sharing origin type) are reinitiated, I/O queue files are recovered (if the IPRDECK entry QPROTECT is enabled), and the system is reloaded from the deadstart tape.

rrr	Level of System Deadstart
001	Indicates a level 1 recovery deadstart whereby the system, all jobs, and all active files are recovered from checkpoint information on mass storage. All permanent files and I/O queue files are also recovered. Level 1 recovery is effective only if the DSD command CHECK POINT SYSTEM (refer to operator's guide) has been successfully completed immediately prior to deadstarting. Additionally, if a significant amount of system activity has occurred since the last checkpoint was performed, the checkpoint information may no longer be reliable. In this case, it is recommended that a level 0 (initial) deadstart be performed. In a multimainframe environment, it is assumed that ECS is intact for a successful level 1 recovery. Unless the operator is certain that the content of central memory was destroyed by the system malfunction, he should always attempt to perform a level 3 recovery deadstart (rrr=011).
010	Indicates a level 2 recovery deadstart whereby all jobs and active files are recovered from checkpoint information. Permanent files are also recovered. However, the system is loaded from a deadstart tape. Level 2 recovery deadstart is normally used in system test situations. In a multimainframe environment, it is assumed that ECS is intact for a successful level 2 recovery.
011	Indicates a level 3 recovery deadstart whereby all jobs, active files, and the system, with the exception of the library directory, are recovered from central memory tables. The library directory is recovered from mass storage. Permanent files are also recovered. In a multimainframe system, MMF tables are recovered from ECS. A CHECK POINT SYSTEM command must have been issued to prevent loss of SYSEDIT information. A level 3 recovery deadstart is normally performed following an equipment malfunction (for example, channel or PPU hung) if the system is intact. It is recommended that a level 3 recovery deadstart be attempted before resorting to a level 0 deadstart. A level 1 or 2 deadstart should never be attempted after a level 3 deadstart fails.

NOTE

Since memory confidence testing destroys PPU memory (central memory is also destroyed in level 0, 1, and 2 deadstarts), complete the dumping of PPU memory (as well as, central memory for levels 0, 1, and 2) before deadstart occurs.

3.1 GENERAL DESCRIPTION

If a system deadstart with deadstart options is specified on the deadstart panel (word 13, yyy), the following deadstart options display appears on the console screen when the DEADSTART switch is activated. The various options that can be selected when this display appears are described in the following sections.

CDC CYBER 170 Display

```
DEAD START OPTIONS.

CR      LOAD SYSTEM.
 I      IGNORE ERROR PROCESSING.
 K      SET *CMRDECK* NUMBER.
 T      DEAD START ANOTHER TAPE.
 W      ENTER WORD 14.
 X      DISABLE CEJ/MEJ.
 0      TURN OFF CPU 0 / CMU.
 1      TURN OFF CPU 1 / CMU.
 D      DEAD START DUMP.
 F      FULL DUMP TO TAPE.
 P      PP 0 DISPLAY.
 S      DIAGNOSTIC SEQUENCER.
 R      DISPLAY S/C REGISTER.
```

Description
section 3.2
section 3.5
section 3.6
section 3.10
section 3.11
section 3.12
section 3.13
section 3.14
section 3.3
section 3.4
section 3.7
section 3.9
section 3.8

CDC CYBER 70/6000 Display

```
DEAD START OPTIONS.

CR      LOAD SYSTEM.
 I      IGNORE ERROR PROCESSING.
 K      SET *CMRDECK* NUMBER.
 T      DEAD START ANOTHER TAPE.
 W      ENTER WORD 14.
 X      DISABLE CEJ/MEJ.
 0      TURN OFF CPU 0 / CMU.
 1      TURN OFF CPU 1 / CMU.
 D      DEAD START DUMP.
 F      FULL DUMP TO TAPE.
 P      PP 0 DISPLAY.
 S      DIAGNOSTIC SEQUENCER.
```

Description
section 3.2
section 3.5
section 3.6
section 3.10
section 3.11
section 3.12
section 3.13
section 3.14
section 3.3
section 3.4
section 3.7
section 3.9

NOTE

> The CDC CYBER 70/6000 console contains two screens, referred to as the left and right screens. Displays which appear on either the left or right screen of a CDC CYBER 70 appear on the single screen of a CDC CYBER 170 by placing the PRESENTATION switch in the LEFT or RIGHT position. When the PRESENTATION switch is placed in the MAINTENANCE (central) position, both the left and right displays appear on the single screen on the CDC CYBER 170. In this manual, no distinction between console types is made when referring to displays. Displays are noted only as left or right displays.

3.2 LOAD SYSTEM OPTION

This option provides two possibilities.

To continue deadstart using the system configuration defined in the current CMRDECK.	Refer to section 3.2.1.
To modify the current CMRDECK (and also IPRDECK, if desired) before continuing deadstart.	Refer to section 3.2.2.

Additional options cannot be selected after this entry. However, if an error status bit is found to be set in the S/C register of a CDC CYBER 170, processing continues as if the R option had been entered.

3.2.1 LOAD SYSTEM WITHOUT MODIFICATIONS

Use this procedure to continue deadstart by loading the system library on the system devices.

1. Word 13, xxx xxx must be set to the number of the CMRDECK that defines the system configuration. If the appropriate number is not indicated on the deadstart panel, use the K option described in section 3.6 to specify the correct number.

2. Word 14, sss sss indicates the devices onto which the system library is to be loaded. If sss sss is not set on the deadstart panel, use the W option described in section 3.11 to specify the system devices.

3. At the system console, press CR .

   The system library is loaded onto the system devices after memory confidence checking. The next display after memory confidence checking requests the date. For subsequent procedure, refer to section 1.1, step 11.

3.2.2 MODIFY CMRDECK AND/OR IPRDECK BEFORE LOADING SYSTEM

1. Word 13, xxx xxx must be set to the number of the CMRDECK that defines the system configuration. If the appropriate number is not indicated on the deadstart panel, use the K option described in section 3.6 to specify the correct number.

2. Word 14, sss sss must be set to 000 000. If it is not set to 000 000 on the deadstart panel, use the W option described in section 3.11 to do so.

3. At the system console, press CR.

   The next display after memory confidence checking is the CMRINST display. For subsequent procedure, refer to section 1, step 6.

3.3 DUMP TO PRINTER OPTION

DDD (deadstart dump program) is a maintenance program that can be used to dump a PPU memory or central memory. The following are possible procedures.

Normal PPU memory dump	Refer to section 3.3.1.
Dump PPU 0 saving all memory destroyed by DDD	Refer to section 3.3.2.
Dump central memory	Refer to section 3.3.3.

3.3.1 NORMAL PPU MEMORY DUMP

1. Load DDD in either of the following ways.

   a. If deadstart options are displayed, type:

      D

   b. If a dump is requested at the deadstart panel, word 13, bits 0 through 2 must be set to 011.

At the system console:

2. DDD requests channel number of printer to which PPU memory is to be dumped:

   LP CH 00

3. Type ch
   Press CR

   ch Two-digit printer channel number; enter value from right to left.

4. DDD requests the same printer's controller number:

   LP NO 00

5. Type no
   Press CR

   no Two-digit controller number; enter value from right to left

If the error message NOT RDY. appears on the display, the printer specified is not ready to print. Respond with either of the following.

a. Press DEADSTART switch on system console; deadstart again beginning with step 1 designating channel and controller numbers for printer that is ready.

b. Ready the printer already designated.

6. DDD requests whether a PPU or central memory is to be dumped:

   P OR C

7. Type P

8. DDD requests the number of the PPU whose memory is to be dumped:

   PP NO 00

9. Type no
   Press CR

   no   Two-digit PPU number; 0 through 11_8; enter value from right to left

   DDD dumps the selected PPU memory to the printer. The format is a matrix of 100_8 words per block with 12 blocks per page.

   During the dump process, the system console screens are blank.

10. After the dump is completed, DDD displays the message P OR C for further dumping. If central memory is to be dumped, continue with step 6 of the procedure in section 3.3.3; if another PPU's memory is to be dumped, continue with step 6 of section 3.3.1. Otherwise, activate the DEADSTART switch.

3.3.2 SAVE ALL PPU 0 MEMORY DURING DUMP

When DDD dumps a PPU memory, it destroys the following memory locations.

Locations 0 through 21_8 of PPU 0 (CDC CYBER 170)	Deadstart program
Locations 0 through 15_8 of PPU 0 (CDC CYBER 70/6000)	Deadstart program
Locations 6600 through 7777 of PPU 0	Deadstart dump program
Locations 0 through 3 of all other PPUs	PPU idler routine

Locations 6600 through 777 of PPU 0 can be saved by transferring PPU 0 memory to another PPU first and then dumping that PPU to a printer. Use the following procedure to do so.

At deadstart panel:

1. Before deadstarting, enter the following program.

Word	Setting
0001	2000
0002	7776
0003	73pp
0004	0000
0005	0300

   pp   Number of PPU to which PPU 0 memory is to be transferred.

   Words 6 through 14 are not used at this time.

2. Activate DEADSTART switch.

3. Reset the deadstart panel to load DDD as indicated in section 3.3.1, step 1. Continue with that procedure.

4. For 20 PPU systems, an alternate procedure is possible to transfer PPU 0 memory.

   a. Toggle the PPU-A switch (for CDC CYBER 70/6000 machines) or the PPS-0, PPS-1 switch (for CDC CYBER 170 machines) on the deadstart panel before deadstarting. This switch determines whether the internal PPU chassis is numbered 0 through 11 or 20 through 31. When this switch is toggled, PPU 0 becomes PPU 20. PPU 20 can now be dumped to a printer.

   b. Activate the DEADSTART switch.

   c. Continue with the procedure to load DDD as indicated in section 3.3.1, step 1.

5. In addition, on CDC CYBER 170 machines, the following procedure may be used.

   a. Reset the PPS select switches on the deadstart panel before deadstarting. These switches indicate which peripheral processor is to contain PPU 0. This selection is made by enabling the switches to form the binary number of the desired peripheral processor. The peripheral processor which previously contained PPU 0 can now be dumped to a printer.

      NOTE

      The PPS select switches are normally set to 0. When dumping PPU 0, reset these switches to 1. This causes PPU 0 to become PPU 11. PPU 11 can now be dumped. (For 7 PPU systems, PPU 0 becomes PPU 6.) If this process is to be repeated, the PPS select switches can be set to 0. PPU 0 becomes PPU 1, and PPU 1 can be dumped. For additional information, consult the appropriate system hardware reference manual listed in part I, section 4.

   b. Activate the DEADSTART switch.

   c. Continue with the procedure to load DDD as indicated in section 3.3.1, step 1.

3.3.3 DUMP CENTRAL MEMORY

1. Load DDD in either of the following ways.

   a. If deadstart options are currently displayed, type:

      D

   b. If a dump is requested at the deadstart panel, word 13, bits 0 through 2 are set to 011.

At system console:

2. DDD requests channel number of printer to which PPU memory is to be dumped:

   LP CH 00

3. Type ch
Press CR

ch Two-digit printer channel number; enter value from right to left

4. DDD requests the same printer's controller number:

LP NO 00

5. Type no
Press CR

no Two-digit controller number; enter value from right to left

If the error message NOT RDY. appears on the display, the printer specified is not ready to print. Respond with either of the following actions.

a. Press DEADSTART switch on system console; deadstart again beginning with step 1 designating channel and controller numbers for printer that is ready.

b. Ready the printer already designated.

6. DDD requests whether a PPU or central memory is to be dumped:

P OR C

7. Type C

8. DDD requests the starting address of the portion of central memory to be dumped:

CM FROM 000000

9. Type address
Press CR

address Six-digit starting address of memory to be dumped; enter value from right to left

10. DDD requests the end address of the portion of central memory to be dumped:

CM TO 000000

11. Type address
Press CR

address Six-digit ending address of memory to be dumped; enter values from right to left

12. DDD dumps the selected portion of central memory to the printer specified. The format is four central memory words per line. Any 4-word line that is a duplicate of the preceding line is indicated by a line with no address on the left and the message DUPLICATED LINES.

During the dump process, the system console screens are blank.

DDD does not destroy any central memory locations.

After the dump is completed, DDD displays the message P OR C for further dumping. If a PPU memory is to be dumped, continue with step 6 of the normal PPU memory dump procedure. If no further dumps are desired, activate the DEADSTART switch.

3.4 DUMP TO TAPE OPTION

Express Deadstart Dump (EDD) is a maintenance program that can be used to dump all PPUs' memories, central memory, the deadstart contents of the CPU hardware registers, and the S/C registers to magnetic tape during deadstart. ECS memory can optionally be dumped. Later, all or part of this dump can be printed using the utility routine DSDI as described in part IV, section 4. This process is useful because it requires less time at deadstart than using the D option (section 3.3) to dump directly to a printer.

1. When EDD dumps a PPU memory, it destroys the following PPU memory locations.

Locations 0 through 21_8 of PPU 0 (CDC CYBER 170)	Deadstart program
Locations 0 through 15_8 of PPU 0 (CDC CYBER 70/6000)	Deadstart program
Locations 5400 through 7777 of PPU 0	EDD program
Locations 0 through 7 of PPU 1	PPU 1 idler routines
Locations 0 through 3 of all other PPUs	PPU idler routines

PPU 0 can be saved by transferring PPU 0 memory to another PPU before performing the express dump. Use the following procedure to do so.

a. Before deadstarting, enter the following program on the deadstart panel.

Word	Setting
0001	2000
0002	7776
0003	73pp
0004	0000
0005	0300

pp Number of PPU to which PPU 0 memory is to be transferred.

Words 6 through 14 are not used at this time.

b. Activate DEADSTART switch.

c. Reset the deadstart panel for system deadstart (part II, section 2) and activate the DEADSTART switch.

d. For 20 PPU systems, an alternate procedure is possible to transfer PPU 0 memory.

   1) Toggle the PPU-A switch (for CDC CYBER 70/6000 machines or the PPS-0, PPS-1 switch (for CDC CYBER 170 machines) on the deadstart panel before deadstarting. This switch determines whether the internal PPU chassis is numbered 0 through 11 or 20 through 31. When this switch is toggled, PPU 0 becomes PPU 20. PPU 20 can now be dumped.

   2) Activate the DEADSTART switch.

   3) Continue with step 2.

e. In addition, on CDC CYBER 170 machines, the following procedure may be used.

   1) Reset the PPS select switches on the deadstart panel before deadstarting. These switches indicate which peripheral processor is to contain PPU 0. This selection is made by enabling the switches to form the binary number of the desired peripheral processor. The peripheral processor which previously contained PPU 0 can now be dumped to a printer.

      NOTE

      The PPS select switches are normally set to 0. When dumping PPU 0, reset these switches to 1. This causes PPU 0 to become PPU 11. PPU 11 can now be dumped. (For 7 PPU systems, PPU 0 becomes PPU 6.) If this process is to be repeated, the PPS select switches can be set to 0. PPU 0 becomes PPU 1, and PPU 1 can be dumped. For additional information, consult the appropriate system hardware reference manual listed in part I, section 4.

   2) Activate the DEADSTART switch.

   3) Continue with step 2.

2. To load EDD:

   Type F

3. EDD requests channel number of tape unit to which memory is to be dumped.

   MT CH 00

4. To enter the tape channel number:

   Type ch
   Press CR

   ch Two-digit channel number; enter values from right to left

5. EDD requests the tape unit's controller and unit numbers:

   MT E0UU 0000

6. To enter the tape controller and unit numbers:

   Type e0uu
   Press CR

   e — One-digit number of tape unit's controller (0 if dumping to a 667 or 669 magnetic tape unit)

   uu — Two-digit number of tape unit

   Enter e0uu from right to left.

7. EDD requests the dump identifier; this identifier will be placed on the first record of the tape for future reference.

   EXPRESS 00 hh. mm. ss. yy/mm/dd

   The time and date are from central memory; they may be incorrect if a system malfunction occurred before using the dump procedure.

8. To enter the dump identifier:

   Type id
   Press CR

   id — Two-digit dump identifier; enter from right to left

9. EDD requests the tape rewind option:

   NONZERO INHIBITS REWIND 00

10. To enter the tape rewind option:

    Type x
    Press CR

    x — Type a nonzero octal entry to prevent tape rewind before and after dump; enter 0 (or press CR) to cause tape rewind before and after dump. Multiple dumps may be performed on a tape by inhibiting rewind on each dump.

    Dump of central and PPU memories occurs. Also, the S/C register contents of a CDC CYBER 170 and the CPU hardware register contents are dumped. If a CPU is logically turned off, a flag will be set in the dump denoting this. This CPU is not exchanged during EDD execution. If CPU 0 is down on a one CPU machine, both CPU 0 and CPU 1 must be logically turned off at deadstart time to avoid any exchanging of the registers.

11. EDD requests whether ECS memory is to be dumped:

    ECS SIZE/1000 0000

12. Type size
    Press CR

size	Four-digit ECS size/1000; to prevent ECS from being dumped, type 0 (or press CR without typing an entry); enter value from right to left

An area of central memory that is 1100_8 words in length is read into PPU 1 prior to the CPU hardware register dump and is restored at the completion of the EDD dump. This area is used for dumping the CPU hardware registers and ECS memory.

NOTE

If both CPUs are logically turned off, no ECS dump will be performed regardless of the value entered.

13. EDD indicates that the dump is complete with the following message.

    DUMP COMPLETE 00

14. Activate the DEADSTART switch to continue with another deadstart operation. To dump the memory that is now on tape to a printer, refer to part IV, section 4.

If a tape error occurs during the dump process, a message of the following form appears at the system display.

err CSaaaa DSbbbb 00

err	Indicates error type	
	CON	Connect reject error
	FCN	Function reject error
	WRT	Write error
aaaa	Channel converter status	
bbbb	Controller status	

Pressing CR causes EDD to retry the dump operation.

3.5 OPTION TO IGNORE ERROR PROCESSING

The nonfatal errors noted during the automatic checking of the S/C register and the presetting of PPU memories on all recovery levels of deadstart and CM memories on nonlevel 3 recoveries may be ignored during the deadstart sequence to allow the operator or customer engineer to down a PPU or equipment on a channel during CMRDECK processing. To ignore error processing, type

I

If errors occur during memory confidence testing or in the S/C register, they are displayed, but operation continues if the errors are not fatal.

3.6 OPTION TO SELECT ANOTHER CMRDECK

This option displays the current CMRDECK number and allows the selection of another CMRDECK for deadstarting the system. If a level 1 or 2 recovery deadstart is to be performed, the CMRDECK selected during the most recent level 0 deadstart must be used.

1. Type K

2. The following display occurs.

   SET CMRDECK 00

   00 Current CMRDECK number

3. Type no
   Press CR

   no One or two octal digit number of the CMRDECK desired for deadstart; enter values from right to left

4. The deadstart options display reappears.

3.7 PPO: DISPLAY PPU 0 MEMORY OPTION

Use this option to display the contents of PPU 0 memory. This function is generally used to enter sample PPU programs (to test peripheral devices on the system, for example).

1. Type P

2. PPU 0 memory is displayed on both console screens.

3. Activate the DEADSTART switch.

3.8 SCR: DISPLAY S/C REGISTER OPTION (CDC CYBER 170 ONLY)

This option displays the contents of the S/C registers of a CDC CYBER 170 and also the explanations and bit numbers of the set error status bits. The operator may set or clear bits in the S/C register via console commands, which appear in the lower portion of the display. This program assures that all errors detected via the S/C registers are acknowledged by the operator before a successful deadstart sequence can be completed, unless an operator specifically decides to ignore all errors (refer to section 3.5).

1. Type R

2. Enter one or more of the following commands.

CR	Reenter the deadstart sequence
CE.	Clear all error status bits in the S/C register
C,x,y.	Clear bit x in register y
S,x,y.	Set bit x in register y

   y=0 for the channel 16 S/C register and y=1 for the channel 36 S/C register.

   Bits set through this option may be used by diagnostics running under the Deadstart Diagnostic Sequencer (DDS). The following information identifies restrictions associated with the setting of bits by this option.

   - Any error status bit set by this option is available during the deadstart sequence and can be cleared by command under this option or by error logging once the operating system is loaded.

   - Any control bit set by this option, excluding those cleared by deadstart master clear (refer to the appropriate hardware reference manual), remains set upon entry to DDS for use by the diagnostics.

   NOTE

   The deadstart sequence must be reinitiated to call the sequencer after setting bits via this option.

3. One of the following can be done.

   a. Reenter the deadstart sequence by pressing CR

   b. Reinitiate the deadstart sequence by pressing the deadstart button

3.9 DDS: DEADSTART DIAGNOSTIC SEQUENCER

Use this option to call DDS by typing S.

Refer to the On-Line Maintenance Software Reference Manual for descriptions of the options available.

3.10 DEADSTART ANOTHER TAPE OPTION

Use this option to select a deadstart tape that is on tape equipment other than the equipment specified on the deadstart panel. The equipment to be specified must be connected to the same channel as the equipment indicated on the deadstart panel.

1. Type T

2. The following message appears on the system console.

   DEADSTART TAPE 0000

3. Mount the new deadstart tape on the tape unit connected to the same channel as the tape unit containing the current deadstart tape; this unit must be of the same type as that containing the current deadstart tape (if the current tape is mounted on a 667/669 unit, the new tape must be mounted on a 667/669 unit). Ready the unit.

4. Type e0uu
   Press CR

e	One-digit controller number of tape unit containing newly selected deadstart tape (0 if tape unit is a 667 or 669)
uu	Two-digit tape unit number

   Enter values from right to left.

5. The deadstart options display from the new deadstart tape appears on the system display.

NOTE

The load address in word 12 of the deadstart panel must be the same for both tapes.

3.11 DISPLAY AND MODIFY WORD 14 OPTION

This option is available to modify any or all parameters in word 14 of the deadstart program. To alter only bits 6, 7, or 8, refer to sections 3.12, 3.13, and 3.14.

1. Type W

2. The octal image of the current contents of word 14 (xxxx) is displayed.

   WORD 14 xxxx

3. Type rpss
   Press CR

Type entire word beginning with the r parameter, even if only changing one parameter. Refer to part II, section 2.4 (word 14) for a detailed description of the parameters.

r Designates the level of system deadstart

- 0 Initial or level 0 system deadstart
- 1 Level 1 recovery deadstart
- 2 Level 2 recovery deadstart
- 3 Level 3 recovery deadstart

p Designates the central processor control settings; this parameter can also be modified with the options described in sections 3.12, 3.13, and 3.14.

Bit 6 refers to CPU 0
Bit 7 refers to CPU 1
Bit 8 refers to the CEJ/MEJ option

ss Specifies whether the CMRDECK and IPRDECK are to be modified before loading the system library or whether deadstart is to load the system library (set to 00) without further displays.

4. The deadstart options display reappears.

3.12 DISABLE CEJ/MEJ OPTION

If the CEJ/MEJ option is enabled (word 14, bit 8), use this option to disable it.

1. Type X

The deadstart options remain displayed. The only indication that the CEJ/MEJ option is disabled is if the W option is selected; word 14 will indicate that it is disabled.

Use of this option is not recommended because the system automatically determines if the CEJ/MEJ option is present and initializes accordingly. System failure may result if the CEJ/MEJ option is present in the system and is disabled by this method. If it is necessary to disable the CEJ/MEJ option for maintenance purposes, use the keylock switch labeled CEJ/MEJ on the deadstart panel before activating the DEADSTART switch.

Once disabled, CEJ/MEJ can only be enabled by redeadstarting.

3.13 TURN OFF CPU 0/CMU OPTION

Use this option to turn off CPU 0/CMU (word 14, bit 6).

1. Type 0

The deadstart options remain displayed. The only indication that CPU 0 is turned off is if the W option is selected; word 14 will indicate that it is off.

CPU 0 and CPU 1 should not be disabled simultaneously. On dual CPU systems, disabling one CPU may be desired to run benchmarks for systems with only one CPU or may be necessary if one CPU is down.

If CPU 0 is turned off, the CMU is also turned off.

To specify CPU 0 as available, redeadstart.

3.14 TURN OFF CPU 1/CMU OPTION

Use this option to turn off CPU1/CMU (word 14, bit 7).

1. Type 1

The deadstart options remain displayed. The only indication that CPU 1 is turned off is if the W option is selected; word 14 will indicate that it is off.

CPU 0 and CPU 1 should not be disabled simultaneously. On dual CPU systems, disabling one CPU may be desired to run benchmarks for systems with only one CPU or may be necessary if one CPU is down.

If CPU 1 is turned off, the CMU is also turned off.

To specify CPU 1 as available, redeadstart.

4.1 GENERAL DESCRIPTION

The central memory resident deck (CMRDECK) is a text deck on the deadstart tape that is processed by the SET program. It contains entries defining the following types of information.

Central memory descriptions	Section 4.2
Nonmass storage equipment descriptions	Section 4.3
Mass storage equipment descriptions	Section 4.4
Track reservations	Section 4.5

There can be up to 77_8 CMRDECKs on the deadstart tape. Having several CMRDECKs on the same deadstart tape is advantageous so that one tape can be used to deadstart up to 64_{10} configurations. All CMRDECKs may be listed using the CATALOG control statement. Refer to volume 1 of the system reference manual for more information concerning CATALOG.

The released settings of the CMRDECK may be modified in two ways: by typing a new entry during deadstart when the CMRDECK is displayed or by creating a new deadstart tape. The usual method of creating a configured CMRDECK, beginning with a deadstart using the released tape, is the following.

1. Deadstart using the released deadstart tape and selecting that the CMRINST and the CMRDECK be displayed. — Part II, section 1.1

   CMRINST defines all valid CMRDECK entries. The entries listed are examples of appropriate CMRDECK entries. The released values are listed in the released CMRDECK which appears at the end of this section. Default values are described in section 4.2. Default values are assumed if the entries do not appear in the CMRDECK being used. If either the CMRDECK or CMRINST overflows two screens, the display can be advanced by pressing the + key.

2. Modify the released version of CMRDECK by entering the appropriate changes or additions from the system console. These entries can be made when either the CMRDECK or CMRINST is displayed. Each console entry supersedes the value currently specified in the CMRDECK (or default values noted for central memory entries in section 4.2). However, the modified CMRDECK remains in effect only until the next level 0 deadstart is performed. That is, changes to the CMRDECK are not recovered across deadstart unless a new deadstart tape is created to reflect them. — Part II, sections 4.2 through 4.5

3. To expedite subsequent deadstarts, modify the CMRDECK on the deadstart tape using GENSYS. — Part III, section 1.1.6

When constructing or modifying a CMRDECK, note the following restrictions.

1. The EQ entry must precede any other assignments for a device (such as assigning it for permanent file, system, or temporary file use). If the EQ entry is modified, all other assignments for that equipment must be reentered.

2. All parameters must be specified unless they are noted as optional.

3. Each entry must be separated with a comma when indicated in the entry format.

4. Each entry must be terminated with a period.

5. An arrow (↑) occurs at the position of an error in an entry typed at the system console. When there is an error in an entry in a CMRDECK on the deadstart tape, the CMRDECK is displayed with an arrow indicating the error. This occurs even if the CMRDECK is not selected to be displayed.

CMRINST Released Format:

```
CMRINST
INSTRUCTIONS FOR INITIAL SETUP OF THE OPERATING SYSTEM.

BELOW IS A LIST OF ALL SYSTEM ENTRIES.  TO MAKE THESE ENTRIES,
ENTER THEM AS INDICATED FOR THE DESIRED INITIAL CONFIGURATION.
THE ENTRY - NEXT. WILL CAUSE THE LOAD TO CONTINUE.
THE ENTRY - GO. WILL CAUSE THE LOAD TO CONTINUE
WITHOUT FURTHER DISPLAYS.  DISPLAYS BREAK 39 LINES/SCREEN.
THE RIGHT BLANK KEY TOGGLES THE DISPLAY.

DAYFILE=0,400.  DAYFILE RESIDES ON EQ 0, CM BUFFER LENGTH = 400.
ACCOUNT=0,400.  ACCOUNT RESIDES ON EQ 0, CM BUFFER LENGTH = 400.
ERRLOG=0,100.  ERRLOG RESIDES ON EQ 0, CM BUFFER LENGTH = 100.
FNT=1000. SET FNT LENGTH = 1000.
NCP=17.   SET THE NUMBER OF CONTROL POINTS = 17.
PPU=X,Y,...Z.  TURN OFF PPU X,Y,...,Z.  (0,1,2,10 ILLEGAL)
NAME=CCC-CCC. SET THE SYSTEM NAME = CCC-CCC.
VERSION=CCC-CCC.  SET VERSION NAME = CCC-CCC.
IPD=0.    ASSEMBLE INSTALLATION PARAMETER DECK 0.
LIB=N.    BUILD SYSTEM USING LIBDECK N (N = 0-7)
MID=MM.   SET MACHINE ID = MM (DEFAULT MNEMONIC = *AA*)
CM=XXXX.  SET CENTRAL MEMORY SIZE TO XXXX HUNDRED WORDS.
PRESET,N. PRESET MMF LINK DEVICE FOR *N* SHARED EQUIPMENTS.

EQXX=TY,ST,EN,UN,A,B,C,D,OP. DEFINE EQUIPMENT XX AS FOLLOWS
          TY = TYPE (2 LETTERS)
          N  = NUMBER OF CONTIGUOUS UNITS STARTING AT *UN*
          ST = STATUS (ON, OFF)
          EN = EQUIPMENT NUMBER
          UN = UNIT NUMBER
          A - D = CHANNELS
          OP = TAPE HARDWARE OPTION (1, 2, 4, OR 20)
EQXX=YYYY.  ENTER YYYY AS OCTAL ENTRY FOR EQ XX.
EQXX=DE,ST,1000.  SET ECS EQUIPMENT 1000K (250K).
EQXX=DD-N,ST,EN,UN,A.    ENTER 853/854-N FOR EQXX.  (N=1-4)
EQXX=DI-N,ST,EN,UN,A,B. ENTER 844-21-N FOR EQXX.  (N=1-8)
EQXX=DJ-N,ST,EN,UN,A,B. ENTER 844-41-N FOR EQXX.  (N=1-8)
EQXX=MD-N,ST,EN,UN,A,B. ENTER 841-N FOR EQXX.     (N=1-8)
EQXX=MT-N,ST,EN,UN,A,B,C,D,OP. ENTER CONSECUTIVE MAGNETIC TAPES.

RESERVE TRACKS.
     CTK=X,Y,..Z.   CLEAR PREVIOUS RTK,STK AND TTK ON EQX,Y,..Z.
     STK=XX,NNNN.   SET RESERVATION ON LOGICAL TRACK NNNN.
     TTK=XX,CCCC.   TOGGLE RESERVATION. (SAME FORMAT AS RTK.)
     RTK=00,T200,G10,S144. DA 00, 6603.
     RTK=01,P40,H40,S144.  DB 01, 6638.
     RTK=02,U10,G100.      DC 02, 3536/863.
     RTK=03,C310.          DD 03, 854-N.
     RTK=04,A757500.       DE 04, ECS.
     RTK=05,C200,T10000.   DF 05, 813.
     RTK=06,C2000.         DH 06, 3553/821.
     RTK=07,C630,S2130.    DI 07, 7054/844-21-N.
     RTK=10,C1464,S2230.   DJ 10, 7054/844-41-N.
     RTK=11,C310,S2316.    MD 11, 3553/841-N.
```

```
          THE FOLLOWING ENTRIES ARE CLEARED IF EQ IS REDEFINED -
ASR=X,Y,...,Z. SET ALTERNATE SYSTEM DEVICES.
FAMILY=NN. SET EQUIPMENT NUMBER OF DEFAULT FAMILY.
LINK=XX.   SET EQUIPMENT XX AS MMF LINK DEVICE (XX .NE. 0)
PF=XX,TY,DM,SM,FM,DN,NC.  SET PF CONTROLS FOR DEVICE XX.
          APPLIES IF INITIALIZE,XX,AL. IS USED, OTHERWISE
          PARAMETERS ARE TAKEN FROM THE DEVICE LABEL.
          TY = TYPE OF PERMANENT FILE RESIDENCE.
                 *F* = FAMILY DEVICE.
                 *X* = AUXILIARY DEVICE.
          DM = DEVICE MASK.
          SM = SECONDARY MASK.
          FM = FAMILY NAME (1-7 CHARACTERS)
          DN = DEVICE NUMBER.
          NC = CATALOG TRACKS (POWER OF 2 .LE. 200)
INITIALIZE,XX,OP.
          INITIALIZE DEVICE XX BASED ON THE OPTION *OP*.
          THE OPTIONS ARE AL, PF, QF, DF, AF, EF, FP.
          ONLY THE *AL* AND *FP* OPTIONS ASSUME A BAD LABEL.
REMOVE=X,Y,...Z. SET X,Y,...Z AS REMOVABLE DEVICES.
SHARE=X,Y,...Z.  DEFINE SHARED EQUIPMENTS FOR MMF SYSTEM.
SYSTEM=X,Y,...Z. SET X,Y,...Z AS SYSTEM DEVICES. (ALL SAME TYPE)
TEMP=X,Y,...Z.   SET X,Y,...Z FOR SYSTEM ALLOCATION OF SPACE.

          DEADSTART OPTIONS (EACH ENTRY TOGGLES OPTION) -
AUTOLOAD.  DISABLES AUTOLOADING OF BUFFER CONTROLLERS.
GRENADE.  SELECTS GRENADE OPERATION AFTER AUTOLOADING.
```

Format of the CMRDECK on the Released Deadstart Tape:

```
CMRDECK
MID=27.
NCP=17.
FNT=2000.
EQ0=DI-1,ON,0,0,0.          SYSTEM,TEMP     P0503(AX)
EQ1=DI-1,ON,0,12,1.         SYSTEM,TEMP     P0510(AU)
EQ2=DI-1,ON,0,33,1.         TEMP
EQ3=DJ-1,ON,0,6,1.          TEMP
EQ4=DI-2,ON,0,1,0.          PF-S(DEV 40)    P0506,P0507(AY,AZ)
EQ5=DI-2,ON,0,22,1.         PF-S(DEV 41)    P0508,P0509(BG,AT)
EQ6=DI-1,ON,0,5,0.          REMOVE
EQ7=DI-1,ON,0,3,0.          REMOVE
EQ10=DS,ON,7,0,10.
EQ12=DI-1,ON,0,5,1.         REMOVE
EQ13=DI-1,ON,0,4,1.         REMOVE
EQ14=DI-1,ON,0,4,0.         REMOVE
EQ15=MD-1,ON,6,3,4.         REMOVE
EQ30=CR,ON,4,,12.
EQ31=CP,ON,5,,12.
EQ32=LP,ON,7,,12.           580-20
EQ33=LP,ON,6,,12.           512
EQ34=LP,OFF,3,,12.          512
EQ40=TT,ON,7,,3,,50.        6676
EQ41=TT,ON,0,1,5,,0,1.      TIME-SHARING STIMULATOR
EQ42=TT,ON,7,2,5,,30.       TRANEX STIMULATOR
EQ43=ST,ON,7,,2.            6671
EQ50=MT-7,ON,0,0,13,,,,20.  66X
PF=4,F,252,377,CLS127,40.
PF=5,F,125,377,CLS127,41.
TEMP=0,1,2,3.
REMOVE=6,7,12,13,14.
FAMILY=4.
```

4.2 CENTRAL MEMORY DESCRIPTIONS

The following entries are already specified in the SET program with the released default values indicated. These values are appropriate for most installations.

The general function of these entries is to determine the amount of central memory that is to be used for central memory resident and the amount to be used for job processing. The simplified relationship is that the more central memory that is assigned to dayfile buffers, the less is available for job field lengths.

If, for example, a large portion of central memory is needed to run a job, it might be advisable to decrease the size of the dayfile buffers area in central memory resident (CMR) in order to accommodate that job. However, when the buffers are smaller, the information stored in them is written to mass storage more often, thus requiring more system overhead. If only a few batch jobs are to be run, fewer control points may be required. Thus, the control point area in CMR (which requires 300_8 words per control point: 200_8 for the control point area and 100_8 for the dayfile buffer) could be decreased also.

Entry Format	Released Default Value	Significance
ACCOUNT=eq, length.	400_8	This entry sets the residence and length of the account dayfile buffer. The account dayfile is an accounting record containing messages indicating information such as kind and amount of resources used, as well as jobs and execution times. This account information is written to the central memory account file buffer during job processing and the central memory buffer is written to mass storage when the buffer is full. The account file buffer resides in CMR in the dayfile buffer area. eq — 1 or 2 octal digit EST ordinal of equipment on which the account dayfile is to reside. The residence of this file is normally determined by the recovery of the existing dayfile. This parameter is used if no dayfiles are recovered. length — 3 or 4 octal digit length of account file buffer; must be a multiple of 100_8
CM=size.	Available core size	The available size of core memory is automatically determined by the SET program during deadstart. The available size is equal to or less than the actual size of the machine. It is less than the actual size on a CDC CYBER 170 if memory is being operated in a degraded mode. This command may be used to enter any size of memory as long as it is less than the available size and any previous entries via this command. If a value much below 1000B is entered, the system may not deadstart properly. Memory phasing is such that the entry of CM=2000 causes the upper half of memory to be ignored on all CDC CYBER 70 machines with 98K or 131K words of memory.

Entry Format	Released Default Value	Significance
		The CM parameter allows the actual size to be decreased. This may be necessary, for example, to try to benchmark a configuration that has less central memory size than is available on the computer. size — 1-to 4-octal digit value that restricts the actual core size; this value represents the amount of central memory in hundreds; value cannot be 0 and cannot exceed actual core size.
DAYFILE=eq, length.	400_8	This entry sets the dayfile buffer residence and length. The dayfile buffer contains the dayfile information that is maintained in the same way as the account file buffer. It resides in CMR in the dayfile buffer area. eq — 1 or 2 octal digit EST ordinal of equipment on which the dayfile is to reside. The residence of this file is normally determined by the recovery of the existing dayfile. This parameter is used if no dayfiles are recovered. length — 3 or 4 octal digit length of dayfile buffer in CMR; must be a multiple of 100_8
ERRLOG=eq,length.	100_8	This entry sets the residence and length of the error log dayfile buffer. The error log dayfile is a record of error messages along with execution time for a particular job. This information is maintained in the same manner as the account file buffer. eq — 1 or 2 octal digit EST ordinal of equipment on which the errlog dayfile is to reside. The residence of this file is normally determined by the recovery of the existing dayfile. This parameter is used if no dayfiles are recovered. length — 3 or 4 octal digit length of the error log buffer which resides in the dayfile buffer area of CMR; must be a multiple of 100_8
FNT=length.	1000_8	This entry sets the length of the file name/file status table. This table consists of two one-word entries for each active file, the file name (FNT) and the file status (FST). The table resides in CMR. length — 3 or 4 octal digit length of FNT and FST area; must be multiple of two; minimum value is 100_8

Entry Format	Released Default Value	Significance
IPD=iprdeck.	First IPRDECK on deadstart tape	This parameter indicates which IPRDECK to use at deadstart. The IPRDECK contains installation parameters defined in part II, section 5. There can be up to 4096 IPRDECKs on a deadstart tape. If an IPD entry is not included in the CMRDECK to be used, the first IPRDECK on the deadstart tape is processed without being displayed. If this parameter is not entered, the first deck (IPRDECK) is processed without being displayed.
LIB=n.	n=0 Use record named LIBDECK	This entry indicates the LIBDECK to be used in building the system. LIBDECK is a directive record used by SYSEDIT. It is discussed in part III, section 1.1.4. If a LIB entry is not included in the CMRDECK being used, the LIBDECK is selected from the deadstart panel (word 13, www). Conversely, if the LIB entry is present, it overrides the deadstart panel setting.
MID=id.	AA	This entry specifies the two-character machine identification which is maintained in the MMFL word of CMR. The characters supplied must be alphanumeric.

Entry Format	Released Default Value	Significance
NAME=date line.	CDC MULTI-MODE OPERATING SYSTEM	This parameter specifies the system date line that is displayed on the system console display. It is stored in words 32 through 35 of CMR. date line — Alphanumeric character system date line; must be less than 39 characters in length
NCP=number.	12_8	This entry sets the number of control points available for job processing to a value other than the default value. Refer to section 5.8.2 for a discussion of the proper number of control points to select. number — 1 or 2 octal digit number of control points available in central memory; maximum is 27_8; value stored in CMR word 2, bits 12 through 23
PPU=*,	PPUs 10 through 20 are available if they exist	The released system determines the number of PPUs hardwired into the system. To indicate that only 10 PPUs are available on a 20 PPU machine, enter this command in the CMRDECK. Reentering the command toggles the setting so that the upper 10 PPUs are available.
PPP=ppu,...,ppu.	All available PPUs are active	This command pertains to any physically available PPU. Its purpose is to change the status of any specific PPU except for 0, 1, 2, or 10, which must always be active. Active status means available for system use; inactive means not available for system use. This is a toggle entry; reenter to change status. This command may be useful if PPU memory is failing or if a channel is causing problems on its associated PPU. ppu — 1 or 2 octal digit PPU number of PPU whose status is to be changed. Any number of PPUs can be specified separated by commas. For example, the following (if there are not other entries of this type) deactivates PPU 3 and PPU 4. PPU=3,4
PPU=2X.	500-nanosecond cycle	This entry toggles the status of the 2X PPU option. This option selects either the 500-nanosecond or 1000-nanosecond PPU cycle. This command applies only to CDC CYBER 170 machines.
VERSION=name	NOS 1.1 419/420	This parameter specifies the system version that is displayed on the system console display. It is stored in words 36 and 37 of CMR. name — Alphanumeric character version name; must be less than 19 characters in length

4.3 EQUIPMENT ASSIGNMENTS: NONMASS STORAGE

The following EST entries are described in this section.

Clear EST assignment	section 4.3.1
Nonstandard equipment EST entry	section 4.3.2
Dummy equipment EST entry	section 4.3.3
System console display equipment EST entry	section 4.3.4
Unit record equipment EST entry	section 4.3.5
Magnetic tape equipment EST entry	section 4.3.6
Multiplexer equipment EST entry	section 4.3.7

4.3.1 CLEAR EST ASSIGNMENT ENTRY

Use the following entry to clear an assignment that currently exists for an EST ordinal. Clearing the assignment does not clear flaw entries for that equipment.

```
EQord=0.
  or
EQord=.
```

ord 1 or 2 octal digit EST ordinal of equipment; 0 through 75_8

4.3.2 NONSTANDARD EQUIPMENT EST ENTRY

The nonstandard EST entry is available so that an analyst can define nonstandard equipment or can add his local debugging modifications.

It is possible to make a nonstandard EST entry at two different times, deadstart or during system operation.

During deadstart, use the nonstandard EQ entry described in the following format to enter the actual octal value that is to reside at that EST ordinal.

During system operation, use a DSD memory entry command (refer to section 3 of the operator's guide) to enter the actual octal value to reside in the EST. Specify the low core location of this value using the octal address of the EST ordinal (instead of the EST ordinal number). The octal address can be obtained by using the DIS E display (section 7 of the operator's guide).

The format of the nonstandard EST entry is:

```
EQord=value.
```

ord 1 or 2 octal digit EST ordinal of equipment; 0 through 75_8

value 1 to 20 octal digit value; this value is entered in the EST word for the specified ordinal.

4.3.3 DUMMY EQUIPMENT EST ENTRY

EST ordinals 76_8 and 77_8 are EST entries that are automatically reserved by the system; they cannot be used for other assignments and are the only dummy entries allowed in the EST.

Even though they are automatic dummy entries, they do not appear as EQ assignments when the CMRDECK is displayed. If it is desirable that they be displayed along with the CMRDECK, they can be entered using the following format.

```
EQord=type, status,,,.
```

ord — EST ordinal of dummy equipment; 76_8 for TE; 77_8 for NE

type — Equipment type:

TE	Equipment type for 76_8
NE	Equipment type for 77_8

status — Specifies whether equipment is available for use

ON	Available
OFF	Equipment is ignored during system operation

76_8 ENTRY

If a file name-volume serial number association is established with an ASSIGN, LABEL, REQUEST, or VSN control statement, the system automatically enters EQ76 in the file's FNT/FST entry. When a tape having the desired volume serial number is assigned to the file, the system replaces EQ76 in the file's FNT/FST entry with the EST ordinal of the tape unit on which the tape is mounted. If a file which has had the file name-volume serial number association established by a VSN control statement is returned prior to attempting to assign tape equipment to the file, the FNT/FST entry is cancelled.

77_8 ENTRY

77_8 is used internally by the system to signify that a file is assigned but that there is no space on the device. If a read is tried, end of information status (EOI) occurs. If a write is tried, an infinite sync occurs; the data is discarded.

For example, it can be used with the permanent file utility to validate the integrity of a permanent file device without taking the time to actually create a dump file on tape. In this case, when the permanent file utility issues the request for an archive file, enter the DSD command n. ASSIGN,77. (Refer to part IV, section 2. 2.) This causes all dump data to be discarded even though the permanent file device is read and informative messages about the permanent file device are issued to the system console. These messages are described in part IV, section 2.

4.3.4 SYSTEM CONSOLE DISPLAY EQUIPMENT EST ENTRY

NOS requires at least one system console to be available for use.

```
EQord=DS, status, controller,0, channel.
```

ord	1 or 2 octal digit EST ordinal of display; 1 through 75_8; most sites set to 10
DS	Display console equipment type
status	2 or 3 digit indicator of whether equipment is available for system use ON — Available OFF — Can use console only for the DSD commands
controller	1 or 2 digit number of system console controller; 0 through 7
channel	1 or 2 octal digit number of channel to which console equipment is connected; 0 through 13_8, 20 through 33_8; most configurations use channel 10
Example:	EQ10=DS,ON,7,0,10.

4.3.5 UNIT RECORD EQUIPMENT EST ENTRY

The recommended amount of unit record equipment is two line printers and one card reader.

```
EQord=type, status, controller, unit, channel.
   or
EQord=type-n, status, controller, unit, channel.
```

ord — 1 or 2 octal digit EST ordinal of equipment; 1 through 75_8

type — 2-digit equipment type; the following unit record equipment is supported with NOS.

Equipment	Type
Card reader	
405-3447/3649	CR
Card punch	
415-3446/3644	CP
415-30	CP
Line printer	
512-1-3555-1	LP or LQ
580	LP or LR

n	Specifies print train for 512 and 580 line printers; 1 through 6; the following print trains are currently supported by NOS.

Line Printer	Print Train	n
512-1-3555-1	595-1	1
	595-5	5
580	596-1	1
	596-5	5

If a nonsupported print train value is set, n is defaulted to a supported value. If n is omitted or set to 2, 3, or 6, n is defaulted to 1. If n is set to 4, the default value is 5.

status	Specifies whether equipment is available for system use ON — Available OFF — Unit is ignored during system operation
controller	1 - digit controller number for equipment; 0 through 7
unit	Not applicable for unit record equipment; enter a 0 or a comma
channel	1 or 2 octal digit number of channel to which equipment is connected; 0 through 13_8, 20 through 33_8

NOTE

When performing a coldstart, a card reader should be available on channel 12 or 13. Also, to ensure that all printers are restored to their original states (such as 8 lines per inch, auto page eject) after a master clear has been issued, all unit record equipment should be available on dedicated channels. If this is not done, printers revert to 6 lines per inch, no auto page eject status after a master clear is issued.

Example:

```
EQ11=CR,ON,4,,12.
EQ12=CP,ON,5,,12.
EQ20=LP,ON,6,,12.
```

4.3.6 MAGNETIC TAPE EQUIPMENT EST ENTRY

The minimum number of magnetic tape devices that NOS requires is two 657's, 659's, 667's, or 669's.

EQord=MT-n, status, equipment, unit, $chan_1$, $chan_2$, $chan_3$, $chan_4$, option
or
EQord=NT-n, status, equipment, unit, $chan_1$, $chan_2$, $chan_3$, $chan_4$, option

ord	1 or 2 octal digit EST ordinal of equipment; 1 through 75_8; most sites use 50_8. Refer also to the MT-n or NT-n parameter.
MT-n or NT-n	Equipment type; n is total number of magnetic tape units connected to controller. The system automatically generates n number of EST entries with consecutive EST ordinals beginning with the ordinal specified in the first parameter. The n units begin with the unit number specified in the unit parameter. The following magnetic tape equipment is supported with NOS.

For MT:

Controller	Units	n
3518-1/2/3	657	$1\text{-}10_8$
3528-1/2/3	657	$1\text{-}10_8$
7021	667	$1\text{-}10_8$

For NT:

Controller	Units	n
3518-1/2/3	659	$1\text{-}10_8$
3528-1/2/3	659	$1\text{-}10_8$
7021	669	$1\text{-}10_8$

To clear an MT-n or NT-n assignment, enter an EQord=0 entry for all n units.

status — Indicates whether equipment is available for access

ON	Available for access
OFF	Unit is ignored during system operation

equipment — 1-digit controller equipment number; ignored for 667 and 669 tape units and should be set to 0; otherwise must be 4 through 7

unit — 1 or 2 octal digit number of lowest numbered magnetic tape unit to be processed; units must have consecutive physical unit numbers; 0 through 7 for 667 and 669 units, otherwise 0 through 17_8.

chan_{1-4} — 1 or 2 octal digit number of channel to which equipment is connected; 0 through 13_8, 20 through 33_8. If the upper bit of the channel number is set (40 through 53_8, 60 through 73_8), a 6684 is on the channel. A 6684 is used for conversion only when conversion memory does not exist. Conversion memory is used for BCD conversion.

A controller can be connected to from one to four channels, depending on the controller model. However, a maximum of four channels can be handled regardless of the number of controllers.

NOTE

There must be a tape drive available on channel 12 or 13 for the deadstart tape. When deadstarting (coldstarting) from a 667 or 669 magnetic tape unit, the tape unit and a card reader must be available on channels without additional PPUs. The usual configuration is for the card reader to be assigned to channel 12 and the tape unit to channel 13.

option	Optional hardware feature parameter		
		0 or omitted	No significance
		1	**Status 2 hardware feature is not available.**
		2	Set option to 2 if the conversion memory (code conversion) feature is not available. This option must be used for 3518-1 and 3528-1.

CAUTION

The following option is provided only for diagnostic purposes and should never be used in a normal production environment.

		4	This option disables the use of controlled backspace and the selection of abnormal EOP on read operations. This option must be selected unless all 6681/6684s on tape controller channels contain the FCOs to recognize the 1460 and 1660 function codes.
		10	Programmable clipping is not available. This option applies for **657 and 659 tape units.**
		20	This option must be selected for the 667/669 tape drive (667 or 669 tape units). When this option is selected, no others have significance.

NOTE

Some 3x2x controllers do not properly reject the functions used to test for the availability of conversion memory. The user can avoid this problem by specifying an option number of 3 or 7, whichever is appropriate.

4.3.7 MULTIPLEXER EQUIPMENT EST ENTRY

There are three types of EST entries for communication equipment:

- Time-Sharing Module/Transaction Subsystem (TRANEX) entries
- Time-Sharing Module and Transaction Subsystem stimulator entries
- Export/Import entry

TIME-SHARING MODULE/TRANSACTION SUBSYSTEM (TRANEX) ENTRIES

The format is:

```
EQord=TT, status, controller, 0, channel, 0, lines.
```

ord	1 or 2 digit EST ordinal of multiplexer; 1 through 75_8
TT	Equipment type for Time-Sharing Module and Transaction Subsystem; 6671 or 6676
status	Specifies whether the equipment is available for use ON — Available OFF — Equipment is ignored during system operation
controller	1 or 2 digit number of multiplexer controller; can be 0 through 7
0	Indicates that this is not an EST entry for the stimulator
channel	1 or 2 octal digit number of channel to which multiplexer equipment is connected; 0 through 13_8, 20 through 33_8
0	Parameter is not used
lines	1, 2, or 3 octal digit number of lines available 6671 — 1 through 20_8 (16 lines is maximum); if 0 is specified, a channel hang occurs. 6676 — 0 through 100_8; the terminal driver scans only the number of lines specified. For example, if the lines parameter is set to 3, the driver only scans lines 1 through 3. Specifying fewer lines also saves central memory space in TELEX. The driver scans all 64 (0 through 100_8) lines if the parameter is omitted.

Examples:

EQ30=TT, OFF, 0, 0, 4, 0, 20.	6676 described with 20 of 64 lines available; multiplexer not available for use
EQ41, TT, OFF, 7, 0, 3, 0, 10.	6671 to be used for time-sharing terminals; 10 of 16 lines are available; multiplexer not available for use

Standard operation of the 6676 and the 6671 data set controllers allows communication with terminals at a line speed of 10 characters per second (cps). There are options that allow other line speeds to be connected to the 6676. NOS supports all of these up to 600 baud.

NOS also supports the 6676 option to communicate with terminals that issue the 9-bit code. These are usually referred to as correspondence type terminals.

However, when configuring the data sets to the 6676, be sure that the proper terminals are always connected to the proper lines; that is, 10 cps terminals to 10 cps lines and 30 cps terminals to 30 cps lines.

SUBSYSTEM STIMULATOR EST ENTRIES

The time-sharing stimulator is described in part IV, section 5; the transaction stimulator is described in the TRANEX documentation.

The stimulator EST entry format is:

```
EQord=TT, status, controller, num, channel, 0, lines.
```

ord	1 or 2 octal digit EST ordinal of multiplexer; 1 through 75_8
TT	Equipment type for time-sharing stimulator or Transaction Subsystem
status	Specifies whether the equipment is available for use ON — Available OFF — Equipment is ignored during system operation
controller	1 or 2 octal digit number of multiplexer controller; can be 0 through 7
num	Designates the subsystem 1 — Time-sharing stimulator 2 — Transaction subsystem stimulator; indicates to system that communication with 1TD is to be direct, without functioning the channel 4 — Transaction subsystem stimulator with the communications stimulating a 6671 multiplexer
channel	1 or 2 octal digit number of channel to which multiplexer equipment is connected; 0 through 13_8, 20 through 33_8
lines	Number of lines to stimulate; must be less than 1000_8. 100_8 is default indicated with 0 entry. For the TRANEX stimulator, the value specified must be the same as the number of lines specified in NETWid or SIMFid (where id is the machine id). If num parameter is 4, the lines parameter must be 1 through 100_8 (limitation of 64 lines).

Examples:

EQ37=TT,ON,7,1,2,0,0.	Time-sharing stimulator EST entry that allows 100_8 terminals to be stimulated using channel 2 as communications channel
EQ41,TT,ON,7,2,5,0,0.	Transaction subsystem stimulator EST entry that allows 100_8 lines to be stimulated

EXPORT/IMPORT EQUIPMENT EST ENTRY

```
EQord=ST,status,controller,0,channel.
```

ord	1 or 2 octal digit EST ordinal of multiplexer; 1 through 75_8
ST	Equipment type for 6671 multiplexer used as a synchronous low or medium speed remote batch terminal
status	Specifies whether the equipment is available for use ON — Available OFF — Equipment is ignored during system operation
controller	1 or 2 digit number of multiplexer controller; can be 0 through 7
0	Not used
channel	1 or 2 octal digit number of channel to which multiplexer equipment is connected; 0 through 13_8, 20 through 33_8

Example:

EQ40=ST,OFF,7,0,3.	6671 described with 16 lines available; multiplexer not available for use

4.4 MASS STORAGE EQUIPMENT ASSIGNMENTS

This section describes the following mass storage assignments.

Mass storage equipment EST entry	EQ	section 4.4.2
Autoloading of 7054/844 controller	AUTOLOAD	section 4.4.3
Clear 7054 reservations entry	GRENADE	section 4.4.4
ECS equipment EST entry	EQ	section 4.4.5
Temporary files device assignment	TEMP	section 4.4.6
Permanent files device assignment	PF	sections 4.4.7, 4.4.8
System library device assignment	SYSTEM	section 4.4.9
Alternate system library device assignment	ASR	section 4.4.10
Default family name assignment	FAMILY	section 4.4.11
Removable device assignment	REMOVE	section 4.4.12
Initialization command	INITIALIZE	section 4.4.13
Shared device designation	SHARE	section 4.4.14
Link device declaration	LINK	section 4.4.15
Preset multimainframe link device	PRESET	section 4.4.16

4.4.1 NOS MASS STORAGE CONCEPTS

The following descriptions and chart summarize the NOS mass storage terminology and the kinds of mass storage assignments that can be specified in the CMRDECK. Other terms are defined along with the CMRDECK descriptions in the following sections.

ALTERNATE SYSTEM DEVICE

Whereas a system device contains all routines in the system library, an alternate system device contains copies of selected system library routines. The ASR entry in CMRDECK (part II, section 4.4.10) specifies which mass storage devices are to contain system library routines; the *AD LIBDECK entry (part III, section 2.6) on the deadstart tape specifies which system library routines are to reside on these mass storage devices. During system processing, the routines on the alternate system device are used instead of the ones on the system device.

This feature is valuable because it allows each routine in the system library to reside on the mass storage device that is most appropriate to its use. For example, if the system device is an 844, a routine that is frequently used could use ECS, which has a faster transfer rate, as an alternate system device instead of the system device.

ALTERNATE PERMANENT FILE FAMILY

There can be more than one permanent file family in a system. One is always the default permanent file family that is specified with the FAMILY CMRDECK entry as the default family. If another system's permanent file family is introduced to that system, it is an alternate permanent file family. It can be added without interrupting the default permanent file family's operation.

This is a useful feature if a site has more than one system or has groups of installations. If one system fails, its permanent files can be accessed from another system.

An an example, a site with two systems might run with the following mass storage configuration.

System	Ord	Device	Spindles	Access Used	Contents
X	0	844	2	A	Direct access files
X	1	841	4	A	Master device, indirect access files
Y	0	844	2	B	Direct access files
Y	1	841	4	B	Master device, indirect access files

If system Y became inoperative, the B accesses for the 844's and the 841's could be connected to system X. This could be done without interrupting system X's operations.

The CMRDECK entries in system X would be:

EQ0=DI-2, ON, 0, 0, 0.	(defines access A)
EQ1=MD-4, ON, 4, 0, 2.	(defines access A)
EQ2=DI-2, OFF, 0, 0, 1.	(defines access B)
EQ3=MD-4, OFF, 4, 0, 3.	(defines access B)
REMOVE=2, 3.	(allows introduction of access B into system X during operation)

The CMRDECK in system Y would be:

ED0=DI-2, ON, 0, 0, 1.	(defines access B)
EQ1=MD-4, ON, 4, 0, 3.	(defines access B)
EQ2=DI-2, OFF, 0, 0, 0.	(defines access A)
EQ3=MD-4, OFF, 4, 0, 2.	(defines access A)
REMOVE=2, 3.	(allows introduction of access A into system Y during operation)

To allow for introduction of an alternate permanent file family:

1. The equipment to be introduced or removed must be defined in the CMRDECKs for both systems. **(Refer to the previous example.)**

2. Specify all of the equipment that may be introduced or removed during system processing as removable.

3. When it is desired to introduce the equipment into a system, use the ON operator command to indicate that the equipment that is set to the OFF position in the system in operation is now available. This introduces the alternate permanent file family.

AUXILIARY DEVICE

An auxiliary device is a mass storage device that is not part of a family. It is a supplementary permanent file storage device that may be privately owned (PRIVATE) or may be shared by many users (PUBLIC). On the PF entry for an auxiliary device (for a pack device as well as a fixed device without packs), a pack name is specified instead of a family name.

An auxiliary device can reside on a removable or nonremovable device.

Refer to volume 1 of the system reference manual for a detailed description of PRIVATE and PUBLIC.

Example:

Four 841 spindles to be used as an auxiliary device could be defined as follows:

```
EQ3-MD-4,ON,5,1,2.
PF=3,X,name.
```

FAMILY DEVICE

A family device can be a removable device or a nonremovable device. The only distinction is that a nonremovable device containing permanent files can also contain a copy of the system library and/or temporary files. Refer to the alternate permanent file system description.

On the PF entry, the family concept is only important if two systems' permanent files are to run on the same system. A user can only use one family of permanent files; if he does not specify one, the default FAMILY entry is used.

A family device can contain direct and/or indirect access files. These files are defined in volume 1 of the system reference manual. The files that will be allowed are set with the type parameter on the PF entry.

LINK DEVICE

Extended core storage is the medium through which several computer systems are linked together to form a multimainframe operating enironment (MMF). The link device contains information necessary for the orderly management of that mass storage which can be shared by more than one machine.

Example:

An ECS to be used as a link device could be defined as follows:

```
EQ11=DE,ON,1000.
LINK=11.
```

Use of DDP could be defined as follows:

```
EQ11=DP,ON,1000,27.
LINK=11.
```

MASTER DEVICE

The master device is a device in a permanent file family that contains all permanent file catalog entries and indirect access files for a specific user. The user's master device must be available on the running system if permanent file access is required, unless all access is to be to an auxiliary device. The user index (refer also to part IV, section 2.1) and family name uniquely describe a user's master device. To access an auxiliary device, the user must specify the pack name as part of each permanent file request.

Each master device is organized into five logical sections.

1. Allocation Information

   The device label contains information describing the device, such as family name and user mask, as well as locations of permit and catalog information and indirect access files. Refer to the INITIALIZE entry.

   The track reservation table (TRT) is the key to allocating information on this device and to describing the physical layout of data on the device. Refer also to part II, section 4.5 and part IV, section 2.1.

2. Catalog Information

   The catalogs for a master device are allocated to contain catalog entries for a specific group of user indices. A particular catalog track may contain entries for many users, the number depending upon the number of catalog tracks defined for the device. The user index provides the mechanism for differentiating between user's files on a particular catalog track. Refer to part IV, section 2.1 for a more detailed description.

3. Permit Information

   A user can allow other users to access his permanent files. This can be done by implicitly or explicitly permitting the user to access a particular file. Refer to the permanent file commands in volume 1 of the system reference manual. Information describing the permission for all permanent files is in the permit file. Catalog entries contain a relative sector address within this permit file for permissions that have been granted for the file.

4. Indirect Access Files

   The master device for a user contains all of his indirect access files. These files can be accessed by commands that generate working copies for manipulation by the user.

5. Direct Access Files

   Direct access files are files that can be accessed from their location on mass storage. A working copy is not generated, so any updates or alterations made to the file are permanent. Direct access files can reside on the family master device or on any other device in the family.

MULTISPINDLE DEVICE

To accommodate files that are larger than one device, there is the multispindle device assignment. Up to eight spindles can be included in the EQ definition of one logical device. This definition must occur when the device is first defined. All spindles must be available for access whenever the device is accessed.

Multispindle devices are treated as one logical device, having a track size equal to n times the single-spindle track size (where n is the number of spindles in the device). The tracks of an n-spindle device are broken down into n equal-sized segments which have a length equal to the single-spindle track size. Each segment is contained on a different physical unit. The addressing scheme used to access the n segments making up a logical track requires that the physical units containing the segments have contiguous unit numbers and that the segments of a track are located on the contiguous physical units in order of ascending unit number. For all equipment other than 844 type, contiguous unit numbering is identical to consecutive unit numbering. Refer to 844 Expander in this section for an explanation of contiguous unit numbering for 844 type equipment.

For example, four 841 spindles used as one device could be defined in the CMRDECK as:

EQ3=MD-4, ON, 4, 1, 2.

Refer to section 4.4.2 (type and unit parameters) for specific information on assigning multispindle devices.

844 EXPANDER

A nonexpanded controller (7054) can have up to eight disk drives connected to it. Each of the connection paths is called a port and is identified by a port number ranging from 0 through 7. An expander (10304 extender) is a hardware device that can be connected between controllers and 844 disk drives to increase the number of disk drives that each controller can access. This expander can be used only with 844-21 drives although all equipment definitions and equipment driving software support the 64 drive addressing scheme for both 844-21 (DI) and 844-41 (DJ) type equipment. Each expander consists of either two or four expansion elements. An expansion element connects to a single controller port and forms a connection path from that port to one of up to eight disk drives. The connection paths between an expansion element and the eight possible disk drives are called ranks and are identified by a rank number ranging from 0 through 7. Two expanders with four expansion elements each can be connected to a single controller to allow that controller to access up to 64 disk drives. Each expansion element, however, is logically independent and, as such, could be connected to any port of any controller.

A single controller maximum configuration can be visualized as an 8 by 8 checkerboard with each square representing one of 64 disk drives (Figure II-4-1).

Controller Port

Expander Rank	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	10	11	12	13	14	15	16	17
2	20	21	22	23	24	25	26	27
3	30	31	32	33	34	35	36	37
4	40	41	42	43	44	45	46	47
5	50	51	52	53	54	55	56	57
6	60	61	62	63	64	65	66	67
7	70	71	72	73	74	75	76	77

Unit Numbers (octal)

Port Digit

Rank Digit

Figure II-4-1. Expander Addressing Map

A column of squares in Figure II-4-1 represents all drives which are accessed through a single controller port. A row of squares represents all drives which have the same expander rank. Each disk drive that can be accessed by the controller is addressed by a 6-bit unit number. The rightmost three bits of this unit number select which of eight controller ports the drive is connected to. The leftmost three bits of the unit number select which of eight ranks on an expansion element the drive is connected to. This unit number is specified as a 2-digit octal number in the mass storage equipment EST entry (refer to section 4.4.2). The right digit (port digit) of the unit number is the port number and the left digit (rank digit) is the rank of the unit in the particular expansion element.

If two disk drives are vertically adjacent on the expander addressing map (Figure II-4-1), their unit numbers are considered to be vertically contiguous. That is, both drives are connected to the same expansion element, both have the same port number, and their rank numbers differ by one (refer to example 1).

If two disk drives are horizontally adjacent on the expander addressing map, their unit numbers are considered to be horizontally contiguous. That is, both drives have the same rank number, and their port numbers differ by one (refer to example 2). The special case of horizontally contiguous, in which the rank number is zero, is equivalent to the definition of contiguous unit numbers for other equipment.

All drives connected to a controller, either directly or indirectly, through an expansion element are supported as single unit or multiunit logical devices. Unit numbers can range from 0 through 77_8, rather than from 0 through 7, as for other equipment. Thus, each of up to 64 844 disk drives connected to a single controller can be addressed. Multiunit logical devices are restricted to a set of physical drives having either vertically contiguous or horizontally contiguous unit numbers. With this restriction, to obtain the physical unit number of any unit in a horizontally contiguous multiunit device, add the lowest physical unit number of the drive set to the logical unit number of the unit desired. To obtain the physical unit number of any unit in a vertically contiguous multiunit device, add the lowest physical unit number of the drive set to 10_8 times the logical unit number of the unit desired. Since there is a maximum of eight units in any one horizontally or vertically contiguous group, there is a maximum of eight units per multiunit device. In addition, all units of a multiunit device must be connected to the same channel and, hence, to the same controller.

Figure II-4-2 illustrates a configuration in which two expansion elements and 20 disk drives are connected through one controller. An expansion element with eight drives is connected to port 0, an expansion element with six drives is connected to port 1, and six drives are connected to ports 2 through 7. Each disk drive is shown as a square with its appropriate unit number inside. This configuration is used in the following three examples to illustrate multiunit device assignments. The controller is assumed to be connected to channel 1. Refer to section 4.4.2 for specific information on assigning these devices.

Controller Port

Expander Rank		0	1	2	3	4	5	6	7
	0	00	01	02	03	04	05	06	07
	1	10	11						
	2	20	21						
	3	30	31						
	4	40	41						
	5	50	51						
	6	60							
	7	70							

Figure II-4-2. 844 Expander Configuration with 20 Drives

<u>Example 1:</u>

Figure II-4-3 illustrates a possible configuration for a 3-unit vertically contiguous multi-unit device. This device could be assigned in the CMRDECK, specifying equipment 2 as:

```
EQ2=DI-3, ON, 4, 50, 1.
              ↑   ↑  ↑
              |   |  └──── Channel
              |   └─────── First Unit
              └─────────── Vertically Contiguous Flag
```

The lowest physical unit number of the vertically contiguous device is specified in the unit parameter.

Controller Port

Expander Rank		0	1	2	3	4	5	6	7
	0	00	01	02	03	04	05	06	07
	1	10	11						
	2	20	21						
	3	30	31						
	4	40	41						
	5	50	51						
	6	60							
	7	70							

← Vertically Contiguous 3-Unit Device

Figure II-4-3. Vertically Contiguous 3-Unit Device

Example 2:

Figure II-4-4 illustrates a possible configuration for a 2-unit horizontally contiguous multi-unit device. This device could be assigned in the CMRDECK, specifying equipment 3, as:

EQ3=DI-2, ON, 0, 40, 1.

Channel
First Unit
Horizontally Contiguous Flag

The lowest physical unit number of the horizontally contiguous device is specified in the unit parameter.

Controller Port

Expander Rank	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	10	11						
2	20	21						
3	30	31						
4	40	41	Horizontally Contiguous 2-Unit Device					
5	50	51						
6	60							
7	70							

Figure II-4-4. Horizontally Contiguous 2-Unit Device

Example 3:

Figure II-4-5 illustrates a possible configuration of the 20 disk drives into seven devices. These devices could be assigned in the CMRDECK as:

EQ1=DI-1, ON, 4, 60, 1. (Vertically contiguous)
EQ2=DI-1, ON, 4, 70, 1. (Vertically contiguous)
EQ3=DI-2, ON, 0, 50, 1. (Horizontally contiguous)
EQ4=DI-4, ON, 4, 10, 1. (Vertically contiguous)
EQ5=DI-4, ON, 4, 11, 1. (Vertically contiguous)
EQ6=DI-4, ON, 0, 0, 1. (Horizontally contiguous)
EQ7=DI-4, ON, 0, 4, 1. (Horizontally contiguous)

Equipments 1 and 2 have been defined as vertically contiguous units. This would allow them to be on-line initialized into a vertically contiguous 2-unit device if they were also defined as removable.

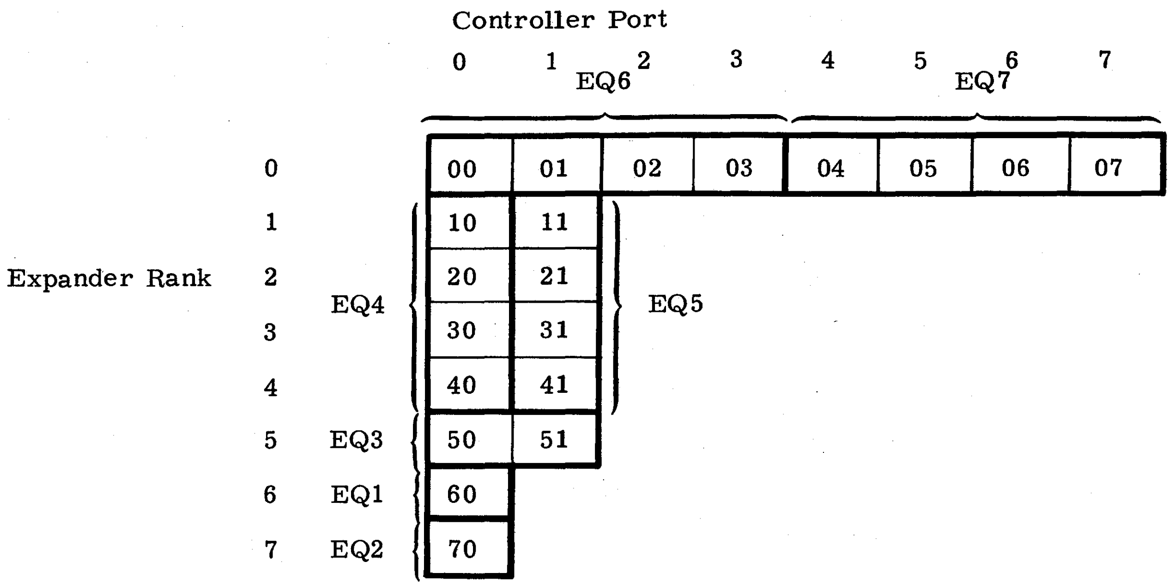


Figure II-4-5. Hardware Configured into Seven Devices

NONREMOVABLE MASS STORAGE DEVICE

This is a device that cannot be physically removed during system operation. It can contain a copy of the system library, which means it is a system device; it can also be available for temporary files. It may or may not contain permanent files.

REMOVABLE MASS STORAGE DEVICE

A removable mass storage device is any mass storage device than can be logically or physically introduced or removed during system operation without causing system malfunction.

A device is specified as removable with the REMOVE entry in the CMRDECK. During deadstart, a removable device is recovered just as any other mass storage device, if the status is ON. If the device is not available, then the status is displayed for the operator (E, M display).

Removable devices can contain permanent files but cannot contain the system library or temporary files because a device containing active files (such as local or library files) cannot be removed from the system. It can be an auxiliary device or an alternate permanent file family device.

SHARED MASS STORAGE DEVICE

A shared mass storage device is a device which contains preserved files that can be accessed by more than one mainframe. To have these preserved files accessable to the mainframe, the device must be defined as shared in the mainframe.

SYSTEM DEVICE

The system device is a nonremovable device on which the system library resides. It may also contain permanent and temporary files.

TEMPORARY FILE DEVICE

The temporary file device is a nonremovable mass storage device on which temporary system files (nonpermanent files) reside. They include:

CM	Common files †
LI	Library files
LO	Local files
SY	System files
TE	Timed/event rollout files
RO	Rollout files

Table II-4-1 summarizes the various functions that a particular mass storage device can serve. For example, if a device is an alternate system device (column on the left), then it cannot be a system device; it can contain temporary files, direct access files, and indirect access files; it can be a master device or a nonmaster device; it cannot be removable; and it can be either an auxiliary device or a family device.

† Only previously locked files can be made common.

TABLE II-4-1. MASS STORAGE DEVICE FUNCTIONS

	Other Possible Functions												
	Alternate system device	System device	Contain temporary files	Contain direct access files	Contain indirect access files	Master device	Nonmaster device	Removable device	Nonre-movable device	Auxiliary device	Family device	Shared device	Link device
System device	no	---	yes	yes	yes	yes	yes	no	yes	yes	yes	yes	yes
Alternate system device	---	no	yes	yes	yes	yes	yes	no	yes	yes	yes	yes	yes
Device containing temporary files	yes	yes	---	yes	yes	yes	yes	no	yes	yes	yes	yes	yes
Device containing direct access files	yes	yes	yes	---	yes	yes	yes	yes	yes	yes	yes	yes	yes
Device containing indirect access files	yes	yes	yes	yes	---	yes	no	yes	yes	yes	yes	yes	yes
Auxiliary device	yes	yes	yes	yes	yes	yes	no	yes	yes	---	no	yes	yes
Default family device	yes	yes	yes	yes	yes	yes	yes	no	yes	no	---	yes	yes
Shared device	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	---	---
Link device	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	---	---

4.4.2 MASS STORAGE EQUIPMENT EST ENTRY

The purpose of the mass storage EQ entries is to describe all mass storage peripheral equipment. NOS requires that at least 6 million words of mass storage be available.

There can be up to 16 logical mass storage devices (up to 16 mass storage EST entries). (This number can be changed at assembly time by modifying the NMSD parameter described in part III, section 2.4.) Each 841 or 844 entry, however, may refer to more than one physical unit. For example, two 844 spindles may be defined as two logical devices with two EQ entries, or they could be defined as one logical device with one EQ entry.

A unit is a dual access unit if it is accessed by one computer system through two different controller-channel access routes. To define a unit as dual access with its EQ entry, specify two channel parameters. Only one controller number entry is necessary because both controllers must have the same number. However, only one channel of a dual-channel access controller is supported for use on a single computer system since using both channels of the controller would result in a performance degradation rather than an improvement. Therefore, if both channel accesses of a controller are physically connected to the same computer system, only one of them may be defined on an EQ entry.

NOTE

> If any of the following entries are to be made for a device, they must follow the device's EQ entry: ASR, TEMP, REMOVE, PF, INITIALIZE, SHARE, LINK. If a device's EQ entry is redefined, then any of those entries must also be redefined.

The format of the entry is:

```
EQord=type, status, controller, unit, chan1,chan2,R.
```

ord — 1 or 2 octal digit EST ordinal of mass storage equipment; 0 through 17_8 [this range depends upon the value of the number of mass storage devices (NMSD), part III, section 2.4.14 which is set when the system is assembled]

When assigning the mass storage equipment to EST ordinals, note the following information concerning system devices. A system device is one which contains a copy of the system library from the deadstart tape. Mass storage devices can be specified as system devices by the following.

With word 14, bits 0 through 5, of deadstart panel

1. When setting the deadstart panel (part II, section 2)
2. When the deadstart options are selected to be displayed (part II, section 3.9)

With the SYSTEM entry

1. When the CMRDECK is selected to be displayed (SYSTEM entry description in section 4.4.6)

Word 14, bits 0 through 5, allow only the mass storage equipment that is assigned to ordinals 0 through 5 to be specified as system devices.

The SYSTEM entry allows any mass storage devices (of the same equipment type) assigned to ordinals 0 through 17_8 to be system devices.

type

2, 3, or 4 digit equipment type; the following mass storage equipment is supported with NOS; n is the number of units connected to a controller.

Equipment	Type	n
3553/841	MD-n	1-8
7054/844-21	DI-n	1-8
7054/844-41	DJ-n	1-8

The 841 and 844 physical units can be defined with each unit being a separate EQ entry, or, if more continuous storage is needed than is possible with one unit, more than one unit of an equipment type can be defined to the system as one logical device with one EQ entry. In this case, the type parameter identifies the number of contiguously numbered units defined with the EQ entry, and the unit parameter identifies the lowest numbered unit of n contiguously numbered units.

For example, two 844 units (0 and 1) to be accessed as two units would be defined as:

EQord=DI-1, status, controller, 0, channel.
EQord=DI-1, status, controller, 1, channel.

The two 844 units (0 and 1) to be accessed as one continuous unit would be defined as:

EQord=DI-2, status, controller, 0, channel.

An advantage to accessing the units as one continuous unit is that less space in CMR is used (630_8 words for the 844-21). A disadvantage is that if either unit malfunctions or is destroyed, both units are affected.

status

Specifies whether or not the equipment is available for access

ON — Available

OFF — The device is not accessed during system operation. Set to the OFF position if the equipment is malfunctioning and access is not desirable.

If the equipment is removable and is not available at deadstart, the system determines that it is unavailable, even if its EQ entry is set to the ON position. Device is not initialized if INITIALIZE is entered until it is set to ON status. During system operation, the operator may initiate access to this device by entering the ON command.

If the equipment can be used with either of two different systems (removable devices, not dual access), define the EQ entry for the system to which it is currently available for access as ON and define the EQ entry for the system to which it is not currently available for access as OFF.

controller	1 or 2 digit controller number (orientation flag for the 7054); 0 through 7

Controller Name	Number	
3553-1	4-7	
7054	0	Specifies horizontally contiguous units
	4	Specifies vertically contiguous units

Refer to Figures II-4-3, II-4-4, and II-4-5 for examples of this parameter for 844 equipment.

If a unit has dual access, then both of its controllers must have the same controller number for both channels.

unit

1 or 2 digit unit number; 0 through 7 (0 through 77_8 for 844 equipment)

If the EQ entry is defining more than one unit of a multispindle device, the unit entry refers to the lowest numbered unit of n units that have contiguous physical unit numbers. Refer to the type parameter description.

When more than one unit of a multispindle device are defined separately as removable devices, there must be enough contiguous physical units to accommodate the largest sized device that is to be used during system operation. For example, if there are four units of an 841 that are to be removable and there is an MD-3 pack, then at least three of these units must have contiguous physical unit numbers. The equipment assignments for these four single-unit devices might be:

```
EQ3=MD-1,OFF,5,2,7.
EQ4=MD-1,OFF,5,3,7.
EQ5=MD-1,OFF,5,4,7.
EQ6=MD-1,OFF,5,5,7.
REMOVE=3,4,5,6.
```

$chan_{1-2}$

1 or 2 octal digit number of channel or channels to which controller is connected; 0 through 13_8, 20 through 33_8

Enter two channel parameters for dual access; the second channel cannot be 0. The two channels must be connected to different controllers.

Equipment	Number of Channels
841 – 3553-1	1 or 2
844-21/41 – 7054	1 or 2

If only one channel parameter is entered, it is the last parameter in the EQ entry; commas are not necessary to indicate that the last two parameters are not used.

R

This optional parameter indicates that the reservation of a controller and disk unit is released when the channel is dropped. For single-access equipment, the reservations are not released when the channel is dropped unless the R option is specified for at least one equipment on that channel. For dual-access equipment, the R option is assumed by the system whether specified or not. It is also assumed for a shared mass storage device operating in a multimainframe environment.

4.4.3 AUTOLOADING OF 7054/844 CONTROLLERS

The 7054/844 controller is a programmable buffer controller and requires that controlware be autoloaded into it before it can be used to control disk drives. The autoloading of this controlware is done automatically at deadstart time unless the AUTOLOAD console entry is made. The format of this entry is:

```
AUTOLOAD.
```

Each occurrence of the AUTOLOAD entry toggles the selection of buffer controller autoloading. This entry is illegal in the CMRDECK and can only be entered from the console at deadstart time. This entry, which affects all 7054/844 controllers defined on the mainframe being deadstarted, is not necessary for normal system operation but is provided as an aid to hardware checkout.

4.4.4 CLEAR 7054 RESERVATIONS ENTRY

If more than one system can access a group of 844 devices, the possibility exists of the first system going down, leaving channel reservations on some or all of these devices. In order for the second system to access devices in this condition, a GRENADE entry must be made. The format of this entry is:

```
GRENADE.
```

This entry indicates that a grenade function (11) will be issued to all 7054 controllers, once the controlware is loaded. This function causes channel reservations to be cleared on all 844 units physically connected to each 7054 controller. Use this command with caution since it can interrupt the operation of another machine that could be accessing affected units via another controller. Each occurrence of the GRENADE. entry toggles the selection of the grenade function. This entry is illegal in the CMRDECK and can only be entered from the console at deadstart time.

4.4.5 ECS EQUIPMENT EST ENTRY

There are two possible ECS configurations. The coupler is part of the mainframe and allows a CPU (or two if dual CPUs are available) to communicate with ECS. The distributive data path (DDP) is standard with a CDC CYBER system and optional with a 6000 system. It allows any PPU in the system as well as the CPU directly to communicate with ECS. NOS supports up to four DDPs.

The format of the entry is:

EQord=type, status, size, $chan_1$, $chan_2$.

ord — 1 or 2 octal digit EST ordinal of ECS; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14).

type — ECS equipment type

DE	DDP is not available
DP	DDP is available

status — Specifies whether or not ECS is available for access

ON	Available
OFF	Equipment is ignored during system operation

size — One of following octal digit numbers representing size of ECS unit

Size	ECS Available	Number of Banks
400	125K	1
1000	250K	2
2000	500K	4
4000	1000K	8
10000	2000K	16

The track CETK defined in COMSESS is always logically flawed to allow a fixed area for testing by DDS.

$chan_{1-2}$ — 1 or 2 octal digit number of channel or channels to which the DDP is connected; 0 through 13_8, 20 through 33_8

If the equipment type is DE, do not specify a channel parameter. If a channel parameter is included in a DE entry, the system recognizes it as a DP entry.

If the equipment type is DP, specify either one or two channels. The second channel cannot be 0. If a DDP is present, program loads of CPU programs residing in ECS will still occur via the CPU. A DDP must be connected to a channel by itself.

Example:

```
EQ4=DE,OFF,1000
EQ5=DP,ON,2000,2,3
```

NOTE

If ECS is not included in the hardware configuration, an ECS equipment EST entry must not be made.

4.4.6 TEMP: TEMPORARY FILES DEVICE ASSIGNMENT

To assign a mass storage device (nonremovable device only) as available for temporary files, add a TEMP entry for that device to the CMRDECK. Do not add a TEMP entry for a device with an EQ entry set to OFF or which already has a REMOVE assignment. Temporary files include:

CM	Common files †
LI	Library files
LO	Local files
SY	System files
TE	Timed/event rollout files
RO	Rollout files

TEMP=ord,...,ord.

ord — 1 or 2 octal digit EST ordinal of mass storage device; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14); one or more ordinals may be specified with one entry.

† Only previously locked files can be made common.

4.4.7 PF: PERMANENT FILES DEVICE ASSIGNMENT

If a mass storage unit is to be defined for permanent files, enter a PF entry for that unit in the CMRDECK anywhere after the unit's EQ entry.

The PF entry information becomes part of the unit's label when the unit is initialized; this label is always recovered during subsequent deadstarts. In this case, it is not necessary that the PF entry be part of the CMRDECK on the configured deadstart tape; if it is, it is ignored. However, if the unit is to be initialized (with the INITIALIZE entry in CMRDECK), the system requires a PF entry in order to create the label.

If the unit is a family device, the format is †:

```
PF=ord, F, dm, sm, name, device, nc.
```

If the unit is an auxiliary device, the format is †:

```
PF=ord, X, name, nc.
```

ord	1 or 2 octal digit EST ordinal of mass storage device; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14)
F, X	This parameter specifies the type of device the unit is to be F — Unit is a family device. It can contain indirect access files if the dm parameter is set to 1 through 377_8. It can contain direct access files if the sm parameter is set to 1 through 377_8. It is a master device if the dm parameter is set. X — Unit is an auxiliary device and can contain both direct and indirect access files. X must be specified on a unit's PF entry if any of the auxiliary device commands are to be used for the device. Refer to Table II-4-2 for dependencies.
dm	1, 2, or 3 octal digit parameter specifying the unit's device mask; range is 0 through 377_8; omit if auxiliary device. Set according to information in part IV, section 2.1. The device mask for a permanent file device defines the groups of users whose catalogs reside on the device for a particular family.
sm	1, 2, or 3 octal digit parameter specifying the unit's secondary mask; range is 0 through 377_8; omit if auxiliary device. This parameter controls the residence of direct access files in the same manner that dm controls the residence of indirect access files. Set according to information in part IV, section 2.1.

†The PF entry is illegal if the equipment is specified as OFF.

name

1 through 7 alphanumeric character parameter designating either name of family to which unit belongs or its pack name if it is an auxiliary device.

The family name describes the permanent file devices available to a user.† A family may consist of 1 through 63 logical devices; however, the master devices within the family must have user masks totaling 377_8 if all possible user indices are to be accommodated. Usually a system runs with one family of permanent file devices available. But additional families can be activated on a system in order to allow the users of these additional families to access their permanent files through an alternate system. This might be helpful if one system were to supply backup service to another system.

The pack name for a device specifies the unique 7-character name associated with an auxiliary device. An auxiliary device is a self-contained permanent file device which means that all files (direct and indirect access) represented by the catalogs on the device reside on that device. To access a file from an auxiliary device, the pack name must be specified as part of the permanent file request. The pack name is used instead of the usual algorithm for determining catalog location (user masks and family name). Any user who knows the pack name and has appropriate permissions and validations may access files on an auxiliary device.

device

1 or 2 octal digit number of device in family; must be a unique number in family (1 through 77_8); omit if auxiliary device

All permanent files that exist on devices other than the master device (where the catalog entry resides) will have a device number in the catalog entry indicating which alternate device within the family contains the file. Auxiliary devices do not have device numbers.

nc

Optional parameter specifying the number of catalog tracks; range is 1 through 200_8. This value must be a power of 2. If nc is not specified, a default value based on the equipment type is supplied.

The default values for each mass storage equipment type are:

Device	Equipment Type	Default Number Catalog Tracks
841	MD	20
844-21	DI	20
844-41	DJ	20
ECS	DE	2
ECS with DDP	DP	2
Private device		1

Examples:

```
PF=2,F,125,125,SYSTEM,3,200.
PF=17,X,PACK.
```

† If not otherwise specified, the default family name becomes part of the tape label information. It is checked and verified if the user specifies the FA=A parameter on a control statement. Refer to volume 1 of the system reference manual for a discussion of this parameter.

TABLE II-4-2. PF ENTRY

Type of PF Device	Files Permitted on Device	PF Entry Parameter Settings				
		type	dm	name	device	sm
Auxiliary †	Indirect and/or direct	X	omit	pack	omit	omit
Family	Direct only	F	0	family	$1\text{-}77_8$	$1\text{-}377_8$
	Indirect only (master device)	F	$1\text{-}377_8$	family	$1\text{-}77_8$	0
	Indirect and direct (master device)	F	$1\text{-}377_8$	family	$1\text{-}77_8$	$1\text{-}377_8$

†If a user number is specified for an auxiliary device, use the INITIALIZE command after deadstart. Refer to the operator's guide for procedure.

4.4.8 KRONOS 2.1.1 PERMANENT FILE CONVERSION TO NOS VERSION 1

Permanent files created under KRONOS systems that are updated through PSR level 397 may be recovered by NOS systems.

If the permanent file devices are reconfigured under KRONOS or NOS, an installation can alternate between NOS and KRONOS systems without reloading permanent files.

Only the identical EQ entries are necessary in the NOS CMRDECK to recover permanent files created with KRONOS systems. The PF entry is not required to recover KRONOS 2.1.1 permanent files; however, the PF entries are identified in the following example to indicate what the PF entry characteristics are for NOS.

To determine the permanent file definitions needed for the KRONOS 2.1.1 CMRDECK, use the following information.

The KRONOS 2.1.1 PF entry format was:

```
PF=ord, type, mask, name, device.
```

The NOS PF entry format is:

```
PF=ord, type, dm, sm, name, device, nc.
```

type — To use the permanent files created under a KRONOS system with a NOS system, convert the parameters as follows:

KRONOS Parameter	NOS Parameter
D	F and sm=377
I	F and sm=0
X	X and omit sm

mask — The KRONOS mask parameter is the same as the NOS dm parameter.

name — Family name

device — Device number

All KRONOS devices which did not have a name or device number must be initialized under NOS.

Example:

If the KRONOS CMRDECK is:

```
EQ2=MD-4, ON, 4, 0, 2.
EQ3=MD-2, ON, 4, 4, 2.
TEMP=0, 1, 3.
PF=2, I, 377, , 41.
PF=3, D, , , 42.
```

the NOS CMRDECK is:

```
EQ2=MD-4, ON, 4, 0, 2.
EQ3=MD-2, ON, 4, 4, 2.
TEMP=0, 1, 3.
PF=2, F, 377, , , 41.
PF=3, F, , 377, , 42.
```

All dayfiles, recovered or newly created, are made preserved files. The pointers to these dayfiles are written as part of the checkpoint information of the device along with the machine id with which the machine is running. A maximum of 31 unique ids can be maintained on the device.

If such a device is then recovered on a system not updated through PSR level 419 only the pointers written as checkpoint information by the last machine in the multimainframe environment to checkpoint are retained; all other pointers are discarded. The preserved files (associated with the discarded pointers) are left on the device, but access is impossible. The space is dead space until the device is initialized. In recovering such a device, an additional comparison is made between the id of the recovering machine and the id from the label. If they match, recovery proceeds; if they do not match, operator intervention is necessary to proceed, in which case the dayfiles and pointers are discarded.

The inactive queue file table (IQFT) file, recovered or newly created, is made a preserved file. Queued files that were created on systems not updated through PSR level 419 do not have a machine id in their system sectors. QREC processing uses the id parameter as part of its selection criteria. To process queued files that have no id, the QREC id parameter must specifically be cleared. (The IQFT file and QREC processing are discussed in detail in the operator's guide.)

4.4.9 SYSTEM: SYSTEM LIBRARY DEVICE ASSIGNMENT

The SYSTEM entry specifies which mass storage devices are to contain copies of the NOS system library. A system device can be any mass storage device as well as ECS.

Throughput can be greatly improved by specifying more than one system device. For example, if two system devices are specified and they are on different channels, the time required to access system programs can be reduced by as much as one half. When the channel for one system device is busy, the other is accessed.

Note the following restrictions.

- The SYSTEM entry can be typed only at deadstart when the CMRDECK is displayed; it cannot be included in a CMRDECK on the deadstart tape.
- The EQ entry for a system device cannot have the status set to OFF.
- There cannot be a REMOVE entry in the CMRDECK for a device to be specified as a SYSTEM device.
- If more than one device is specified as a system device, all devices specified must be of the same type (such as all MD-n, with n the same for all devices).
- If no devices are specified as system devices, the system library resides on the mass storage device defined by EST ordinal 0.
- If an ASR entry is made for a device with a SYSTEM entry, the ASR entry is ignored.

These devices can also be specified in word 14, bits 0 through 5, either at the deadstart panel (part II, section 2.4) or when the deadstart options are displayed (part II, section 3.11). Word 14 restricts the number of devices that can be specified as system devices to six (those defined in EST ordinals 0 through 5). The SYSTEM entry supersedes the word 14 settings.

The format is:

```
SYSTEM=ord,...,ord.
```

ord — EST ordinal of the unit to contain a copy of the system library on the deadstart tape; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14); the EQ entry must be set to ON status. One or more ordinals may be specified with one SYSTEM entry.

4.4.10 ASR: ALTERNATE SYSTEM LIBRARY DEVICE ASSIGNMENT

This entry specifies which mass storage devices are to be alternate system devices. An alternate system device is a mass storage device other than a system device on which duplicate copies of ABS, OVL, PP, and REL type routines can be placed by the system for faster access than is possible from a system device or because they are frequently used programs.

The procedure for selecting the routines to be placed on the alternate device is in part III, section 2.6.

The format is:

```
ASR=ord,...,ord.
```

ord — 1 or 2 octal digit EST ordinal of mass storage device to be used as an alternate system device, 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14). Note the following restrictions:

1. Device must be a mass storage device, including ECS.
2. Device cannot be a removable device.
3. Device cannot be a system device. If a SYSTEM entry is made for the same device after an alternate device entry, the SYSTEM entry supersedes the ASR entry.

4.4.11 FAMILY: FAMILY NAME ASSIGNMENT

If there is only one permanent file family in the active system and the family name is the same as the name on EQ0, it is not necessary to specify a family name. However, if the names differ, a FAMILY entry is needed to access permanent files. When more than one family of permanent file devices are active in a system, the user must identify the family to which he belongs whenever he logs in or initiates a job. If he does not, the system assumes the default family.

The FAMILY entry in the CMRDECK defines the default family. It must follow the EQ entry for the permanent file device.

Note the following restrictions.

1. The status parameter for a system device's EQ entry cannot be set to OFF.
2. There cannot be a REMOVE entry in the CMRDECK for a device to be specified as a FAMILY device.

```
FAMILY=ord.
```

ord	1 or 2 digit EST ordinal number of the mass storage device that the system automatically uses to determine the user's family when the user does not specify a family name at login or job initiation

4.4.12 REMOVE: REMOVABLE DEVICE ASSIGNMENT

If a mass storage device is to be considered removable, it must be specified as such at deadstart with the REMOVE entry. This allows it to be introduced or removed during system operation.

Note the following restrictions.

1. A device specified as removable cannot also have an ASR, SYSTEM, TEMP, LINK, FAMILY, DAYFILE, ACCOUNT, or ERRLOG entry associated with it.
2. A device assigned to EST ordinal 0 cannot be specified as removable.

```
REMOVE=ord,...,ord.
```

ord	1 or 2 octal digit EST ordinal of mass storage device that is removable; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14); one or more ordinals may be specified with one REMOVE entry.

4.4.13 INITIALIZE: INITIALIZATION ENTRY

In order to use a mass storage device that is defined with an EQ entry, it must have a label. A label is written on a device when it is initialized, using either the INITIALIZE entry in the CMRDECK at deadstart time or the INITIALIZE command during system operation (refer to the operator's guide).

A mass storage device's label is contained on a logical track (usually track 0). It contains information about the allocation and characteristics of a device (and its units, if there is more than one unit on a device). This information is in the form of a label sector for the first unit, a TRT for the device, and a label sector for each unit.

Initialization does not automatically occur at each deadstart because mass storage device labels are recovered during all deadstarts. Therefore, initialize a device only in the following situations.

1. To add a new mass storage device (no label exists on the device) use the INITIALIZE entry.
2. If parts of the label on a permanent file device have been destroyed by maintenance operations (permanent files having been dumped to another device before diagnostics were run), use the INITIALIZE entry during deadstart to write a new label. Then reload the permanent files.
3. If a device (usually auxiliary or alternate permanent file family device) is added to a system during operation, use the INITIALIZE command (refer to the operator's guide) to initialize it if it does not have a valid label on it when it is added to the system.

The INITIALIZE entry has the following characteristics.

1. It can only be entered at the system console when the CMRDECK is displayed during a level 0 deadstart. It can be entered anywhere after the EQ entry for the device.

   If it is placed in the deadstart tape CMRDECK, the system issues the error message ILLEGAL ENTRY when the CMRDECK is read from the tape.

2. A total initialize (op=AL) assumes that no valuable information exists on the device and creates a new label. When the new label is created, all previously existing information on the device, including permanent files, is lost.

3. If the EQ status for the device is OFF when INITIALIZE is entered, initialization of the device occurs whenever the device is set to ON status using the DSD ON operator command during normal system operation.

4. If the device is not a master device, INITIALIZE (op=AL) only writes a label; if it is a master device, then it also initializes the catalog track and writes EOIs at the beginning of the permit track, the indirect access track (data chain), and each catalog track.

5. During a deadstart initialization (op=AL), all flaw reservations specified for a device are lost and must be reentered, except for 844 type devices with factory-formatted disk packs. Refer to part II, section 4.5.

6. Always use the INITIALIZE CMRDECK entry to specify a user number for a private auxiliary device.

The entry format is:

```
INITIALIZE, ord, op.
```

ord — 1 or 2 octal digit EST ordinal of mass storage device to be initialized; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14)

If the ordinal refers to a family permanent file device, then family name, device number, and mask (if it is a master device) are specified on the PF entry.

If it is an auxiliary device, the pack name is specified on the PF entry.

op — Level of initialization. The following values are acceptable.

AL	Total initialize
PF	Initialize permanent files
QF	Initialize queued files
DF	Initialize system dayfile
AF	Initialize account dayfile
EF	Initialize error log dayfile
FP	Initialize format pack (An automatic selection of AL also occurs)

The total initialization (op=AL or FP) is the only initialization that is independent of the content of the pack, if the initialization occurs during deadstart. If the initialization is an on-line initialization (refer to the operator's guide), initialization is applied to the device after CMS has recovered it. If CMS cannot recover the device, the initialization is similar to a deadstart initialization (that is, all information on the device is lost).

The device number, family name, and device masks may only be changed during a total initialization. Since all devices may contain permanent files, a PF entry should be included for a device when performing a total initialization. If a PF entry is not included with the INITIALIZE entry, the device is assigned a default family name, device number, and device masks. It is possible that these parameters may conflict with other devices in the system. If a conflict occurs, it must be resolved through the use of PF entries. If no PF entry is specified when a device is initialized, the default family name is SYSTEM, and the default device numbers begin at 1 and increase by 1 for each device that is initialized without a PF entry. If equipment 0 is initialized without a PF entry, the device mask and secondary mask are set to 377_8. For all other equipment, the default masks are set to 0.

The INITIALIZE entry operates in conjunction with the dayfile entries DAYFILE, ACCOUNT, and ERRLOG (refer to section 4.2) to determine where the dayfiles actually reside. The following examples illustrate the various cases. Assume that the system has three mass storage devices (equipment 0, 1, and 2).

Example 1:

For this example, no dayfile entries are made, and no previous dayfiles exist.

The following CMRDECK entries are made.

```
INITIALIZE, 0, AL.
INITIALIZE, 1, AL.
INITIALIZE, 2, AL.
```

In this case, no dayfile entries are made. All dayfiles reside on equipment 0.

Example 2:

In this example, dayfile entries are made, but no previous dayfiles exist.

The following CMRDECK entries are made.

```
DAYFILE=0, 200.
ACCOUNT=1, 200.
ERRLOG=2.
INITIALIZE, 0, AL.
INITIALIZE, 1, AL.
INITIALIZE, 2, AL.
```

In this case, the dayfiles reside on the indicated devices (dayfile on equipment 0, account file on equipment 1, and errlog on equipment 2). The default buffer length is used for the error log dayfile buffer.

Example 3:

In this example, dayfile entries are made, and previous dayfiles do exist.

Assume that the CMRDECK entries in example 2 are used.

Since each device has been subjected to a total initialization, no dayfiles are recovered. They reside on the indicated devices.

Example 4:

In this example, dayfile entries are made, previous dayfiles exist, but no dayfile initialization entries are made.

The following CMRDECK entries are made.

```
DAYFILE=0.
ACCOUNT=1.
ERRLOG=2.
INITIALIZE, 0, PF.
```

Two possibilities exist: the dayfiles may already reside on the specified devices, or they may reside on some permutation of the possible devices. In either case, since no dayfile initialization entries are made, the old dayfiles are recovered. The residence of these dayfiles is governed by the residence of the old dayfiles. The PF initialization entry returns all permanent file space and relabels the device based on the recovered device parameters. The dayfiles and queued files on this device are not affected by this entry.

Example 5:

In this example, dayfile entries are made, previous dayfiles exist, no dayfile initialization entries are made, and duplicate dayfiles are in existence.

Assume that the CMRDECK entries in example 4 are used.

For the dayfiles that do not have duplicates, the residence is defined by the current residence of the files, not the CMRDECK entries. But assume that an error log dayfile is recovered from equipments 0 and 2. In this case, the most recent file becomes the active error log. Its previous residence overrides the CMRDECK entry. The other file becomes an inactive dayfile (an entry exists in the mass storage table of the device pointing to the inactive file, but the file is not in use by the system).

To produce an inactive error log dayfile, the site must run in the following manner.

1. Assume an 844 Disk Subsystem with two or more spindles is being used. Run with unit 1 equated to EQ0 and unit 0 unused.

2. Redeadstart, equate unit 0 to EQ0, and do not use unit 1.

3. Redeadstart, equate unit 0 to EQ0, and unit 1 to EQ1.

Since unit 0 has the most recent copy of the error log dayfile, this copy would become an active error log dayfile and the copy on unit 1 would become an inactive error log dayfile.

Example 6:

In this example, dayfile entries are made, the previous dayfiles from example 2 exist, and initialization entries are made.

The following CMRDECK entries are made.

```
DAYFILE=1.
ACCOUNT=1.
ERRLOG=2, 300.
INITIALIZE, 0, DF.
INITIALIZE, 0, QF.
```

In this case, the account dayfile is recovered and continued on equipment 1. The error log dayfile is recovered and continued on equipment 2 with a CM buffer of 300_8 words. The dayfile space on equipment 0 (from example 2) is released and the new dayfile starts on equipment 1. The QF initialization entry releases all space reserved by queued files on equipment 0.

The CM buffer length is not affected by dayfile recovery. It is always specified by the values defined in the CMRDECK entries. If no buffer length entries exist, the system default values are used.

4.4.14 SHARE: SHARED DEVICE DESIGNATION

This entry identifies mass storage devices which are to be shared between two or more machines of a multimainframe (MMF) community (machines sharing mass storage). The MST, TRT, and other information relative to the management of these devices are maintained on the link device (ECS) for rapid access and interlock purposes. The link device is considered a shared device, but it is not necessary to include it in the SHARE entry. The following table lists equipment types which may be shared.

Equipment Type	Device Mnemonic
ECS	DE
7054/844-21	DI
7054/844-41	DJ
DDP	DP
3553/841	MD

Presence of the SHARE entry implies an MMF configuration, and as such, a LINK entry is required (refer to section 4.4.15).

SHARE=ord_1, ord_2, . . ., ord_n.

ord_i — 1 or 2 octal digit EST ordinal of the mass storage device being shared; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14). One or more ordinals may be specified with each entry.

4.4.15 LINK: LINK DEVICE DECLARATION

This entry indicates deadstart into an MMF configuration and specifies the equipment number of the link device. The link device must be ECS, identified as either DE or DP and cannot be defined as removable. It is assigned shared device status automatically.

```
LINK=ord.
```

ord — 1 or 2 digit EST ordinal of the ECS entry; 0 is not a legal value for ord.

4.4.16 PRESET: PRESET MMF LINK DEVICE

This entry causes initialization of tables on the link device in preparation for the creation of an MMF environment. It is valid only from the system console and used only by the first machine deadstarting. Once PRESET is issued, the SHARE command is disabled, since the layout of ECS resident is dependent upon the number of shared devices. Therefore, all SHARE entries must precede the PRESET entry. The following two forms of this entry are available.

```
PRESET.
or
PRESET,n.
```

If n is not specified, the link device is preset and allocated according to the number of shared devices specified. If n is specified, the link device is preset and allocated for n shared devices, in the range:

shared device count $\leq$ n < 77B

The PRESET entry applies only for level 0 deadstarts.

A maximum of 4 x (NMSD-1), where NMSD is an assembly constant with release value of 30B, unique mass storage devices (shared and nonshared) can be defined by the whole MMF environment. (NMSD is described in part III, section 2.4.14.) This is constrained by the device access table (DAT) size.

The second form of the PRESET entry should be used when the total number of shared devices in the MMF environment is greater than the number of shared devices defined by the machine being preset. The total number is the parameter supplied in the entry.

The PRESET command is required if the machine is the first machine being brought up in an MMF environment on a level 0 deadstart.

4.5 TRACK RESERVATIONS

These entries identify areas of mass storage that are unusable (flawed areas) and prevent the system from accessing them. The information in the entries is used by the system to build the TRT for each device which resides in CMR and also in the mass storage device label.

Flaws can be entered at three different times.

- During deadstart when the CMRDECK is displayed
- During system operation using the FLAW entry that is defined in the operator's guide
- When configuring a deadstart tape

The formats described in this section are those for entering flaws during deadstart or on the deadstart tape.

- Use the CTK entry to clear all reservations on a device.
- Use the RTK entry to specify the physical address of a flaw. (If that reservation already exists, it remains in effect.)
- Use the TTK entry to cancel a particular RTK entry. (If that reservation does not exist, TTK makes that reservation instead of cancelling.)
- Use the STK entry to specify the logical address of a flaw. (If that reservation already exists, it remains in effect.)

Obtain flaw addresses from the customer engineer or the system analyst, or run the MST (mass storage test) on the device to determine the bad areas. The MST specifies the physical address of flaws.

The total number of reservations for all mass storage devices in the system cannot exceed 255_8.

The flaw information recorded on the utility flaw map of an 881/883 disk pack is read during the initialization of 844 equipment, and the system reserves the appropriate areas. This automatic flawing occurs in addition to any CTK, RTK, TTK, or STK entries. However, areas recorded as flawed on the 881/883 pack cannot be cleared using the TTK entry. Refer to part IV, section 7 for information on clearing these flaws.

The track CETK defined in COMSESS for ECS is always logically flawed to allow a fixed area for ECS testing by DDS.

4.5.1 CTK: CLEAR DEVICE TRACK RESERVATIONS

The CTK entry clears all flaw reservations that were previously made with RTK, STK, or TTK entries. This is the only way to cancel reservations made with STK entries. Reservations made with an RTK or a TTK entry can be cancelled with a duplicate TTK entry as well as with a CTK entry.

The difference is that a CTK entry cancels all track reservations on a device, whereas a TTK cancels individual physical track reservations.

Entering EQord=0 (part II, section 4.3.1) does not clear flaw entries for that equipment.

```
CTK=ord,...,ord.
```

ord	1 or 2 octal digit EST ordinal of mass storage device; 0 through 17_8 (this range depends on the value of NMSD, part III, section 2.4.14); one or more ordinals may be specified with one CTK entry.

4.5.2 TTK: CANCEL PHYSICAL ADDRESS ENTRIES

To cancel a flaw made with an RTK entry, enter the identical information with a TTK entry. If, however, the flaw did not exist before the TTK is entered, the area specified is reserved in the same way as with an RTK entry. A TTK entry can be cancelled with a duplicate TTK entry.

```
TTK=ord,address.
```

ord	1 or 2 octal digit EST ordinal of device; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14)
address	Physical address of mass storage area to be shown by the RTK entry descriptions that follows.

4.5.3 STK: RESERVE LOGICAL AREAS ON ANY MASS STORAGE DEVICE

Use this entry to specify the logical address of a flaw. If the track was reserved previously, that reservation remains in effect.

```
STK=ord,track.
```

ord	1 or 2 octal digit EST ordinal; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14).
track	4 octal digit logical track to be reserved; 4000 through 7777_8

4.5.4 ECS RESERVATION ENTRY

Use this entry to prevent the system from using blocks (tracks) of ECS.†

```
RTK=ord,Aaddress.
```

ord	1 or 2 octal digit EST ordinal of ECS; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14).
address	1 through 6 octal digit logical address in a track of ECS; track containing absolute address is reserved

Example:

RTK=4,A714140.

4.5.5 841 MULTIPLE DISK DRIVE RESERVATION ENTRY

Use the following entry to reserve areas of an 841 multiple disk drive.

```
RTK=ord,Ccylinder,Ssector.
```

ord	1 or 2 octal digit 841 EST ordinal; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14).
cylinder	1 through 3 octal digit cylinder number; 0 through 307_8
sector	1 through 4 octal digit sector number; 0 through 2315_8

4.5.6 844-21 DISK STORAGE SUBSYSTEM RESERVATION ENTRY

Use the following entry to reserve areas of an 844-21 disk.

```
RTK=ord,Ccccc,Sttss.
```

ord	1 or 2 octal digit 844-21 EST ordinal; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14).
cccc	1 through 4 octal digit cylinder number; 0 through 627_8
ttss	1 through 4 octal digit track/sector number; 0 through 2127_8. The upper two digits specify the physical track; 0 through 21_8. The lower two digits specify the physical sector, 0 through 27_8.

Flaw entries should not be specified for areas that already appear in the flaw information recorded on the disk pack. Refer to part IV, section 7 for a further description of this flaw information.

† The track CETK defined in COMSESS is always set to a logical flaw when ECS is initialized, independent of any other track flaw operation. If an attempt is made to change the track status, an ILLEGAL ENTRY error is reported.

4.5.7 844-41 DISK STORAGE SUBSYSTEM RESERVATION ENTRY

Use the following entry to reserve areas of an 844-41 disk.

```
RTK=ord, Ccccc, Sttss.
```

ord	1 or 2 octal digit 844-41 EST ordinal; 0 through 17_8 (this range depends upon the value of NMSD, part III, section 2.4.14).
cccc	1 through 4 octal digit cylinder number; 0 through 1463_8
ttss	1 through 4 octal digit track/sector number; 0 through 2227_8. The upper two digits specify the physical track, 0 through 22_8. The lower two digits specify the physical sector, 0 through 27_8.

Flaw entries should not be specified for areas that already appear in the utility flaw map recorded on the disk pack. Refer to part IV, section 7 for a further description of this flaw information.

IPRDECK 5

5.1 GENERAL DESCRIPTION

Two displays pertaining to the IPRDECK can be displayed atlernately by pressing the right blank key. The first is the IPRINST display. It lists the valid IPRDECK entries that are described throughout this section. The second display is the current IPRDECK. The IPRDECK contains system installation parameters that describe the mode of system operation. There can be up to 4096_{10} IPRDECKS on a deadstart tape. If there are more than one on the tape, specify the deck to be used during a particular deadstart with the IPD=entry in the CMRDECK (part II, section 4.2). If either the IPRDECK or the IPRINST overflow two screens, the display can be advanced by pressing the + key.

IPRDECK modification is accomplished by entering the appropriate changes or additions from the console keyboard. These entries can be made when either IPRINST or IPRDECK is being displayed. Each console entry supersedes the value currently specified in the IPRDECK. However, the modified IPRDECK remains in effect only until the next level 0 deadstart is performed. That is, changes to the IPRDECK are not recovered across deadstart unless a new deadstart tape is created to reflect those changes (refer to part III, section 2.5).

Most of the IPRDECK entries are also valid DSD commands that can be used to make the same changes during system operation. These DSD assignments are not retained on any level of recovery deadstart.

<u>IPRINST Released Format:</u>

```
IPRINST
INSTALLATION PARAMETER ENTRIES.

CERTAIN INSTALLATION PARAMETERS REFER TO THE JOB ORIGIN
TYPE *OT*.  *OT* MAY BE REPLACED BY THE FOLLOWING -
     SY   SYSTEM        BC   BATCH
     EI   E/I 200       TX   TELEX

ACCOUNT.   TOGGLE ACCOUNT CARD IGNORE SWITCH.
AUTOROLL.  TOGGLE AUTO ROLL DISABLE.
DEBUG.     TOGGLE DEBUG SWITCH.
ENGR.      TOGGLE ENGINEERING MODE.
LOCK.      TOGGLE CONSOLE LOCK STATUS.
PRIORITY.  TOGGLE PRIORITY EVALUATION DISABLE.
USERS.     TOGGLE SECONDARY USER CARDS DISABLE.
VALID.     TOGGLE JOB VALIDATION DISABLE.
BATCHIO.   TOGGLE AUTO STATUS OF *BATCHIO*.
EI200.     TOGGLE AUTO STATUS OF *EI200*.
MAGNET.    TOGGLE AUTO STATUS OF *MAGNET*.
REMOVABLE PACKS.   TOGGLE REMOVABLE PACK CHECKING.
TELEX.     TOGGLE AUTO STATUS OF *TELEX*.
TRANEX.    TOGGLE AUTO STATUS OF *TRANEX*.
QPROTECT.  TOGGLE QPROTECT ENABLE.

DELAY,T1XXX,T2XXX,...,TNXXX.
           SET DELAY TIME *TN* = XXX.
     TN =  AR      AUTO RECALL (MILLISECONDS)
           CS      CPU JOB SWITCH (MILLISECONDS)
           CR      CPU PROGRAM RECALL (MILLISECONDS)
           JA      JOB ADVANCE (MILLISECONDS)
           JS      JOB SCHEDULER (SECONDS)
           TC      THRESHOLD COUNT (NUMBER SINGLE SECDED ERRORS)
```

```
QUEUE,OT,QT,Q1XXX,Q2XXX,...,QNXXX.
          SET QUEUE PARAMETERS *QN* = XXXX, FOR QUEUE *QT* OF
          JOB ORIGIN *OT*.
     QT = IN   INPUT     QN = OP   ORIGINAL PRIORITY
          RO   ROLLOUT        LP   LOWER BOUND FOR PRIORITY AGE
          OT   OUTPUT         UP   UPPER BOUND FOR PRIORITY AGE
                              IN   AGE INCREMENT

CSM=CC.
          SET SYSTEM CHARACTER SET MODE.
     CC = 63 - 63 CHARACTER SET      64 - 64 CHARACTER SET
TCVM=CC.
          SET ASSUMED MAGNETIC TAPE CONVERSION MODE.
     CC = AS   ASCII     EB   EBCDIC     US   USASI

TDEN=CC.
          SET ASSUMED TAPE DENSITY ACCORDING TO CC.
     CC = LO   200     HI   556     HY   800
          HD   800     PE   1600

TDTR=CC.
          SET ASSUMED MAGNETIC TAPE TYPE ACCORDING TO CC.
     CC = MT   SEVEN TRACK   NT   NINE TRACK

DSD,X,CCC-CCC
          SET INITIAL KEYBOARD COMMAND = CCC-CCC IF
          RECOVERY MODE X IS SELECTED.
          ONLY THOSE CHARACTERS THAT MAKE THE ENTRY UNIQUE
          ARE REQUIRED.

SERVICE,OT,P1XXX,P2XXX,...,PNXXX.
          SET JOB SERVICE PARAMETERS *PN* = XXXX, FOR JOB
          ORIGIN *OT*.
     PN = PR      INITIAL CPU PRIORITY
          CP      CPU TIME SLICE (MILLISECONDS*64)
          CM      CENTRAL MEMORY TIME SLICE (SECONDS)
          NJ      MAXIMUM NUMBER OF JOBS
          FL      MAXIMUM FIELD LENGTH FOR ANY JOB
          AM      MAXIMUM FIELD LENGTH FOR ALL JOBS
          FC      NUMBER OF FILES IN CATALOG
          FS      INDIVIDUAL INDIRECT ACCESS FILE SIZE
          CS      CUMULATIVE INDIRECT ACCESS FILE SIZE
          DS      INDIVIDUAL DIRECT ACCESS FILE SIZE

MORE - TYPE + FOR NEXT PAGE
MSAL,F1XX,F2XX,...FNXX.
          ASSIGN JOB FILE TYPE *FN* TO MASS STORAGE EQ XX.
     FN = IN   INPUT     LO   LOCAL(SCRATCH)     LG   LGO
          OT   OUTPUT    RO   ROLLOUT

CPM,N1=XX1,N2=XX2.
          N = 0 OR 1          XX = 1 - 6
```

Format of a Released IPRDECK:

```
IPRDECK
TDEN=HY.
CSM=64.
LOCK.
EI200.
TELEX.
TRANEX.
QUEUE,SY,IN,OP6600,LP700,UP3000.
QUEUE,SY,RO,OP6000,LP100,UP1000.
QUEUE,SY,OT,OP400,LP100,UP7700.
SERVICE,SY,PR1,CP100,CM20.
QUEUE,BC,IN,OP2400,LP2000,UP4010.
QUEUE,BC,RO,OP2400,LP1010,UP4004.
QUEUE,BC,OT,OP200,LP100,UP7000.
SERVICE,BC,PR30,CP400,CM200.
QUEUE,EI,IN,OP3400,LP2400,UP4010.
QUEUE,EI,RO,OP3400,LP1400,UP4006.
QUEUE,EI,OT,OP200,LP100,UP7600.
SERVICE,EI,PR30,CP400,CM200.
QUEUE,TX,IN,OP4000,LP3770,UP7006.
QUEUE,TX,RO,OP4004,LP3740,UP7000.
QUEUE,TX,OT,OP200,LP100,UP7000.
SERVICE,TX,PR30,CP40,CM10.
QUEUE,MT,IN,OP6774,LP6700,UP7400.
QUEUE,MT,RO,OP6774,LP4000,UP7400.
QUEUE,MT,OT,OP6000,LP100,UP7700.
SERVICE,MT,PR31,CP400,CM60.
DELAY,JS1,CS10,AR1000.
DSD,0,MAI%X.ISF.%X.QREC.
DSD,3,AUTO.
```

5.2 ACCOUNT.

Format	Default	Significance
ACCOUNT.	enabled	Use the ACCOUNT. entry to enable or disable the processing of VAL= entry point programs (refer to appendix F of the system reference manual, volume 2). The currently supported VAL= control statement requests are USER, CHARGE, and FAMILY. Disabling ACCOUNT. causes all VAL= entry point program request statements to be issued to the dayfile but not processed further. ACCOUNT. is normally enabled in a production environment. This feature is usually disabled only during performance testing when it is desirable to reduce the overhead of processing USER/CHARGE statements. The system assumes that the ACCOUNT. feature is enabled if there is no ACCOUNT. entry in the current IPRDECK.

5.3 VALID.

Format	Default	Significance
VALID.	enabled	This option may be disabled to allow jobs to run without USER statements. However, any USER statement encountered is processed as defined by the ACCOUNT. entry in section 5.2. The system will then run jobs with no validation file defined, but no accesses to tapes, permanent files, or removable packs are allowed to these jobs. The console must be unlocked to modify the selection of this feature. The system assumes that VALID. is enabled if there is no entry in the current IPRDECK.

The ACCOUNT. status (section 5.2) relates with the status of VALID. (section 5.3) as follows:

	VALID. Enabled	VALID. Disabled
ACCOUNT. Enabled	A USER statement is required to follow the JOB statement; it will be validated. If a CHARGE statement† is required, it must follow the USER statement; it will be validated, if required.	A USER statement is not required in the job stream, but if it does appear, it is validated. If a CHARGE statement is required,† it must follow the USER statement; it will be validated, if required.
ACCOUNT. Disabled	This is not a valid selection because USER statements are required for validation. On the first occurrence of a VAL=program request, the message IMPROPER VALIDATION is issued.	A USER statement is not required in the job stream. If it does exist, it is entered in the dayfile only. The CHARGE statement is copied to the dayfile, if it exists.

5.4 USERS.

Format	Default	Significance
USERS.	disabled	This option may be enabled to allow jobs to issue more than one USER statement. If disabled, any USER statement encountered after the first causes the job to abort without exit statement processing. If a terminal job issues the USER statement, it is logged off. The following message is issued to the user dayfile. SYSTEM ABORT. The following message is issued to the account dayfile. SIUN, usernum. The system assumes that USERS. is disabled if there is no entry in the current IPRDECK.

†When a USER statement is validated, part of the validation parameters indicate whether a user is required to have a CHARGE statement.

5.5 AUTOROLL.

Format	Default	Significance
AUTOROLL.	enabled	Enter to reverse the current status (enabled or disabled) of the autoroll feature. enabled — Allows automatic job rollout of jobs. This should be the mode of operation if time-sharing is to be supported by the system. disabled — No automatic job rollout. This setting may be advantageous in a batch environment. The system assumes that AUTOROLL. is enabled if there is no entry in the current IPRDECK.

5.6 BATCHIO.

Format	Default	Significance
BATCHIO.	enabled	Use this entry to reverse the status (enabled or disabled) of BATCHIO. enabled — BATCHIO is initialized. disabled — BATCHIO is not initialized. This status is advantageous when not running local batch, because it frees a control point for other use. The system assumes that BATCHIO. is enabled if there is no entry in the current IPRDECK.

5.7 DEBUG.

Format	Default	Significance
DEBUG.	disabled	Use this entry to select or clear debug mode, depending upon the current status. enabled — Debug mode is selected. The message DEBUG appears in the header of the left screen display. Debug mode provides system origin privileges to validated users and allows modifications to be made to the running system. disabled — Debug mode is cleared. It is not recommended to allow debug mode in a normal production environment. The system assumes that debug mode is disabled if there is no entry in the current IPRDECK.

5.8 ENGR.

Format	Default	Significance
ENGR.	disabled	Use this entry to select or clear engineering mode, depending upon the current status. enabled — Engineering mode is selected. The ENGR. message appears in the header of the left screen display. Engineering mode allows the PPU/hardware diagnostics and the 881/883 pack reformatting utility FORMAT to run while the system is in operation. disabled — Engineering mode is cleared.

5.9 JOB CONTROL INFORMATION

The QUEUE, SERVICE, and DELAY entries in IPRDECK relate to job control. The following is general information concerning job control.

5.9.1 QUEUE CONTROL

The first three words in each job control area are for the control of jobs in the queues. These words contain the values used for job aging as follows:

Byte 0	Original (entry) queue priority; specified in OP parameter of QUEUE entry
Byte 1	Lower bound for priority aging; specified in LP parameter of QUEUE entry
Byte 2	Upper bound for priority aging; specified in UP parameter of QUEUE entry
Byte 3	Priority increment; specified in IN parameter of QUEUE entry

If the priority is below the lower bound or above the upper bound, the job is never aged. As long as the job remains in the queue and is within the limits, it is aged upward in the following manner. Every time the aging program 1SP is activated, the value in byte 4 is incremented by one. This value is then checked against the priority increment (byte 3), and when they match, the priority for all jobs is incremented by one.

The aging program is activated according to the cycle time contained in byte 4 of the job scheduler control word, JSCL; refer to a listing of SYSTEXT. This parameter (JS in DELAY entry), along with the queue control parameters, is set in IPRDECK. However, all of these parameters can be changed dynamically with the QUEUE and DELAY commands. The job scheduler cycle time is normally set to 1 second in IPRDECK.

Another queue control feature is the ability to specify the entry priority for jobs in the queue. This is the value that the Export/Import, BATCHIO, and Time-Sharing Module programs use when they enter jobs into the queue. The entry value can be set to give better service to a particular origin type or a particular queue within an origin type.

Some additional comments about the meaning of entry priority for the rollout queue are necessary. This is not the priority given to jobs when they are rolled out; they retain the priority assigned to them at a control point (refer to section 5.9.2). The entry priority for the rollout queue is meaningful only for time-sharing jobs. This is the priority that a terminal job is assigned when reentering the queue after terminal input/output has been performed.

5.9.2 CONTROL POINT CONTROL

The fourth word in the job control point area directs the control of a job while it resides at a control point in central memory. This word contains the following values.

Byte 0	Initial CPU priority set at job initiation or user login; specified in PR parameter in SERVICE entry
Byte 1	CPU time slice in milliseconds divided by 64; specified in CP parameter in SERVICE entry
Byte 2	Central memory time slice in seconds; specified in CM parameter in SERVICE entry
Bytes 3 and 4	Not used

A job leaves a central memory control point because:

A job completes or aborts.

Terminal input/output is required.

The control point is made available for a higher priority job.

The first category is self-explanatory.

In the second category, whenever a time-sharing job exceeds its resource allocation, a check is made to determine whether the job has generated output. If no output is available, the job priority is dropped. If output is available, the job is rolled out. This causes the output to be sent to the terminal. This feature ensures that if a compute-bound terminal job is the only job in the system, output will be provided as it is generated rather than only when the output buffer becomes full.

The third category is the mechanism that ensures reasonable service to all users in the system. In a system servicing a large number of users, restrictions must be placed on the memory time and CPU time each job type can use within a certain period of time. The control point control word specifies the amount of central processor or central memory time each type of job may use when it is at a control point. This is used to ensure that one job does not monopolize system resources.

If a job exceeds either of these resource allocations, and its queue priority is in the range $100_8 \leq qp \leq 7760_8$, the queue priority is set to the lower bound priority for input or rollout files of that origin type. This value normally is less than the entry priority for input and rollout jobs. Thus, any job in the queue with a priority higher than the entry priority forces the resource-bound job to be rolled out. The rolled-out job normally ages until its priority is higher than the priorities of jobs entering the queue and is again scheduled to a control point.

Once a job is scheduled, it is desirable to use the resources allocated before another job forces it out. If a job maintained its queue priority when assigned to a control point, it would be possible for another job to age past that job before having an opportunity to use its time slice. For this reason, when a job is assigned to a control point and is within the queue aging range, it is given a priority equal to the upper bound for the origin and queue type. However, if the entry priority at the time of scheduling is greater than the upper bound priority, the job retains that value.

Selecting the number of control points available on the system depends on the amount of memory space available, the job mix, and the mode the system is being run in. Each control point needs 300_8 words of CMR space. For example, if an installation is running only TRANEX, then four or five control points may suffice. On the other hand, if the system is running a large number of time-sharing terminals with heavy permanent file activity, then 20 or more control points may be needed. Section 4.2 describes the CMRDECK entry for selecting the number of control points. The installation may need to study memory and control point utilization in order to correctly determine the setting of this option. If memory utilization is high and control point utilization is low, the site should select fewer control points. If control point utilization is high and memory utilization is low, more control points should be selected.

5.9.3 MEMORY CONTROL

The maximum memory allowed for any job type can be controlled by use of the fifth word in the job control area. This word has the following format.

Byte 0	Maximum number of jobs; NJ parameter in SERVICE entry. This parameter for time-sharing origin jobs is the maximum number of terminals allowed to log in at any one time.
Byte 1	Maximum field length divided by 100_8 for any job; FL parameter in SERVICE entry
Byte 2	Maximum field length divided by 100_8 for all jobs of the specified origin type; AM parameter in SERVICE entry
Byte 3	Maximum ECS length in tracks divided by 1000_8 for any job; EC parameter in SERVICE entry†
Byte 4	Maximum ECS length in tracks divided by 1000_8 for all jobs of the specified origin type; EM parameter in SERVICE entry†

Byte 0 only has meaning for time-sharing origin jobs. Byte 1 for time-sharing origin jobs is the maximum field length any job is given even though the user is validated for more, the user requests more, or the system calculates a higher value (such as in the algorithm used for FORTRAN or BASIC jobs). Byte 1 for all origin types except time-sharing origin type jobs and byte 2 for all origin types influence scheduling in the following way. Initially, the scheduler attempts to find the highest priority job which meets the constraints specified in both byte 1 and byte 2. However, if the scheduler is unable to schedule a job and has explicitly rejected one or more jobs because of these constraints, a second attempt to schedule a job is made. During this second attempt, any job that requires other jobs to be rolled out is not scheduled. Otherwise, the constraints are ignored. This means that the constraints are applied as long as there are enough jobs of each origin type. However, if core is unused because batch jobs are at a maximum and no other jobs are available, the scheduler will attempt to schedule the batch jobs.

All of these parameters can be changed at the site using the SERVICE, QUEUE, and DELAY commands.

†This feature is not currently utilized by the system, but is provided for future expansion of validation control.

5.9.4 EXAMPLES OF JOB CONTROL PARAMETERS

A theoretical set of job control parameters is illustrated in Figure II-5-1. The following discussion indicates the significance of the values chosen and how they relate to each other.

This is not a recommended set of parameters but strictly an example to aid the local site in selecting its parameters.

		QUEUE PRIORITY						
JOB ORIGIN TYPE	QUEUE TYPE	ENTRY PRIORITY	LOWER BOUND PRIORITY	UPPER BOUND PRIORITY	INCREMENT	TIME SLICE CPU	TIME SLICE CM	INITIAL CPU PRIORITY
SYSTEM	INPUT	6600	700	3000	1	100	20	1
	ROLLOUT	6000	100	1000	2			
	OUTPUT	400	100	7700	1			
BATCH	INPUT	2400	2000	4010	1	400	200	30
	ROLLOUT	2400	1010	4004	1			
	OUTPUT	200	100	7000	2			
EXPORT/ IMPORT	INPUT	3400	2400	4010	1	400	200	30
	ROLLOUT	3400	1400	4006	1			
	OUTPUT	200	100	7600	1			
TELEX	INPUT	4000	3770	7006	1	40	10	30
	ROLLOUT	4004	3740	7000	1			
	OUTPUT	200	100	7000	1			
MULTI-TERMINAL	INPUT	6774	6700	7400	1	400	60	31
	ROLLOUT	6774	4000	7400	1			
	OUTPUT	6000	100	7700	1			

DELAY PARAMETERS

JS	CR	AR	JA	CS	TC
1	10	200	10	10	0

Figure II-5-1. Example of Theoretical Job Control Parameters

The system input queue entry priority is higher than all other entry priorities except for that of the multiterminal origin type because it is assumed that any operator-initiated job should receive prompt attention. A system job rolls out any normal local batch or Export/ Import job. Most system jobs are coded to adjust their priorities correctly, once execution begins.

All of the input queue lower bound limits are higher than the rollout queue lower bound limits so that a job that completes execution in two CPU time slices receives faster service than a job that takes longer. This occurs because the first time a job reaches the end of its time slice, its priority is set to the input queue lower bound for the corresponding job origin type. If the job exceeds its time slice again, its priority is set to the corresponding rollout queue lower bound priority for this and all subsequent rollouts.

Currently, none of the rollout queue entry priorities has significance except for time-sharing origin jobs. The time-sharing rollout queue entry priority is given to a job coming into the rollout queue after being swapped out of central memory for either input or output. The time-sharing input queue entry priority is given to a job when it is entered into the system. Thus, because the time-sharing rollout queue entry priority is higher than the input queue priority, a job that is interacting with the user is given a slight preference.

When a job reaches the end of its time slice, the critical priority ratios are the input queue entry priorities compared to the input queue lower bound priorities, and the rollout queue entry priorities compared to the rollout queue lower bound priorities. In this example, these ratios are especially significant to the terminal user.

- The first time a job exceeds the time slice and is swapped out, it waits 8 seconds (time-sharing input queue entry priority minus the input queue lower bound priority) until the aging program makes its priority equal to the priority of new jobs entering the queue. However, if there are jobs in the queue that have completed terminal I/O, the job waits 12 seconds (time-sharing rollout entry priority minus input queue lower bound priority). This is only true if there are other jobs in the time-sharing input queue.

- The second and each succeeding time a job exceeds its time slice, its priority is set to the time-sharing rollout queue lower bound priority. Once a job's priority has been lowered, the job is swapped out only if a job of higher priority is in the queue (except for the case of time-sharing jobs with output available.) Thus, it is penalized more severely in relation to other jobs that do not exceed their time slices. A compute-bound time-sharing job is given preference over a compute-bound Export/Import or batch job. This can be seen by comparing the rollout queue lower bound priorities in Figure II-5-1 for the various job origin types.

The range between the entry priority and the upper bound priority for all job types in the output queue is quite large. The increment for the output queue is higher than for the other queues. Thus, jobs in the output queue age more slowly, and printer output can wait in the queue for hours.

The queue priorities for batch and remote batch (Export/Import) jobs are set to give remote batch jobs a preference. For example, the first time an Export/Import job exceeds its resource allocation, its priority is lowered to that of entering batch jobs. Thus, only batch jobs that have been waiting in the queue are given preference.

The time slices for the various origin types reflect the following objectives.

- To prevent system jobs with their high entry priority from monopolizing system resources.

- To prevent too much swapping activity from diagnostics running as system origin jobs.

- To allow most batch jobs to run to completion in one time slice. There is no problem with the time slices for batch jobs compared to terminal jobs, because with the priorities shown, a terminal job generally causes a batch job to roll out.

- To ensure reasonably good service to all terminal users without excessive rollouts, the time slices for time-sharing origin jobs are set low. These parameters are especially critical to good terminal performance. In some cases, it may be desirable to change these parameters during operation, depending on the system load, job size, etc.

- To allow completion of multiterminal jobs if at all possible

- Batch jobs have a larger time slice because there is little to be gained from swapping batch jobs.

The CPU priorities reflect the following objectives.

- System jobs are run at the lowest priority. This is done to handle the background CPU and memory diagnostics.

- Multiterminal origin type jobs run at the highest priority in the system. This is done because it is desirable to complete these jobs quickly, and in general, they use little CPU time.

- All other jobs run at the same priority. It is generally not desirable to run one class of jobs at a higher priority than another because the system would swap in jobs which simply occupy memory until they exceed the central memory time slice.

5.10 QUEUE

Format	Significance
QUEUE, ot, qt, qp_1xxxx, . . . , qp_nxxxx.	Use this entry to specify the queue priorities associated with the input, rollout, and output queues for each job origin type.

ot	Job Origin Type
SY	System
BC	Local batch
TX	Time-sharing
EI	Export/Import (remote batch)
MT	Multiterminal

qt	Job Queue Type
IN	Input
RO	Rollout
OT	Output

qp	Queue Priority
LPxxxx	Lowest priority at which a job can enter the queue and still be aged (MNPS $\le$ xxxx $\le$ MXPS). The released values for MNPS and MXPS are 0100 and 7760 octal, respectively.

OPxxxx	Original (entry) priority. This is the priority associated with the job when it initially enters the specified queue. The value of xxxx is normally within the boundaries specifed by LP and UP.
UPxxxx	Highest priority a job can reach in the specified queue; aging stops when this priority is reached. The job is also given this priority when initially assigned to a control point. The value of xxxx is normally greater than LP and OP but cannot exceed MXPS.
INxxxx	Number of scheduler cycles before incrementing the job priority by one.

5.11 SERVICE

Format

SERVICE, ot, $p_1$xxxx, . . . , $p_n$xxxx.

Significance

Use this entry to specify the service limits associated with each job origin type.

ot	Job Origin Type
SY	System
BC	Local batch
TX	Time-sharing
EI	Export/Import (remote batch)
MT	Multiterminal

pi	Service Limits
PRxx	CPU priority ($01 \leq xx \leq 70_8$). Jobs with highest priority get CPU first. All job origin types except SY and MT are normally set to the same CPU priority. System jobs (SY) are run at the lowest CPU priority. Multiterminal (MT) jobs are normally set to a higher CPU priority since they require little CPU time.
CPxx	CPU time slice (milliseconds * 64). This parameter specifies maximum amount of time a job of the specified origin type can use the CPU before its queue priority is set to the lower boundary.

CMxxxx	Central memory time slice in seconds. This parameter specifies the maximum amount of time a job of the specified origin type can remain at a control point before it becomes eligible to be rolled out. The value of xxxx can range from 0 to 7777 octal.
NJxxxx	Maximum number of jobs. For TELEX origin jobs, this parameter specifies the number of terminals that can be logged into the system. The NJ parameter has no meaning for other job origin types.
FLxxxx	Maximum field length/100 for any job of the specified job origin type. Jobs with field length requirements that exceed this value are not scheduled to a control point. It is important to note that this parameter only affects the scheduling of jobs to a control point. Jobs currently assigned to a control point that exceed this value are not aborted. However, if the job is rolled out, it will not be scheduled back to a control point. This parameter is typically used to limit the memory requirement for jobs of a specific job origin type during certain hours of the day. For example, the FL parameter may be used to specify a maximum field length for all batch origin jobs between the hours of 2:00 and 4:00 in the afternoon.
AMxxxx	Maximum field length/100_8 for all jobs of the specified job origin type. This parameter is used to partition central memory by limiting the total field length available to each job origin type. For example, if a job whose field length exceeds that specified for its job origin type is scheduled to a control point, it may not be scheduled until the required field length is available. This means that a lower priority job from a

different origin may be scheduled first. However, a job that would normally exceed the field length for its job origin type can be scheduled to a control point if there are not enough jobs to fill the field length specified for another job origin type. The system always attempts to use central memory to its greatest capacity.

ECxxxx† Maximum ECS/1000 for any job of the specified origin type (expressed in octal)

EMxxxx† Maximum ECS/1000 for all jobs of the specified origin type (expressed in octal)

FCx Number of permanent files allowed. The octal digit x is used as an index into the following table of limits.

x	Limit Value (Octal)
0	Unlimited
1	10
2	20
3	30
4	40
5	50
6	100
7	Unlimited

CSx Cumulative size in PRUs allowed for all indirect access permanent files. The octal digit x is used as an index into the following table of limit values.

x	Limit Value (Octal)
0	Unlimited
1	1000
2	2000
3	5000
4	10000
5	50000
6	100000
7	Unlimited

†This feature is not currently utilized by the system but is provided for future expansion of validation control.

FSx — Size in PRUs allowed for individual indirect access permanent files. The octal digit x is used as an index into the following table of limit values.

x	Limit Value (Octal)
0	Unlimited
1	10
2	20
3	30
4	40
5	50
6	60
7	Unlimited

DSx — Size in PRUs allowed for individual direct access permanent files. The octal digit x is used as an index into the following table of limit values.

x	Limit Value (Octal)
0	Unlimited
1	1000
2	2000
3	5000
4	10000
5	50000
6	100000
7	Unlimited

5.12 DELAY

Format

DELAY, $t_1$xxxx, . . . , $t_n$xxxx.

Significance

Use this entry to specify the system delay parameters.

t_i	Delay
JSxxxx	Job scheduler interval in seconds. This parameter specifies the interval at which the job scheduler and priority increment routines are called. The scheduler may also be called at other times. If not set, the default value is 1 second. This parameter may not be set to 0.
CRxxxx	CPU program recall period in milliseconds. This parameter specifies the amount of time a job remains in recall (X status) when an RCL request is placed in RA+1. This time interval is per individual job. Other activity such as a dropped PPU may cause a job to restart prior to the expiration of this delay period. If not set, the default value is 30_8. This parameter may not be set to 0.

ARxxxx PPU auto recall interval in milliseconds. This parameter specifies the time interval at which PPUs in auto recall are recalled. Only one control point has a PPU activated per time period. If there are more than 14_8 control points, only 14_8 are scanned each time period. If not set, the default value is 1000_8. This parameter may not be set to 0.

JAxxxx Job advance interval in milliseconds. This parameter specifies the time interval at which the system checks to determine if the advance job routine (1AJ) has been called. This field is currently unused. It has been left for possible future utilization.

CSxxxx CPU job switch interval in milliseconds. This parameter specifies the amount of time the CPU executes any one job if several jobs of equal CPU priority all require the CPU. On dual CPU machines the switch interval per CPU is double this. That is, only one CPU is switched at each time interval. If not set, the default value is 10_8. This parameter must not be set to to 0.

TCxxxx Threshold count for corrected SECDED single bit errors. This parameter specifies the number of corrected single bit errors that are detected prior to entering a SECDED message in the error log dayfile, applies only to CDC CYBER 170 machines.

NOTE

This parameter is also used when simulating the S/C register on a CDC CYBER 70 machine. It must be set to zero on a 6000 machine.

5.13 DSD

Format	Significance
DSD,level, command...	Use this entry to specify the initial commands to be executed by the DSD program when the deadstart is complete.

level — Level of deadstart (0, 1, 2, or 3)

command — DSD command to be executed for the level of deadstart specified

Several commands can be specified by separating them with a unique character. This character can be any character not used either as a first character of a DSD command or as a DSD special first character.

NOTE

A character whose display code value is 60_8 or greater is treated as a function code for the display console, rather than as a separator, if it occurs as the upper six bits in a byte.

Example:

DSD, 0, MAI:X. ISF.

5.14 EI200.

Format	Default	Significance
EI200.	enabled	Use this entry to specify whether or not Export/Import is to be initialized.

enabled — Export/Import is initialized.

disabled — Export/Import is not initialized. This status is advantageous if not running remote batch because another control point and PPU are then available for use.

If running only local batch jobs, disable Export/Import and the Time-Sharing Module (TELEX entry).

The system assumes that Export/Import is to be initialized if there is no entry in the current IPRDECK.

5.15 LOCK.

Format	Default	Significance
LOCK.	unlocked	Use this entry to specify whether the system console is to be locked or unlocked.

locked — The console is locked. This software function prevents entry of the restricted commands noted. All other DSD commands can be entered when the console is locked.

The console is normally locked when the system is being used in a production environment.

unlocked — All DSD commands can be entered when the console is unlocked; the following commands are restricted to entry only when the console is unlocked.

DEBUG.
DATE.yy/mm/dd.
TIME.hh.mm.ss.
ENGR.
DISABLE, SECONDARY USER STATEMENTS.
DISABLE, VALIDATION.
ENABLE, SECONDARY USER STATEMENTS.
ENABLE, VALIDATION.
FORMAT, ee.
STEP.
STEP, xx.
n. STEP.
n. STEP, xx.
UNSTEP.
All memory entry commands
All channel control commands

The system assumes that the system console is unlocked if there is no LOCK. entry in the current IPRDECK.

5.16 MAGNET.

Format	Default	Significance
MAGNET.	initialized	Use this entry to specify whether or not the magnetic tape subsystem (MAGNET) is to be initialized. If magnetic tape operations are not used, disabling MAGNET frees a control point for other use. The system assumes that MAGNET is to be initialized if there is no MAGNET. entry in the current IPRDECK.

5.17 MSAL

Format	Significance
MSAL, C, $f_1$xx, . . . , $f_n$xx.	Use this entry to assign job files of type f_i to mass storage device defined by EST ordinal xx. The mass storage device specified must be nonremovable, and its current status must be ON. If the C parameter is entered, any values specified in a previous MSAL entry in the IPRDECK are cleared. If the C parameter is omitted, and an MSAL entry was specified in the IPRDECK, the new values are added to those already specified.

f_i	File Type
LO	Local
IN	Input
OT	Output
RO	Rollout
LG	LGO

5.18 PRIORITY.

Format	Default	Significance
PRIORITY.	enabled	Use this entry to specify whether or not priority aging is to be used.

enabled	Priority aging is active.
disabled	Priority aging does not occur.

5.19 QPROTECT.

Format	Default	Significance
QPROTECT.	enabled	Use this entry to specify whether or not I/O queues are to be protected. If disabled, protected queues are purged during a level 0 deadstart.

5.20 REMOVABLE PACKS.

Format	Default	Significance
REMOVABLE PACKS.	enabled	Use this entry to enable or disable automatic label checking for mass storage devices defined as removable. enabled — Automatic label checking occurs. This status must be available to perform label verification before removable devices can be accessed. disabled — Any removable devices introduced into the system will not be recognized. The system assumes that this feature is enabled if there is no entry in the current IPRDECK.

5.21 TDEN.

Format	Default	Significance
TDEN=density.	800 bpi	Use this entry to set system tape density. When the density is set, any tape unit accessed is automatically set to this density unless specified otherwise by a magnetic tape request. Two TDEN entries may be present, one for 7-track and one for 9-track. HI — 556 bpi (7-track) HY — 800 bpi (7-track) LO — 200 bpi (7-track) HD — 800 cpi (9-track) PE — 1600 cpi (9-track) The system assumes that the density is HY for 7-track and HD for 9-track if there is no TDEN entry in the current IPRDECK.

5.22 TDTR.

Format	Default	Significance
TDTR=track type.	MT	Use this entry to set the default track type. MT — 7-track NT — 9-track

5.23 TELEX.

Format	Default	Significance
TELEX.	enabled	Use this entry to specify whether or not the Time-Sharing Module is to be initialized. If not running time-sharing, disabling TELEX frees a control point and PPU for other uses. The system assumes that TELEX is to be initialized if there is no TELEX. entry in the current IPRDECK.

5.24 TRANEX.

Format	Default	Significance
TRANEX.	enabled	Use this entry to enable or disable the Transaction Subsystem (TRANEX). If the Transaction Subsystem is not being used, disabling TRANEX frees a control point for other use. The system assumes that TRANEX is enabled if there is no TRANEX entry in the current IPRDECK.

5.25 CSM.

Format	Default	Significance
CSM=cc.	63	Use this entry to set the operating system character set mode. To change the character set mode for the common products, a change must be made in IPARAMS, and the products must be reassembled. 63 63-character set 64 64-character set The system assumes a 63-character set if there is no CSM entry in the current IPRDECK.

CAUTION

Unpredictable and possibly serious problems occur if the operating system is operating in one character set and the common product set is operating in another. Therefore, ensure that all installed products and the operating system are in the same mode. Since the products and operating system tapes are normally released in 64-character set mode, it is usually necessary to include a CSM=64. entry in the IPRDECK.

5.26 TCVM.

Format	Default	Significance
TCVM=cc.	AS	Use this entry to set the tape conversion mode to be assumed during system operation. AS ASCII 9-track conversion US USASI 9-track conversion EB EBCDIC 9-track conversion

5.27 KEYPM.

Format	Default	Significance
KEYPM=cc.	26	Use this entry to specify the keypunch mode to be assumed during system operation. 26 O26 keypunch mode 29 O29 keypunch mode This will be used for all batch jobs submitted if the mode is not specified on the job statement.

5.28 CPM.

Format	Default	Significance
CPM,n_1=xx_1, n_2=xx_2.	system selection	Use this entry to alter the CP multiplier of type n_i which is used in SRU calculations. The n_i parameters may be 0 or 1 to indicate the multipliers S0 or S1. Entering 0=xx obtains a multiplier to be used for S0 and entering 1=xx obtains a multiplier to be used for S1. (Refer to part IV, section 1.4 for a discussion of multiplier usage.) The values of xx_i may range from 1 to 6 and are used as indices to values defined in COMSSRU in order to determine the multiplier value. The following are default values.

xx_i	CPU Type	COMSSRU Defined Default Value
1	CP62	1.0
2	CP64	1.0
3	CP66	1.0
4	C172	1.0
5	C173	1.0
6	C175	1.0

GENERAL INSTALLATION PROCEDURES 1

This section describes the general procedures necessary to install and maintain the operating system and product set. It also describes the preliminary operations required before actual installation is possible, the files used in installation, and the order of installation. Specific information for installing the operating system, the maintenance package, the operating system modules, and other products is contained in sections 2, 3, and 4.

1.1 INSTALLING THE OPERATING SYSTEM AND THE PRODUCT SET, AND CREATING A NEW DEADSTART TAPE

1.1.1 TYPES OF INSTALLATION

The release package includes REL tapes for the operating system and product set. These tapes are described in part I, section 2. These tapes include the program libraries and/or binaries for the operating system and product set. The installation and maintenance procedures described in this part provide the following capabilities.

1. The binaries of various products can be installed from the RELeased tapes.

2. Various products and the operating system can be reassembled using one or more of the following.

   a. Released program libraries on REL tapes

   b. PSR corrective code provided on the installation tape (REL0)

   c. User-provided code to change installation parameters

   d. Critical code provided by field support groups

   e. Other user-provided code

3. New deadstart tapes can be created to include one or more of the following.

   a. Binaries of the operating system and products, as created in steps 1 and 2.

   b. User provided binaries, CMRDECKs, IPRDECKs, and LIBDECKs.

1.1.2 INTRODUCTION

To install or modify the operating system and the product set (composed of one or more products), use the installation job decks on REL0. It is important that the installation decks be of the same version as the product to be installed. These decks use permanent files and, therefore, the USER statements used in each job example must be identical.

The following list contains some of the more important features of the procedure to install or modify the operating system and product set.

- No library type files are used (except for SYSTEM). As a result, installation work can be saved across deadstart.
- Options are provided to add installation/critical code to the binaries of various products without adding that code to the program libraries for those products.
- Options are provided to SYSEDIT binaries to the running system, if needed.
- Options are provided to build a system by collecting binaries from released REL tapes.
- Updated REL and deadstart tapes are created.
- The procedures can be run from the console by typing in very few commands. They can also be run from batch by running small card decks.
- The same decks can be used to handle 7- and 9-track tapes by passing appropriate parameters on procedure file calls.
- Options are provided to write (or not to write) the assembly listings to a file.
- Various program libraries can be accessed either from tapes or permanent files. However, if accessing from permanent files, it is the user's responsibility to have the needed files local to his control point.

These installation procedures update old program libraries and create new program libraries on REL tapes. They also assemble various products, generate libraries, and write the binaries for various products to a permanent file called PRODUCT and to REL tapes.

1.1.3 RESTRICTIONS

By design, these installation procedures require the use of a common user number for all installation work. This user number must be validated for 120K of storage, direct and indirect access permanent files, access to the system, and at least one tape drive. These procedures also require some permanent file space; however, the permanent file space required can be greatly minimized. (Refer to section 1.3, step 2.)

1.1.4 GENERAL DESCRIPTION OF INSTALLATION PROCEDURES

The installation procedures consist of nested procedure files. There exists a base deck for each product on a Modify formatted program library named DECKOPL. Each deck is obtained from the DECKOPL by calling the procedure file GENJOB. The name of the product is passed by setting the keyword JOB equal to a specific mnemonic in the statement calling GENJOB. The list of these mnemonics is in Table III-1-1.

For example,

```
CALL(GENJOB(JOB=FTN)
```

obtains deck FTN from DECKOPL. The deck FTN is the install deck for FORTRAN Extended 4. It is a procedure file which in turn calls various other procedure files in order to perform functions such as Updating and library generation. The Modify OPL, DECKOPL, also contains various procedure files, which are used to perform functions that are common across various products.

It is possible to choose various installation options by setting certain keywords to appropriate values in the call statement to GENJOB. A call to GENJOB may be made in the following way.

```
CALL(GENJOB(JOB=deck, S1=x, S2=y, S3=z, MT=track, LIST=YES)
```

Following is a description of the options available.

S1=x Modification option. If omitted, the default value is S1=0.

- S1=0 The job modifies the product with the PSR modifications supplied on the installation tape. These modifications affect the new program library and the new binary file. The product is assembled, and the binaries are copied to the permanent file which is used to generate a new deadstart tape. In addition to the PSR code provided on the batched corrective code tape, the job also modifies the product with the installation/critical code provided by the user on file USER. This code affects the binary files but does not affect the program library.

- S1=1 The job modifies the product with the installation/critical code only, provided by the user on file USER. This code does not affect the new program library, but affects only the new binary file. The product is assembled and the binaries are copied to the permanent files which are used to generate a new deadstart tape.

TABLE III-1-1. INSTALLATION DECKS

Product	Deck Names	
	Installation	Verification
ALGOL 4	ALGOL	VALGOL
-ALGEDIT	ALGEDIT	VALGEDT
APL 1	APL	VAPL
BASIC 3	BASIC3	VBASIC3
CEDIAG 1	CEDIAG	
COBOL 4	COBOL	VCOBOL
COBOL 5	COBOL5	VCOBOL5
COMPASS 3	COMPASS	
Conversion Aids, COBOL 4 to COBOL 5	C4C5	VC4C5
Conversion Aids System 1		
-FCAS	FCAS	VFCAS
-LCAS	LCAS	VLCAS
CYBER Database Control System 1	CDCS	VCDCS
CYBER Loader 1	LOADER	
CYBER Record Manager 1		
-Basic Access Module	BA	VBA
-Advanced Access Module	AA	VAA
Database Utilities 1	DBU	VDBU
Data Description Language 1	DDL	VDDL
Data Description Language 2	DDL2	
Export/Import Module 1	EI200	
FORM 1	FORM	VFORM
FORTRAN Extended 4	FTN	VFTN
-FTN Common Library	FCL	
FTN Extended 4 with Interactive Option	FTNTS	VFTNTS
-FTN Common Library	FCLTS	
Maintenance Tools	TOOLS	
Multimainframe Module 1	MMF	
Operating System	NOS	
Query Update 2	QU	VQU
Query Update 3	QU3	VQU3
Product Texts	TEXT	
Product Texts I/O	TEXTIO	
SIFT	SIFT	
SIMULA 1	SIMULA	VSIMULA
Sort/Merge 4	SORT	VSORT
SYMPL 1	SYMPL	VSYMPL
Time-Sharing Module	TELEX	
TRANEX 1	TRANEX	
Update and CYBER Utilities	UPDATE	
667/669 Controlware	MTS	
8-Bit Subroutines 1	BIT8	VBIT8
881/883 Pack Formatting	FORMAT	
884 Controlware	BCS	

NOTE

> The user may provide any installation/critical code to be used when the S1=0 or S1=1 options are specified. To do this, he must have this code available on a file named USER and must have this file local to his control point. The code on file USER must be in Modify or Update format, depending on the product against which this code is to be applied.

S1=3 The job copies the binary file from the product's REL tape to the permanent file that is used to generate a new deadstart tape.

S2=y SYSEDIT option. If omitted, the default value is S2=0.

S2=0 The job does not SYSEDIT the product's binaries into the running system.

S2=1 The job does SYSEDIT the product's binaries into the running system. The system must be relatively inactive in order to SYSEDIT; in particular, do not run the Time-Sharing Subsystem during this process. In addition, the system must be in debug mode, unless the job is of system origin type.

S3=z z is a local option used by some products. Consult Table III-1-1 and the installation section for each product to determine the correct setting for this option. If the product does not use this option, it may be omitted.

MT=track Track option. If omitted, the default value is MT=MT.

MT=MT The job assumes all tapes to be 7-track.

MT=NT The job assumes all tapes to be 9-track.

LIST=YES Assembly listing option

If LIST=YES is included on the CALL statement, the assembly listings are written to a file named LIST. If LIST=YES is not included in the CALL statement, no assembly listings are written.

1.1.5 INITIAL SETUP OF INSTALLATION PROCEDURES

Since these installation procedures are permanent file dependent, they need to be initialized. The first step is to create DECKOPL and procedure file GENJOB as permanent files. This may be done by using the installation tape (REL 0).

1. Run a job similar to the following to create DECKOPL and GENJOB from REL 0.

```
Job statement
USER(usernumber,password,familyname)
VSN(TAPE=REL0)
LABEL(TAPE,D=800,F=I)
CALL(TAPE)
6/7/8/9
```

2. If any of the installation decks on DECKOPL are to be modified, run the following job.

```
Job statement
USER(usernumber,password,familyname)
COPYBR(INPUT,DECKMOD)
REWIND(DECKMOD)
CALL(GENJOB(JOB=DECKFIX)
7/8/9
     Modifications on the input record
6/7/8/9
```

3. The next step is to run the following job.

```
Job statement
USER(usernumber,password,familyname)
CALL(GENJOB(JOB=GENFILS)
6/7/8/9
```

   This job initializes all permanent files and procedure files used in the installation process. It creates/initializes the following files.

END	MERGE	TAPEOUT	RFT3A
UP	PRODUCT	TAPEAUX	UPDMODS
MO	DIRECT	RFT1E	MDYMODS
TAPEIN	COPYBIN	RFT2A	

   These files are described in detail in section 1.2.

4. Run the following job to obtain listings of the various install decks and procedure files.

```
Job statement
USER(usernumber,password,familyname)
ATTACH(OPL=DECKOPL)
MODIFY(Z,F)/*PREFIXC,+
COPYSBF(COMPILE,OUTPUT,100)
6/7/8/9
```

5. Run the following job to print the PSR summaries from REL0. This job automatically requests REL 0.

```
Job statement
USER(usernumber, password, familyname)
CALL(GENJOB(JOB=PSRLIST, S3=x)
6/7/8/9
```

The following lists are printed, depending on the value of x.

x	Type of List
0	System PSR list. If x is not specified, S3=0 is the default value.
1	All product set PSRs published since the last release
2	All product set PSRs published since the last release, sorted by installation
3	All product set PSRs published since PSR level 383, sorted by routine

1.1.6 ACTUAL INSTALLATION

After initial setup of procedure files, the following steps can be taken to complete the installation process.

1. To install any product, run a job similar to the following.

```
Job statement
USER(usernumber, password, familyname)
CALL(GENJOB(JOB=deck, S1=x, S2=y, S3=z, MT=track, LIST=YES)
```

The various options have been described in section 1.1.4.

The deck name and recommended options are described in Table III-1-1.

2. To generate a deadstart tape, run the following job.

```
Job statement
USER(usernumber, password, familyname)
CALL(GENJOB(JOB=GENSYS, S1=x, S3=y, MT=track)
6/7/8/9
```

S1=x — Tape generation option. If omitted, the default value is S1=0.

- S1=0 — The job generates a deadstart tape from the library file SYSTEM and the permanent file PRODUCT which was generated by installing the products. This permanent file may then be purged depending upon the S3 option.

S1=1	The job generates a deadstart tape from an old deadstart tape and the permanent file PRODUCT which was generated by installing the products. These permanent files may then be purged depending upon the S3 option. The old deadstart tape is requested for via a REQUEST control statement.

S3=y Product file option. If omitted, the default value is S3=0.

S3=0	The permanent file PRODUCT is not purged.
S3=1	The permanent file PRODUCT is purged.

MT=track Track option. If omitted, the default value is MT=MT.

MT=MT	The job assumes all tapes to be 7-track.
MT=NT	The job assumes all tapes to be 9-track.

3. To include user-provided binaries, CMRDECKs, IPRDECKs, and LIBDECKs on the final deadstart tape, the following must be done.

   a. The user-provided information should be available on the file, lfn.

   b. The file USERD should be set up containing the necessary LIBEDIT directives to add this user-provided information to the deadstart tape. These directives should be preceded by the directive:

      *FILE, lfn.

      lfn is the file containing the user provided information.

   c. The files lfn and USERD should be made local to the job described in step 2.

1.2 FILES USED IN THE INSTALLATION PROCESS

The following files are used during the installation and modification of the operating system and product set. For listings of these files, run the job described in section 1.1.5, step 4.

PRODUCT	Direct access file containing absolute binaries of various products. Install decks for various products add appropriate binaries to this file via the LIBEDIT utility.
USER	Local file, established by the user, which contains any installation/critical code
GENJOB	User-called procedure file that selects the desired installation deck and causes it to be processed
UP and MO	Procedure files that perform the Updataing (UP procedure file) or Modifying (MO procedure file) of the product. The S1 option affects the type of Updating or Modifying performed.
END	Procedure file used to build libraries and to add the binaries to the permanent file PRODUCT
DIRECT	Procedure file used to generate the LIBEDIT directives required to add the binaries contained on file PRODUCT to the new deadstart tape
GENSYS	Procedure file used to make the new deadstart tape
COPYBIN	Procedure file used to copy the binaries from the released REL tapes to mass storage files
TAPEAUX	Procedure file used to copy auxiliary tapes to mass storage
TAPEIN	Procedure file used to copy the program libraries from primary tapes to mass storage files
TAPEOUT	Procedure file used to copy the final program library and binaries to the output tapes
NOTE	FORTRAN program used to write individual LIBEDIT directives to a file
MERGE	Procedure file used to merge a product onto a multiproduct tape
GENFILS	Procedure file which initializes all files used in the installation process
RFT1E	Procedure file which reformats the PSR level 410 REL1E for level 420
RFT2A	Procedure file which reformats the PSR level 410 REL2A for level 420
RFT3A	Procedure file which reformats the PSR level 410 REL3A for level 420

UPDMODS	A direct access file which contains the Update formatted PSR modifications
MDYMODS	A direct access file which contains the Modify formatted PSR modifications

1.3 INSTALLATION PROCEDURE SUGGESTIONS

When installing or modifying the operating system and product set, note the following items.

1. Program libraries may be used either from tapes or from permanent/local files. However, if the program libraries are on files other than tapes, note the following items.

   a. Install decks use the file name TAPE for the REL (input tape) of the product being updated. If there is no file assigned by this name, the install decks request this file via a LABEL statement.

   b. Install decks use the file name RELTAPE for the output tape. If it is desired to have the final program library on permanent files, a file named RELTAPE should be local to the user's control point.

2. If many products are to be installed or modified, the permanent files generated by the installation process can become quite large. If it is necessary to minimize disk space, the job described in section 1.1.6, step 2 (generating a deadstart tape), can be run frequently, using the S1=1 and S3=1 options. For example, if a new deadstart tape is created after each installation or modification using these options, disk space is minimized, since the permanent files created by the installation process are purged after the deadstart tape is generated. When doing this, however, it is necessary to rerun the job described in section 1.1.5, step 3, each time a new tape is created.

3. To verify installation, run the following job for each product to be verified.

   ```
   Job card
   USER(usernumber,password,familyname)
   CALL(GENJOB(JOB=deck)
   6/7/8/9
   ```

deck	The verification deck name of the product to be verified. Refer to Table III-1-1 for these deck names.

4. For options S1, S2, and S3, great care should be taken when keypunching. If one of these options is not set properly, the default value may be chosen for this option, without diagnostics being issued. For example, if S1=S1, S1 is set equal to 0. However, if S1=8 is set, the job aborts.

1.4 INSTALLATION ORDER

The required order of installing the product set is illustrated in Tables III-1-2, III-1-3, and III-1-4. The products are identified by installation job deck name. These names are used in the installation job described in section 1.1.6, step 1. Three groups of products are identified.

The headings used in Tables III-1-2, III-1-3, III-1-4, and III-1-5 may require additional explanation. A short description is given for each heading in the following list.

Deck Name	Name of the installation deck for each product to be installed. This name is used in the installation job described in section 1.1.6, step 1.
Recommended Options	Recommended options for S1, S2, and S3. These options are also entered in the installation job.
Product Tape VSN	Volume serial number of the released tape of the product to be installed. This tape should be available when installing the product.
Other Tapes Required	In some cases, the installation job requests tapes other than the tape of the product to be installed. For each applicable product, these tapes are listed in the order they are requested.
Job That Last Used Product Tape	Certain tapes contain more than one product or are used by more than one installation deck. A job deck specified in this column previously used the REL tape of the product to be installed; thus, the output tape of that job should be used as the input tape instead of the REL tape.
Comments	This column contains notes or additional restrictions.

The following group 1 jobs are to be run in consecutive order; no job should be started until the preceding job has finished.

It is recommended that a deadstart tape be built at the end of this phase.

TABLE III-1-2. GROUP 1 PRODUCTS

Deck Name	Recommended Options S1†	S2	S3	Product Tape VSN	Other Tapes Required	Jobs That Last Used Product Tape	Comments
TEXT	0	1	0	REL1E			
TEXTIO	0	1	0	REL1E		TEXT	
UPDATE	0	1	0	REL3A			
COMPASS	0	1	0	REL3A		UPDATE	
LOADER	0	1	0	REL1E		TEXTIO	
NOS	0	0	††	REL1A			Note that S2=0
BCS	0	0	0				Note that S2=0
MTS	0	0	0	REL2A	REL1A		Note that S2=0
BA	0	1	0	REL3B	REL1E		
FTN	0	1	††	REL4A	REL3A		Omit if FTNTS is installed
FTNTS	0	1	††	REL4B	REL3A		Omit if FTN is installed
FCL	0	1	0	REL4A			Omit if FCLTS is installed
FCLTS	0	1	0	REL4B			Omit if FCL is installed
SYMPL	0	1	0	REL2E			
SORT	0	1	0	REL6A			
AA	0	1	0	REL3B		BA	
DDL	0	1	0	REL11B			Omit if DDL2 is installed
DDL2	0	1	0	REL11F			Omit if DDL 1 is installed
COBOL	0	1	0	REL5A			

† The indicated options should be set to assemble and install the various products from the REL tapes released with NOS Version 1. If these products will not be assembled, the binaries for these products may be installed from the RELeased tapes by setting S1 equal to 3. When updating these products with PSR corrective code provided on REL0, use the options documented in the material accompanying REL0.

†† Set S3=1 for 32K installation (LIST=YES must not be used).

The following group 2 jobs should not be started until all jobs in group 1 have been completed. Jobs in group 2 are not order dependent; thus, any number can be running at the same time.

TABLE III-1-3. GROUP 2 PRODUCTS

Deck Name	Recommended Options S1†	S2	S3	Product Tape VSN	Other Tapes Required	Job That Last Used Product Tape	Comments
ALGOL	0	0	0	REL7A			
APL	0	0	††	REL9A	REL1A		
BASIC3	0	0	0	REL8A	REL1A		
BIT8	0	0	0	REL3B		AA	
CDCS	0	0	0	REL11C			
CEDIAG	0	0	0	REL2B	REL1A		
COBOL5	0	0	0	REL5C			
C4C5	0	0	0	REL5D			
DBU	0	0	0	REL11D			
EI200	0	0	0	REL1C	REL1A		
FCAS	0	0	0	REL5B			
FORM	0	0	0	REL3D			
FORMAT	0	0	0	REL2C	REL1A		
MMF	0	0	0	REL1F	REL1A		
QU	0	0	0	REL11A	REL1A REL11B		Install with DDL 1
QU3	0	0	0	REL11E	REL1A REL11F		Install with DDL2
SIMULA	0	0	††	REL10A			
TELEX	0	0	0	REL1B	REL1A		
TOOLS	0	0	0	REL2A	REL1A		
TRANEX	0	0	††	REL1D	REL1A		

† The indicated options should be set to assemble and install the various products from the REL tapes released with NOS Version 1. If these products will not be assembled, the binaries for these products may be installed from the RELeased tapes by setting S1 equal to 3. When updating these products with PSR corrective code provided on REL0, use the options documented in the material accompanying REL0.

†† Set S3=1 for 32K installation (LIST=YES must not be used).

The following group 3 jobs require the output tape from a group 2 job to be used as an input tape. The group 2 job must be completed before the corresponding group 3 job can be started.

It is recommended that a deadstart tape be built at the end of this phase.

TABLE III-1-4. GROUP 3 PRODUCTS

Deck Name	Recommended Options S1 †	S2	S3	Product Tape VSN	Other Tapes Required	Job That Last Used Product Tape	Comments
ALGEDIT	0	0	0	REL7A		ALGOL	
LCAS	0	0	0	REL5B		FCAS	
SIFT	0	0	0	REL2A		TOOLS	

† The indicated options should be set to assemble and install the various products from the REL tapes released with NOS Version 1. If these products will not be assembled, the binaries for these products may be installed from the RELeased tapes by setting S1 equal to 3. When updating these products with PSR corrective code provided on REL0, use the options documented in the material accompanying REL0.

Table III-1-5 contains a list of the verification jobs. Each job can be run after the corresponding product has been installed.

TABLE III-1-5. VERIFICATION JOBS

Deck Names	Recommended Options S1	S2	S3	Product Tape VSN	Other Tapes Required	Job That Last Used Product Tape	Comments
VAA	0	0	0				
VALGEDT	0	0	0				
VALGOL	0	0	0				
VAPL	0	0	0	REL9A	REL1A		
VBA	0	0	0				
VBASIC3	0	0	0				
VBIT8	0	0	0				
VCDCS	0	0	0				
VCOBOL	0	0	0				
VCOBOL5	0	0	0				
VC4C5	0	0	0	REL5D			
VDBU	0	0	0				
VDDL	0	0	0				
VFCAS	0	0	0	REL5B			
VFORM	0	0	0				
VFTN	0	0	0				
VFTNTS	0	0	0				
VLCAS	0	0	0	REL5B			
VQU	0	0	0				
VQU3	0	0	0				
VSIMULA	0	0	0				
VSORT	0	0	0				
VSYMPL	0	0	0				

1.5 OBJECT LIBRARIES

The operating system and product installation decks on REL2A create certain object libraries. The following object libraries are created if all the products supported by NOS Version 1 are installed. The library names that are referenced are reserved for Control Data Corporation.

Libraries

ALGOLIB
BASLIB
BDMLIB
COBOL
FORTRAN
RUN2P3
SIMLIB
SYSIO
SYSLIB
SYSMISC
TRANLIB
TSLIB

To obtain catalogs of these object libraries, run the following job for each library listing required.

```
Job statement
USER(usernumber, password, familyname)
COMMON, SYSTEM.
GTR(SYSTEM, LIB)ULIB/libnam
CATALOG, LIB, N, R.
6/7/8/9
```

libnam	Name of object library whose catalog is desired

The user specified on the USER statement must be validated to use library files in order for this job to run.

OPERATING SYSTEM 2

This section describes what is necessary for the installation and modification of the operating system.

2.1 HARDWARE REQUIREMENTS

The minimum hardware requirements for NOS 1.1 are:

CDC CYBER 70/Model 72-12 or CYBER 170/Model 172-2
Seven PPUs
Two 841 disk systems or one 844 disk system
One line printer
One card reader
Two 657, 659, 667, or 669 tape units

The released NOS system must be modified to run on a 32K system. To accomplish this, modifications to CMRDECK are required (such as table sizes, number of control points, and buffer sizes) and also to LIBDECK (such as central memory resident programs). (Refer to CMRDECK in part II, section 4 and to LIBDECK in part III, section 2.6.) Generally, the minimum configuration only supports one operating system mode at a time. To use additional subsystems and products, additional memory is necessary. Appendix A contains more information concerning installation on a 32K system.

The target hardware configuration for NOS is:

CDC CYBER 70/Model 73-14 or CYBER 170/Model 173-4
Ten PPUs
Two line printers
One card reader
One card punch
Three channels and six 844 disk drives
Six 667 or 669 magnetic tape units
Two 6676 and one 6671 communication multiplexers

2.2 DEFICIENCIES

The deficiencies for NOS will be supplied with the release materials.

2.3 INSTALLATION PROCEDURE

To install or modify the operating system, the following must be done.

1. Complete the initial setup of the installation procedures, as described in section 1.1.5. If this has already been done, there is no need to repeat the procedure.

2. If any code to change installation parameters is available or if any user-provided code exists, place this code in Modify format as named records on file USER.

3. Run a job similar to the following to install and/or modify the operating system.

```
Job statement
USER(usernumber, password, familyname)
COMMENT.  IF USER CODE EXISTS ON FILE USER
COMMENT.  ADD CONTROL STATEMENT(S) TO MAKE THE FILE
COMMENT.  USER LOCAL TO THIS JOB.  FOR EXAMPLE, IF
COMMENT.  USER IS A PERMANENT FILE, ADD A CONTROL
COMMENT.  STATEMENT SUCH AS GET(USER)
CALL(GENJOB(JOB=NOS, S1=x, MT=track)
6/7/8/9
```

The options available with the CALL statement have been described in section 1.1.4; however, the following table describes the selection of appropriate values for x in the S1=x parameter to obtain the desired installation options.

Installation Option	x	Other Action
Assemble the released system OPL without adding any installation or user code.	1	Do not provide the file USER.
Assemble the released system OPL along with installation and/or user code.	1	Have any installation and/or user code available on the file USER; this file should be local to the installation job.
Assemble the released system OPL along with the PSR corrective code provided on the installation tape.	0	Do not provide the file USER.
Assemble the released system OPL along with the PSR corrective code provided on the installation tape and installation and/or user code.	0	Have any installation and/or user code available on the file USER; this file should be local to the installation job.

2.4 INSTALLATION PARAMETERS

Installation parameters for the operating system can be modified using the deck NOS and the procedure described in section 1.3. Obtain listings of the appropriate common decks in order to obtain material, such as line numbers, which is needed when writing code to change installation parameters.

2.4.1 COMPCMX PARAMETERS

COMPCMX is used to calculate the current maximum used in MEMORY requests. The basic calculation for this value is:

current maximum = MIN(validation size,(machine size - CMR size - CMXB))

As an example of this calculation, if x<y, then MIN(x,y)=x.

An additional calculation may be done for each origin type. Currently, TXOT (time-sharing origin type) and MTOT (multiterminal origin type) use the following calculation.

current maximum = MIN(validation size,(machine size - CMR size - CMXB - TELEX size - TFLA), origin type service limit)

Assemble CALLPPU to obtain a listing of COMPCMX.

Parameter	Released Default Value	Significance
CMXB	20B	A constant/100B used in the calculation of maximum machine field length available (machine size - CMR size - CMXB). This is used in determining the current maximum for MEMORY requests.
TFLA	4B	A constant/100B used in the calculation of maximum machine field length available to TXOT and MTOT jobs (machine size - CMR size - CMXB - TELEX size - TFLA). This is used in determining the current maximum for MEMORY requests.

2.4.2 COMSACC PARAMETERS

COMSACC contains a general description of the user validation file. Assemble CALLSYS to obtain a listing of COMSACC.

Parameter	Released Default Value	Significance
APFN	VALIDUZ	This is a micro definition that specifies the name of the file containing user numbers that validate user access to the operating system. Refer to part IV, section 1.1 for further information on VALIDUs.
AUFN	VALINDZ	This is a micro definition that specifies the name of the available user indices file. Refer to part IV, section 1.1 for further information on VALINDs.

For a description of the usage of the following COMSACC user control parameters, refer to part IV, section 1.1.3.

Parameter	Released Default Value	Significance
KNFI	10B	Default limit for the number of local files; the maximum default value is 16B.
KTLI	10B	Default time limit; the maximum default value is 176B.
KLPI	1000B	Default limit for lines printed from a file; the maximum default value is 3776B.
KCPI	0	Default limit for cards punched from a file; the maximum default value is 76B.
KMSI	1000B	Default limit for additionally allocated mass storage PRUs; the maximum default value is 7776B.
KDFI	100B	Default limit for dayfile messages written; the maximum default value is 176B.
KCCI	100B	Default limit for control statements processed; the maximum default value is 176B.
KECI†	0	Default limit for ECS field length; the maximum default value is 176B.
KCMI	10B	Default limit for central memory field length; the maximum default value is 76B.
KSLI	10B	Default limit for SRU accumulation; the maximum default value is 76B.

2.4.3 COMSBIO PARAMETERS

COMSBIO contains parameters used for control of BATCHIO functions. Assemble CALLSYS to obtain a listing of COMSBIO.

Parameter	Released Default Value	Significance
PAGL	64_{10}	The number of lines of print a user is charged for each page of output printed via batch I/O

† This feature is not currently utilized by the system. It is provided for future expansion of validation control.

2.4.4 COMSESS PARAMETERS

COMSESS contains parameters used in diagnostic testing. Assemble CALLSYS or CETEXT to obtain a listing of COMSESS.

Parameter	Released Default Value	Significance
CETK	4000B+1	The reserved track in ECS for DDS ECS testing. The track chosen for CETK should reflect any user requirements such as contiguous ECS tracks. If changes are made to this parameter, CETEXT and DDS must also be reassembled.

2.4.5 COMSEXP PARAMETERS

COMSEXP contains parameters used by Export/Import. Assemble CALLSYS to obtain a listing of COMSEXP.

Parameter	Released Default Value	Significance
PGL	64D	The number of lines of print the user is charged for each page of output printed via Export/Import
N.PORTS	20B	The number of ports that are to be used on a 6671 multiplexer If the terminals have a baud rate that is higher than 2400, decrease N.PORTS which decreases the cycle time. For example, to change from 2400 baud to 4800 baud, change N.PORTS to 6 to decrease the cycle time.
N.TIME	3 milliseconds	Every N.TIME milliseconds, the system services the number of ports specified in N.PORTS. For example, the released values of N.PORTS and N.TIME indicate that the system services 20_8 ports every 3 milliseconds at 2400 baud. Changing N.PORTS to 6 and N.TIME to 1 enables the system to service 6 ports every millisecond at 4800 baud.
T.HUNG	400_{10} milliseconds	This is the line timeout delay.

2.4.6 COMSPFM PARAMETERS

COMSPFM contains parameters used for permanent file symbols and locations, formats of call blocks, and catalog and permit entries. Assemble CALLSYS to obtain a listing of COMSPFM.

Parameter	Released Default Value	Significance
DFPT	DI1	When accessing an auxiliary device with permanent file commands, the permanent file manager checks that the equipment type specified with the R parameter and the pack name are the same as it found when it scanned the equipment. If R is not specified, the equipment type defined with DFPT is assumed. If the default is used for another equipment type, the error message ILLEGAL DEVICE REQUEST occurs.
NFRNG1 through NFRNG7 FSRNG1 through FSRNG7 CSRNG1 through CSRNG7 DSRNG1 through DSRNG7		These parameters control the permanent file access for individual users. There are seven ranges for each classification. The default values and further explanation are in part IV, section 1.1.

2.4.7 COMSPRO PARAMETERS

COMSPRO contains a general description of the PROFILa file. Assemble CALLSYS to obtain a listing of COMSPRO.

Parameter	Released Default Value	Significance
PPFN	PROFILA	This micro definition specifies the PROFILE routine's data base file name. It is further explained in part IV, section 1.2.
PPWD	SECURUS	This micro definition specifies the PASSWORD routine's data base file name.
PUSN	SYSTEMX	This micro definition specifies the catalog location of the PROFILE routine's data base.

2.4.8 COMSREM PARAMETERS

COMSREM contains parameters used by the Time-Sharing Exeuctive. Assemble CALLSYS to obtain a listing of COMSREM.

Parameter	Released Default Value	Significance
UITS	100_8 CPU seconds	This parameter specifies the default CPU time limit for any particular terminal job's activity, if it is not specified with the SETTL command (described in Time-Sharing User's Reference Manual).

Parameter	Released Default Value	Significance
VSPL	20B	VSPL specifies the minimum number of spare pots allowed per 64 terminal users; VMPL specifies the maximum number of spare pots allowed per 64 users.
VMPL	40B	A pot is an eight-central-memory-word internal working buffer in TELEX.
		These two parameters could be reduced for small configurations in order to make more core available for user jobs.
		If the number of pots used is below the minimum default value, the system requests more central memory.
		If the number of pots used is above the maximum value, the system requests the release of central memory.
		Caution should be exercised when lowering this parameter because lost data can occur if system activity increases.

2.4.9 COMSRSX PARAMETERS

COMSRSX contains parameters used by the resource executive. Assemble RESEX to obtain a listing of COMSRSX.

Parameter	Released Default Value	Significance
RPMS	4 minutes	This is the length of time that a job which is waiting for an auxiliary device is kept in the timed/event queue before retrying assignment.
RPOV	8 minutes	This is the length of time that a job which has had a request for an auxiliary device denied because of overcommitment deadlocks is kept in the timed/event queue before retrying assignment.
SUBM	10 minutes	This is the length of time that a non-TXOT job calling RESEX is kept in the timed/event queue before retrying assignment if MAGNET is not active.
MTMS	2 minutes	This is the length of time that a job which is waiting for a magnetic tape with a specified VSN is kept in the timed/event queue before retrying the assignment.
MTOV	8 minutes	This is the length of time that a job which has had a request for a magnetic tape denied because of overcommitment deadlocks is kept in the timed/event queue before retrying the assignment.

2.4.10 COMSSSJ PARAMETERS

COMSSSJ contains parameters used by special system jobs. Assemble CALLSYS to obtain a listing of COMSSSJ.

Parameter	Released Default Value	Significance
ART	4 minutes	This value specifies the default for the NA parameter on the ATTACH control statement or command. It specifies the amount of time that a job will wait in the wait queue for a direct access file to become available before trying to access it again.

2.4.11 COMSLSD PARAMETERS

COMSLSD contains parameters used for referencing information which is maintained in the label sector of a mass storage device. Assemble CALLSYS to obtain a listing of COMSLSD.

Parameter	Released Default Value	Significance
LTKL	20B	If a mass storage device is not initialized during deadstart (using the INITIALIZE entry described in part II, section 4.4.12), the system searches the device for a label that may or may not be in track 0. This parameter specifies the number of tracks the system will search before determining that the device has a bad label or no label. When it reaches that track number (in the released system, track 20_8), it stops searching for a label. If the device is a system device, the system writes a new label; if it is not a system device, the error code LE (label error) and U status (unavailable) are entered in the MST, and the device must be initialized after deadstart. MST is the mass storage table described in the operator's guide.

2.4.12 DSD PARAMETERS

Assemble DSD to obtain a listing. Parameters specified in ENTER macro calls (within the DSD syntax tables) cause the first 25 characters of the associated DSD command to be logged in the system dayfile and/or the error log. The commands are logged just as they are entered by the operator except that the characters:

DS,

are placed before each command. The DSD listing contains an explanation of the ENTER macro.

Parameter	Released Default Value	Significance
SDF	None	When specified in an ENTER macro call, the associated command is logged in the system dayfile.
ERL	OFFxx. ONxx. All memory entry commands All channel control commands	When specified in an ENTER macro call, the associated command is logged in the error log.

2.4.13 MTR PARAMETERS

Assemble MTR to obtain a listing of MTR.

Parameter	Released Default Value	Significance
MSLC	998D	MSLC is an equate in MTR that determines the number of channel 14 clock microseconds that will be considered a millisecond for system time-keeping purposes. This parameter usually does not require change.

2.4.14 PPCOM PARAMETERS

PPCOM contains parameters used by system peripheral processor packages for intercommunication. Assemble SYSTEXT to obtain a listing of PPCOM.

Parameter	Released Default Vaule	Significance
NMSD	20B	NMSD defines the maximum number of EST ordinals that can define mass storage devices. Therefore, in the released system, ordinals 0 through 17_8 can define mass storage devices.
NROS	2	This parameter specifies the number of rollout/rollin operations that can occur simultaneously. This value should be the same as the number of channels specified as available for swapping so that the PPUs do not wait to access channels. The number of channels available for swapping is determined by the TEMP entry in CMRDECK (part II, section 4.4.5) and the MSAL entry in IPRDECK (part II, section 5), or DSD command (operator's guide). TEMP designates the number of mass storage devices that can contain temporary files. MSAL designates the mass storage device to be used for rollout.

Parameter	Released Default Value	Significance
NPFS	4	This is the number of permanent file accesses (PFMs) that are allowed simultaneously. This value should be only slightly larger than the number of channels that are connected to mass storage devices containing permanent file catalogs so that the PPUs do not wait to access channels. For example, a system's permanent files may be split between two devices on two different channels. In this case, NPFS should be set to either 3 or 4.

2.4.15 RESEQ PARAMETERS

Assemble RESEQ to obtain a listing. The micro CSET specifies the time-sharing character set to be used when resequencing BASIC programs. This option applies only to BASIC 3. The selection of the time-sharing character set in RESEQ must correspond to that selected during the installation of BASIC 3.

Parameter	Released Default Value	Significance
CSET	NEW	Selects the new (NOS) time-sharing character set to be used when resequencing BASIC programs. The value of this micro should be set to OLD if the KRONOS 2.1.1 character set is desired for BASIC 3 source programs.

2.4.16 CPUMTR SEGMENTATION

Significant portions of code in CPUMTR support various features such as multimainframe, CMU, and ECS. If a particular feature will not be utilized, it is advantageous to segment that code and not load it at system load time. This helps to minimize the size of CMR.

The presence or absence of these features is detected automatically at CPUMTR load time (during deadstart). The appropriate segments are then loaded to support the desired feature environment. A feature may have as many segments associated with it as necessary, with any number of them being loaded. (The multimainframe feature requires both the multimainframe segment and the ECS segment.)

The features that are presently segmented are CMU, ECS, and multimainframe.

2.5 MAKING A NEW DEADSTART TAPE

Run a job similar to the following to make a new deadstart tape.

```
Job statement
USER(usernumber,password,familyname)
COMMENT.  IF THERE EXIST ANY USER PROVIDED BINARIES
COMMENT.  OR NEW CMRDECKS, IPRDECKS, OR LIBDECKS,
COMMENT.  ADD CONTROL STATEMENT(S) TO MAKE APPROPRIATE FILES
COMMENT.  (USERD,LFN) LOCAL TO THIS JOB.
CALL(GENJOB(JOB=GENSYS,S1=x,S3=y,MT=track)
```

The options available with the call statement are described in section 1.1.6.

Provisions are included in this job to add new CMRDECKs, IPRDECKs, and LIBDECKs to the new deadstart tape. However, in order to do this the following steps must be taken.

1. Set up these decks (CMRDECK, IPRDECK, and/or LIBDECK) as binary records on file, lfn.

2. Set up LIBEDIT directives to add these decks to the new deadstart tape on file USERD. These directives should be preceded by a *FILE,lfn directive.

3. Ensure that the files lfn and USERD are local to the job to create a new deadstart tape.

CMRDECK modifications are described in part II, section 4; IPRDECK modifications are described in part II, section 5; and LIBDECK modifications are described in section 2.6.

2.6 LIBDECK

LIBDECK is a directive record on the deadstart tape. It is read by SYSEDIT during the system load to determine the following information.

Up to eight LIBDECK records may be placed on the deadstart tape. Additional records are named LIBDCKn, where $1 \leq n \leq 7$. A specific record may be selected from the deadstart panel or with a LIB= entry in CMRDECK. The multiple LIBDECK (CMRDECK/IPRDECK) capability enables the use of a single deadstart tape on virtually any system configuration.

Information Defined in LIBDECK	LIBDECK Entry
Which system library records are to reside in the central memory resident library	*CM
Which system library records are to reside on system devices; this is the default residence for routines with no storage area specified in LIBDECK	*MS
Which system library records are to reside on alternate system devices (as well as the system device)	*AD
Which records are procedure files	*PROC
Which programs should use **product set parameter processing**	*SC

A LIBDECK released on the deadstart tape is:

```
LIBDECK
*CM        PP/CIO,2CA,2CB,2CC,2CD,2CE,2CF,2CG,2CH,2CI
*CM        PP/1AJ,TCS,3AE,LDR
*CM        PP/1CJ,1MA,ØBF,ØDF,ØAV,ØRP,ØFA,ØRF
*CM        PP/LFM,3LF,3LG
*CM        PP/2SA,2SC
*CM        PP/1RI,3RH,1RO,3RP,3RQ
*CM        PP/PFM,3PA,3PB,3PD,3PG,3PI,3PK
*CM        PP/1TA,1TO,3TH,3TI
*CM        PP/1MT,3MF,3MG,3MJ,3MS
*CM        PP/1LS
*CM        PP/1IO,1SJ,1SP,1BA
*CM        PP/1DL,9AØ,9A4,9A5,9A6             (DSD RELATED)
*CM        OVL/LDC
*PROC      LIBMOD
*PROC      GENVAL
*PROC      GENHELP
*SC        ABS/FILE,COMPASS,REPORT
*SC        ABS/UPDATE,COPYL,ITEMIZE,DFRCV,DFRST
*SC        ABS/ALGOL,COBOL,FTN,SIMI5,SIMULA,SORTMRG
*SC        ABS/SYMPL,COPY8P,QU,DDL,BASIC,COBOL5
*SC        OVL/RUN
```

2.6.1 MODIFICATION

When determining the residence of system library routines, it is necessary to consider how the programs are used in relation to the characteristics of the hardware.

Concerning the program usage:

1. How large the programs are
2. How often the programs are used

Concerning hardware characteristics:

1. What the storage capacity is
2. What the transfer rate is
3. What the access time is

The following are general guidelines.

1. Large programs (such as the FORTRAN Extended compiler) require a large capacity, high transfer rate storage area.

2. Small programs (such as PFILES, FILES, and most PPU programs) require smaller capacity storage areas.

3. Programs that are frequently used (such as BASIC, PFM, 1AJ, CIO, 1ST, and 1MT) require storage areas with a high transfer rate and access time.

4. Programs that are infrequently used (for example, overlays to PFM such as 3PF, the DEFINE processing overlay, and IMS and MSI, device initialization overlays) require a storage area with lower transfer rate and less fast access time.

For example:

1. Central memory has the best accessibility of any of the possible storage areas. However, the more space that is reserved for system library routines, the less space is available for user programs.

2. ECS is a high transfer (especially to central memory) storage area. It could be utilized for highly used CPU programs.

3. The 841 is a low transfer rate, fast positioning time device. It could be used for small, highly used programs, such as PPU overlays.

4. The 844 is a fast positioning time, medium transfer rate device. It could be used as a system device or as an alternate system device for PPU programs.

2.6.2 DIRECTIVES

The directives that can be in LIBDECK are described in the following paragraphs. Other valid input directives to SYSEDIT (*DELETE and *IGNORE, for example) are described in section 2.6.3.

*CM DIRECTIVE

Specify the system library routines that are to reside in central memory resident with this directive. These routines will reside in RPL (resident peripheral library) and RCL (resident central library).

The format of the directive is:

```
*CM,ty/rec1,ty2/rec2,...,tyn/recn.
```

ty_i/rec_i Identifies the routine's record type and record name

Central memory has the best accessibility of all storage devices. The following programs will automatically reside in central memory.

1. Mass storage drivers
2. Programs specified in the internal tables of SLL (SYSEDIT service routine), such as 1DD and 0DF

In addition, it is suggested that high usage PPU and CPU programs (such as PFILES, CONTROL, 1MT, and PFM) reside in central memory.

Any addition to central memory allows less space for user jobs.

REL type record cannot reside in central memory.

*MS DIRECTIVE

This directive identifies which system library routines will reside on the system device. It is the default residence for routines; any routine not specified as *CM will automatically reside on the system device, even if *MS is not entered in the LIBDECK. System device routines may also be placed on an alternate system device with an *AD assignment.

The format of the directive is:

```
*MS,ty1/rec1,ty2/rec2,...,tyn/recn.
```

ty_i/rec_i Identifies the routine's record type and record name

*AD DIRECTIVE

The purpose of the alternate system device feature is to place copies of specific system library routines on particular mass storage devices so that they will be accessed from a device other than the system device. For example, it is advantageous:

1. To place frequently used system library routines in ECS for faster access than is possible from the system device

2. To place PPU routines on a mass storage device that has better latency than does the system device

Once a routine is on an alternate system device, it is accessed from that device instead of from the system device (*MS LIBDECK entries), except in the following cases of error recovery.

1. If there is an unrecoverable error for a PPU routine on an alternate system device, all of the PPU routines will subsequently be accessed from the system device instead of from any of the alternate system devices.

2. If there is an unrecoverable error for a CPU, ABS, or OVL routine on an alternate system device, only that routine will subsequently be accessed from the system device.

3. If there is an unrecoverable error for an REL CPU routine, the alternate system device must be turned off by the operator to prevent further access to the routine. In this case, the system copy will be used for backup. The alternate system device is not used for loads.

Specify the devices that are to be used as alternate system devices with the ASR entry in the CMRDECK (part II, section 4.4.9); specify the routines that are to reside on each alternate system device in LIBDECK.

The format of the *AD LIBDECK directive to SYSEDIT is:

$$*AD,nn,ty_1/rec_1,ty_2/rec_2,\ldots,ty_n/rec_n.$$

nn Indicates the equipment to be used as an alternate system device; use either the equipment 1 or 2 octal digit EST ordinal or its 2-character equipment type.

The equipment can be any nonremovable mass storage device (including ECS) except for a system device.

If the equipment is not specified in the CMRDECK with the ASR entry, the *AD entry is ignored without an error indication when the system is loaded.

ty_i/rec_i Identifies the routine's record type and record name

ty_i Record type of routine

ABS	CPU multiple entry point overlay
OVL	CPU overlay
PP	PPU absolute
REL	Relocatable CPU routine

If a record type other than ABS, OVL, PP, or REL is specified, the message ILLEGAL CM/AD RESIDENCE. is issued to the output device; the run is aborted.

rec_i Record name of routine

A routine is allowed on only one alternate system device.

Note the following additional qualifications.

1. Once a routine is placed on an alternate system device, SYSEDIT may be used to prohibit access to the routine; however, the space for that routine is not released until LIBDECK is modified and the system is reloaded.

2. IF ECS is an alternate system device, all ABS, OVL, or REL routines residing there will be loaded directly by central monitor function LCEM. This causes the transfer of a program from ECS directly to the load address.

3. If DDP is available, PPU programs residing on ECS will be loaded via DDP, and CPU programs will be loaded via the CPU access to ECS.

*PROC DIRECTIVE

By specifying a TEXT record as a procedure file, a user can use the CALL statement to insert this file (consisting of a group of control statements) at a specific location in his job's control statement stream.

The system obtains the procedure file (specified with the lfn parameter in the CALL statement) by:

1. Searching for a local file, lfn
2. Searching the system library for lfn
3. Attempting to retrieve a working copy of an indirect access file

Procedure files and the CALL statement are defined in section 4 of the system reference manual, volume 1.

The format of the directive is:

*PROC, rec_1, rec_2, . . . , rec_n.

rec_i Record name of routine to be defined as a procedure file

*SC DIRECTIVE

Use this directive to specify that the statements in a certain program are to be processed in product set format rather than NOS format. The formats are defined in section 5 of the system reference manual, volume 1.

The format of the directive is:

*SC, ty_1/rec_1, ty_2/rec_2, . . . , ty_n/rec_n.

ty_i/rec_i Identifies the routine's record type and record name that is to be processed in product set format

2.6.3 SYSEDIT

After the system is loaded, the SYSEDIT control statement provides a method of performing modifications to the system library. A job containing a SYSEDIT control statement must either be a system origin job, or the user must be validated for system origin privileges and have DEBUG set at the system console.

The control statement format is:

SYSEDIT($p_1, p_2, \ldots, p_n$)

p_i — Any of the following in any order

I	Directive input is on file INPUT. If the I option is omitted, file INPUT is assumed.
I=lfn_1	Directive input is on file lfn_1.
I=0	No directive input
B	Binary change statements are on file LGO. If the B option is omitted, file LGO is assumed.
B=lfn_2	Binary change statements are on file lfn_2.
B=0	No binary change statements
L	List output on file OUTPUT
L=lfn_3	List output on file lfn_3
L=0	No list output. If the L option is omitted, the system assumes L=0.
R	Restore to initial deadstart system.
R=n	Restore to copy n of the system. The system copy number is printed on the output listing.
R=0	No system file restoration. If the R option is omitted, the system assumes R=0.
C	Checkpoint the system following SYSEDIT. If the C option is omitted, no checkpoint is performed unless the system was generated employing the alternate system library residency feature. Systems using the alternate library feature are checkpointed automatically following SYSEDIT.

The following are input directives to SYSEDIT. Some of these directives (*CM, *MS, *AD, *PROC, and *SC) have been described in section 2.6.2 as directives which can be contained on LIBDECK. The following is a short description of all directives; a list of valid record types follows the directives.

*AD, nn, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n	Specifies the alternate device to be used instead of the system device(s) for storing ABS, OVL, PP, and REL type routines; nn is either the EST ordinal or the equipment type.
*CM, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n	Defines record rec_i or type ty_i as being central memory resident; legal only for types ABS, OVL, or PP
*MS, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n	Defines record rec_i of type ty_i as being mass storage resident
*DELETE, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n or *D, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n	Deletes record rec_i or type ty_i from the system library. Type ty_i=ULIB is ignored; user libraries cannot be deleted.
*FILE, lfn or *FILE, lfn, NR	Defines file lfn as a file containing system changes. If NR is not present, lfn is rewound before processing.
*IGNORE, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n	Record rec_i of type ty_i is not processed when it appears on the system change file.
*PROC, rec_1, rec_2, ..., rec_n	Defines record rec_i of type TEXT as a procedure file
*RENAME, oe_1-ne_1, oe_2-ne_2, ..., oe_n-ne_n	Renames the CPU entry name oe_i to ne_i
*PPSYN, nam/nam_1, nam_2, ..., nam_n	Adds entries to the system library to provide the synonym nam_i for the PPU program nam
*SC, ty_1/rec_1, ty_2/rec_2, ..., ty_n/rec_n	Defines record rec_i of type ty_i as product set format control statements. The control statement parameters are processed in product set format (refer to volume 1 of the system reference manual, section 5).

The following record types may be specified in SYSEDIT directives.

Type	Description
ABS	Multiple entry point overlay
OPL	Modify old program library deck
OPLC	Modify old program library common deck
OPLD	Modify old program library directory

Type	Description
OVL	Central processor overlay
PP	6000 series peripheral processor program
PPU	7600 peripheral processor program
REL	Relocatable central processor program
TEXT	Unrecognizable as a program
ULIB	User library program

2.7 CONTROLWARE

2.7.1 844 CONTROLWARE

The 844 controlware program BCS that is released on the deadstart tape is identified by card deck part number 2286000/MA710-A08. Use the following job to add new controlware programs to the deadstart tape. It collects the binary card deck for the 844 controlware program BCS and installs the binaries on the permanent files used to generate a new deadstart tape.

```
Job statement
USER(usernumber, password, familyname)
COPYBR(INPUT, IN)
COPYBF(INPUT, INHOLD)
CALL(GENJOB(JOB=BCS)
7/8/9
*COMMENT PPU/BCS **        (** indicates the version of BCS being installed.)
7/8/9
     844 controlware deck
6/7/8/9
```

2.7.2 667/669 CONTROLWARE

The 667/669 magnetic tape controlware program FIRM66X that is released on the deadstart tape is identified by card deck part number 53484900/MB434-A09.

The controlware for driving the 667/669 magnetic tape equipment is required to be on the system for running 667/669 tape operations and also to deadstart from 667 or 669 magnetic tape units.

When deadstarting the system from equipment other than 667/669 equipment, the 667/669 controlware is automatically loaded when MAGNET is brought to a control point. When deadstarting from 667/669 equipment, the controlware must be loaded before the system is deadstarted.

Figure III-2-1 illustrates the structure of the coldstart deck used to load the 667/669 controlware when deadstarting from 667 or 669 magnetic tape units (coldstart).

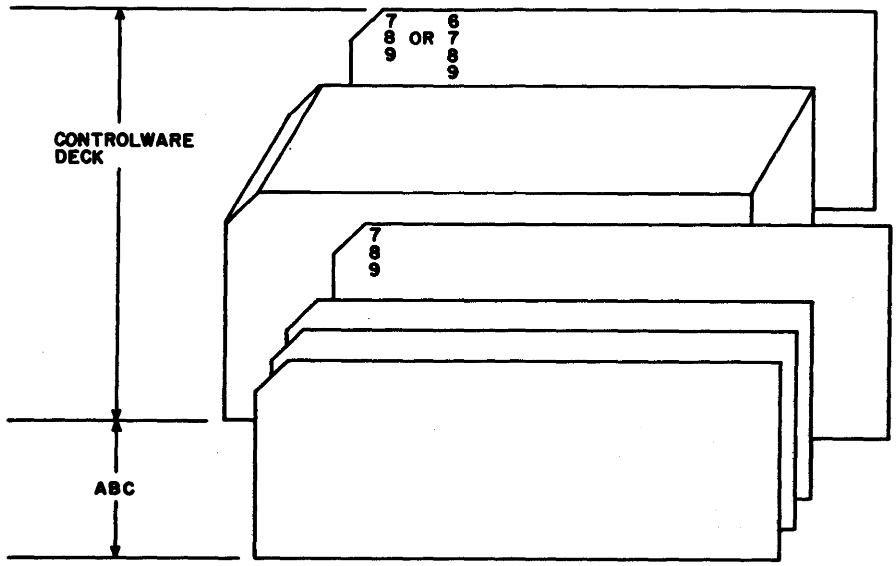


Figure III-2-1. Coldstart Deck Structure

There are two records in the coldstart deck, a three-card program (ABC) and the controlware deck. The program on the deadstart panel reads and executes ABC, which causes the controlware deck to be loaded. When controlware loading is complete, system or maintenance deadstart proceeds.

NOTE

ABC is not on the deadstart tape but is on the REL2A tape.

Use the following job to add new controlware programs to the deadstart tape and also to obtain the coldstart deck. It collects the binary card deck for the 667/669 magnetic tape controlware program FIRM66X, installs the binaries on the permanent files used to create a new deadstart tape, and punches a coldstart deck. (This coldstart deck is also supplied with the released controlware.)

```
Job statement
USER(usernumber,password,familyname)
COPYBR(INPUT,IN)
COPYBF(INPUT,INHOLD)
CALL(GENJOB(JOB=MTS)
7/8/9
*COMMENT PPU/FIRM66X  **  (** indicates the version of FIRM66X being installed.)
7/8/9
    667/669 controlware deck

6/7/8/9
```

2.7.3 PROCEDURE TO IDENTIFY AVAILABLE CONTROLWARE (844 AND 667/669)

If it is necessary to determine which controlware card deck is available, examine either the last two data cards in the card deck (155 and 156) or examine the specific memory locations in the controller's high core using the buffer controller maintenance console.

When examining the cards in the card deck, note that each hexadecimal word of the controller's core memory is punched in two successive columns. The higher order character is first in each pair of columns. On each card, columns 1 and 2 are word count and checksum, column 76 is unused, and columns 79 and 80 are a sequence number.

Octal Card	Columns	Memory Locations Contained in Columns	Contents of Controller Memory Locations
156	14 and 15	0FFD	Product ID (hexadecimal equivalent of 710A, for example)
156	12 and 13	0FFC	Hexadecimal representation of the right four digits of the eight-digit controlware part number
156	10 and 11	0FFB	Hexadecimal representation of the left four digits of the eight-digit controlware part number
156	8 and 9	0FFA	Program code only
156	6 and 7	0FF9	Start of ECO/FCO list; each location contains the hexadecimal equivalent (for 844 controlware) or hexadecimal representation (for 667/669 controlware) of a decimal ECO/FCO number. The list begins with the most recent ECO/FCO number and ends with 0000 (for 844 controlware) or FFFF (for 667/669 controlware).
156	4 and 5	0FF8	
156 and 155	3 of 156 and 77 of 155	0FF7	
155	75 and 76	0FF6	
⋮	⋮	⋮	
155	13 and 14	0FE1	
155	11 and 12	0FE0	

2.8 GENHELP PROCEDURE FILE

GENHELP is a procedure file on the system that is used to generate the file of terminal command explanations for the program HELP. This program is executed using the terminal command HELP (described in the Time-Sharing User's Reference Manual). It provides the terminal user with information and explanations of all the time-sharing commands. This program accesses the file of explanations, which is stored in the permanent file library, whenever a user requests information about a command. To initiate the procedure file to build this HELP library file, run the following job.

X.DIS or JOB,CM60000,T100.	The job must be system origin or system origin priviledges with DEBUG set at console
USER(usernumber,password, familyname) ATTACH(OPL=REL1A)	RELIA must be available
CALL(GENHELP)	Initiates the procedure file
7/8/9	
*BUILD HELPPL	LIBEDIT input directive (refer to volume 1 of the system reference manual)
6/7/9	

The procedure file GENHELP is as follows:

GENHELP	File name statement
RETURN(HLP)	Release file HLP
MODIFY(Z,C=HLP)/*NOSEQ/*EDIT, CMDFILE	Get Modify input directives following the right parenthesis. Edit the deck CMDFILE and write unsequenced compile output to file HLP. Ensure that the OPL is staged before doing so (refer to section 2.10).
LIBEDIT(P=HLP,B=0)	Get compile input from file HLP, directive input from file INPUT, build and append a directory to the compile input, and write it to file NEW. There is no replacement file.
SUI,377776.	Set special permanent file library user index.
DEFINE(CMDFILE/CT=PU,M=R) REWIND(NEW) COPYBF(NEW,CMDFILE)	Specify that CMDFILE be a direct access file with read-only permission and place it in the permanent file library.

2.9 LIBMOD PROCEDURE FILE

LIBMOD, a record on the release tape, is a procedure file used to update user libraries. A copy of this file is as follows:

LIBMOD	
RETURN(Q1,Q2)	Return files for user later.
GTR(OLD, Q1)ULIB/LIB	Get the selected user library.
LIBEDIT(P=Q1, B=LGO, N=Q2)	Replace the routines.
LIBGEN(F=Q2, P=LIB)	Generate a new user library.
LIBEDIT(P=OLD, N=NEW, B=LIB, I=0, L=0)	Generate a new system file.

Before calling this file, file LGO must be created with the routines to be changed. The following is a detailed explanation in example form of each step in the calling sequence, with the appropriate substitutions from the CALL statement.

CALL(LIBMOD(LIB=SYSLIB, OLD=SYSTEM, LGO=CHGS)

RETURN(Q1, Q2)	Make sure files Q1 and Q2 are cleared so these file names can be used later.
GTR(SYSTEM, Q1)ULIB/ SYSLIB)	Get the user library record SYSLIB from SYSTEM and write it to file Q1.
LIBEDIT(P=Q1, B=CHGS, N=Q2)	Take the changes to be made to SYSLIB from the local file CHGS, combine them, and write the result to file Q2.
LIBGEN(F=Q2, P=SYSLIB)	Get the updated record from Q2, generate an updated library, and write it to file SYSLIB.
LIBEDIT(P=SYSTEM, N=NEW, B=SYSLIB, I=0, L=0)	Get the updated library record from SYSLIB, replace the record SYSLIB on SYSTEM with it, and write a new system file to file NEW.

2.10 SYSTEM ORGANIZATION NOTES

To become more familiar with the operation of NOS, use Modify (refer to the Modify Reference Manual) and DOCMENT (refer to volume 1 of the system reference manual) to obtain documentation of the programs from the system program library file (REL1A).

To use these programs, load the OPL from REL1A into mass storage with the STAGE control statement using the following sequence.

Job statement	Refer to volume 1 of the system reference manual for format.
USER(usernumber, password, familyname)	
STAGE(OPL, VSN=xxxxxx)	xxxxxx is the VSN of the released tape.
⋮	
6/7/8/9	

The following are uses of Modify to obtain information and documentation of the system. (The job display program DIS is described in the operator's guide.) In most cases, the OPL deckname, which is the name required on an *EDIT directive for Modify, will be the same as the program name on the deadstart tape. For most overlay decks, such as 9AA, 9AB, etc., the deck in which they are contained is shown in the COMMENTS area of a system CATALOG listing. For instance, for 9AA, the comments read:

DSD - DISPLAY A - DAYFILE MESSAGES

Therefore, edit the dynamic system display program DSD to obtain a listing of the overlay deck 9AA.

The following examples specify uses of Modify.

Example 1:

An assembly listing of the peripheral and central processor communication areas can be very useful to a person seeking detailed information about NOS. To obtain this information, edit SYSTEXT.

The PPU systems communications listing (PPCOM) contains the following information: system constants, PPU memory location assignments, PPU resident and mass storage driver entry points, monitor function values, the contents of low central memory locations, and so forth.

The central program communications macros listing (CPCOM) contains the following information: the contents of a program control area, system request macros, general purpose macros, macros for creation of FETs, and macros to control local file action.

To obtain this listing, at the system console:

Type X.MODIFY(X, CL, Z)/*EDIT, SYSTEXT
Press CR

Any control statement that requires no input, such as the MODIFY and STAGE commands, may be entered directly from the console and is processed as a normal job.

Example 2:

Enter the following from the card reader to obtain external documentation for the job display program (DIS) and the magentic tape executive program.

JOB(CM50000, T77)	
USER (usernumber, password, familyname)	
MODIFY(LO=E, Z/*EDIT, DIS/*EDIT, MAGNET	
DOCMENT.	This statement gets the source from the COMPILE file (from Modify) and generates the external documentation.
6/7/8/9	

Example 3:

Use the following procedure to obtain an assembly listing of the file editing program Modify from a terminal and direct the output to a central site line printer.

Perform standard terminal login (refer to the NOS Time-Sharing User's Reference Manual).

BATCH, 50000	Specify the batch subsystem with a 50K field length when the Time-Sharing Subsystem requests SYSTEM.
MODIFY(Z, X, CL=NEW, CB=0)/*EDIT, MODIFY	The system returns the message ASSEMBLY COMPLETE.
DISPOSE(NEW=PR)	Direct the output file to a line printer.

The operating system prompts each line of input by printing a slash (/) in column 1 at the terminal. This is done only in the batch subsystem to inform the user that another control statement is expected.

When the terminal user is accessing system program decks, the user number and password combination under which this is performed must be validated to access library files. If it is not validated, the message ILLEGAL USER ACCESS will be printed at the terminal. (Refer to part IV, section 1 for validation information.)

Refer to the Modify Reference Manual for further information on listing various programs. The following three parameters can be especially helpful.

X Specifies that the INPUT file containing the Modify directives is to be rewound and the COMPASS assembler is to be called automatically to process the COMPILE file. This parameter cannot be used when running batch jobs because the control statements, the first record on the INPUT file, are read as directive statements, and the message DIRECTIVE ERRORS is issued.

Q Also specifies that the COMPASS assembler is to be called, but does not cause a rewind of the INPUT file. Thus, this parameter should be used instead of the X parameter when running batch jobs for assembly listings.

Z Specifies that the Modify directives will follow immediately after the terminator character following the Modify parameters. The special character following the terminator will be taken as the character which separates multiple directive entries (example 2). The last directive is not followed by a terminator character. If a terminator is used, it will be read as part of the directive and will cause an error message.

If it is desirable to obtain a catalog of the system library, the following job can be run.

```
Job statement
USER(usernumber, password, familyname)
COMMON(SYSTEM)
CATALOG(SYSTEM)
6/7/8/9
```

The user number specified on the USER statement must be validated to use library files to run this job.

MAINTENANCE PACKAGE 3

The maintenance package includes the following.

Product	Released Tape Number
Miscellaneous maintenance tools	REL2A
CEDIAG 1.1	REL2B
881/883 Pack Formatting	REL2C
SYMPL 1.1	REL2E

3.1 MISCELLANEOUS MAINTENANCE TOOLS

The maintenance tools contained on REL2A include the following.

- Binaries for the Time-Sharing Stimulator (STIMULA, 1TS, and DEMUX)
- Binaries for the Dayfile Sort program (DFSORT)
- Binaries for the P register analyzer (PSAMP and SMP)
- Binaries for various games
- Binaries for diagnostic routines
- Binaries for S/C register maintenance programs
- Binaries for the Deadstart Dump Interpreter (DSDI)
- Binaries and program library for SIFT
- CPU debugging routine

3.1.1 INSTALLATION PROCEDURE

INSTALLATION DECKS

Follow the procedures described in section 1.1.5 for the initial setup of the installation decks.

SIFT

Run a job similar to the following to install SIFT.

```
Job card
USER(usernumber,password,familyname)
CALL(GENJOB(JOB=SIFT)
6/7/8/9
```

OTHER TOOLS ON REL2A

Run a job similar to the following to install the binaries of the remainder of the products on REL2A.

```
Job statement
USER(usernumber,password,familyname)
CALL(GENJOB(JOB=TOOLS)
6/7/8/9
```

This job adds the binaries for these tools on the permanent file PRODUCT under the installation user number.

3.2 CEDIAG 1.1, 881/883 PACK FORMATTING, AND SYMPL 1.1

3.2.1 INSTALLATION PROCEDURE

Run a job similar to the following to install any one of these products.

```
Job statement
USER(usernumber,password,familyname)
CALL(GENJOB(JOB=deck, S1=x, S2=y, MT=track)
6/7/8/9
```

The options available with the CALL statement are described in detail in section 1.1.4.

The correct deck name must be used in the JOB=deck option to correctly install each product. These deck names are:

Product	Deck Name
CEDIAG	CEDIAG
881/883 Pack Formatting	FORMAT
SYMPL	SYMPL

3.3 CREATING A NEW DEADSTART TAPE

After running the jobs described in sections 3.1.1 and 3.2.1 to install various products, a deadstart tape can be made which includes these products. To do this, run a job similar to the following.

```
Job statement
USER(usernumber,password,familyname)
CALL(GENJOB(JOB=GENSYS, S1=x, S3=y, MT=track)
6/7/8/9
```

The options available with the CALL statement are described in detail in section 1.1.6.

PRODUCTS AND OPERATING SYSTEM MODULES 4

This section describes the installation of the following products and operating system modules.

ALGOL 4.1
APL 1.2
BASIC 3.1
COBOL 4.5
COBOL 5.0
COMPASS 3.3
Conversion Aids, COBOL 4 to COBOL 5
Conversion Aids System 1.2
- LCAS
- FCAS

CYBER Database Control System 1.0
CYBER Loader 1.2
CYBER Record Manager 1.4
- Basic Access Module
- Advanced access Module

Database Utilities 1.0
Data Description Language 1.1
Data Description Language 2.0
Export/Import Module 1.2
FORM 1.0
FORTRAN Extended 4.6
- FORTRAN Extended 4.6 with Interactive Option
- FORTRAN Common Library

Multimainframe Module 1.0
Query Update 2.1
Query Update 3.0
Product Texts
Products Texts I/O
SIMULA 1.1
Sort/Merge 4.4
Time-Sharing Module 1.2
TRANEX 1.3
Update and CYBER Utilities
8-Bit Subroutines 1.0

Section 4.1 contains installation information which is common to all the above products. Section 4.2 contains special information for some of these products. This special information includes items such as deficiencies and installation parameters.

Table III-4-1 lists the products, operating system modules, and installation and verification deck names and indicates whether special information is needed to install or modify the products. If special information is needed, the section in which this information is contained is listed. If references are not listed for a product, no special information is needed to install and/or maintain that product; follow the procedure outlined in section 4.1.

4.1 COMMON INSTALLATION/MAINTENANCE INFORMATION

4.1.1 HARDWARE REQUIREMENTS

All products included in section 4 can be maintained on the same minimum hardware configuration as the operating system, unless specifically noted in Table III-4-1 and section 4.2.

4.1.2 CORRECTIONS

All operating system modules include all PSR corrective codes published through PSR Summary 430. All other products include all eligible PSR corrective code published through PSR Summary 428.

4.1.3 DEFICIENCIES

None, unless specifically noted in Table III-4-1 and described in section 4.2.

TABLE III-4-1. ADDITIONAL PRODUCT INFORMATION

Product	Installation Deck	Verification Deck	Hardware Requirements	Deficiencies	Additional Procedures	Installation Parameters	Other
Time-Sharing Module	TELEX	-	-	-	-	-	-
Export/Import Module	EI200	-	-	-	-	-	-
TRANEX	TRANEX	-	-	-	4.2.3	4.2.3	-
Product Texts	TEXT	-	-	-	-	4.2.4	-
Product Texts I/O	TEXTIO	-	-	-	-	4.2.4	-
CYBER Loader	LOADER	-	-	-	-	4.2.5	-
COMPASS	COMPASS	-	-	-	-	-	-
Update and CYBER Utilities	UPDATE	-	-	-	-	4.2.7	-
CYBER Record Manager							
• Basic Access Module	BA	VBA	-	-	-	4.2.8	4.2.8
• Advanced Access Module	AA	VAA	-	4.2.8		4.2.8	4.2.8
8-Bit Subroutines	BIT8	VBIT8	-	-	-	-	-
FORTRAN Extended and	FTN	VFTN	-	-	-	4.2.10	4.2.10
FORTRAN Extended with Interactive Option	FTNTS	VFTNTS	-	-	-	-	-
• FORTRAN Common Library	FCL or FCLTS	-	-	-	-	-	-
COBOL	COBOL	VCOBOL	-	-	-	4.2.11	-
Conversion Aids System							
• LCAS	LCAS	VLCAS	4.2.12	4.2.12	-	4.2.12	-
• FCAS	FCAS	VFCAS	4.2.12	4.2.12	-	-	-
Sort/Merge	SORT	VSORT	4.2.13	4.2.13	-	4.2.13	-

TABLE III-4-1. ADDITIONAL PRODUCT INFORMATION (Contd)

Product	Installation Deck	Verification Deck	Hardware Requirements	Deficiencies	Additional Procedures	Installation Parameters	Other
ALGOL	ALGOL ALGEDIT	VALGOL VALGEDT	-	4.2.14	-	4.2.14	-
BASIC	BASIC3	VBASIC3	-	-	-	4.2.15	-
APL	APL	VAPL	-	-	4.2.16	-	4.2.16
SIMULA	SIMULA	VSIMULA	-	4.2.17	-	-	-
Query Update 2	QU	-	-	-	-	4.2.18	-
Data Description Language 1	DDL	-	-	-	-	-	-
CYBER Database Control System	CDCS	VCDCS	-	-	4.2.20	-	-
Database Utilities	DBU	VDBU	-	-	-	-	-
Query Update 3	QU3	VQU3	-	-	-	-	-
Data Description Language 2	DDL2	-	-	-	-	-	-
FORM	FORM	VFORM	-	4.2.24	-	-	-
Multimainframe 1	MMF		-	-	-	-	-
COBOL 5	COBOL5	VCOBOL5	-	4.2.26	-	4.2.26	
Conversion Aids, COBOL 4 to COBOL 5	C4C5	VC4C5	-	4.2.27	-	4.2.27	4.2.27

4.1.4 INSTALLATION/MAINTENANCE PROCEDURE

To install or modify any product or operating system module, the following procedure should be followed.

1. Complete the initial setup of the installation procedures, as described in section 1.1.5. If this has already been done, there is no need to repeat the procedure.

2. If any code to change installation parameters is available or if any user-provided code exists, place this code on file USER. This file should contain Modify or Update corrections, depending on the product being changed.

3. Run a job similar to the following for every product to be installed.

```
Job statement
USER(usernumber,password,familyname)
COMMENT. IF USER PROVIDED INSTALLATION
COMMENT. OR OTHER CODE EXISTS, ADD CONTROL
COMMENT. STATEMENT(S) TO MAKE FILE USER
COMMENT. LOCAL TO THIS JOB.
CALL(GENJOB(JOB=deck, S1=x, S2=y, S3=z, MT=track)
6/7/8/9
```

The options available with the CALL statement have been described in section 1.1.4; however, the following table describes the selection of appropriate values for x in the S1=x parameter to obtain the desired installation options.

Installation Option	x	Other Action
Assemble the product using: • Released program library only	1	Do not provide the file USER.
Assemble the product using: • Released program library • Installation and/or user code	1	Have any installation and/or user code available on the file USER; this file should be local to the installation job.
Assemble the product using: • Released program library • PSR corrective code provided on the batched corrective code tape	0	Do not provide the file USER.
Assemble the product using: • Released program library • PSR corrective code provided on the batched corrective code tape • Installation and/or user code	0	Have any installation and/or user code available on the file USER; this file should be local to the installation job.
Install the binary of the product from the RELeased tape, without actually assembling it.	2	None

In the JOB=deck option, set deck equal to the deck name for the product to be installed. Table III-4-1 contains the installation deck names for the various products.

4.1.5 CREATING A NEW DEADSTART TAPE

The installation job described in section 4.1.4 adds binaries of the products being installed to a permanent file. The following job may be used to add these binaries to a new deadstart tape. This job also has provisions to add user-provided binaries to the new deadstart tape. If a user wants to make use of this feature, however, he must do the following.

1. Provide all binaries to be included on a file lfn.

2. Set up LIBEDIT directives to add these binaries to the new deadstart tape on file USERD. These LIBEDIT directives should be preceded by a

   *FILE, lfn

   directive.

3. Ensure that files lfn and USERD are local to the job to create a new deadstart tape.

4. Run the following job to create the new deadstart tape.

```
Job statement
USER(usernumber, password, familyname)
COMMENT. IF ANY USER PROVIDED BINARIES EXIST,
COMMENT. ADD CONTROL STATEMENT(S)
COMMENT. TO MAKE FILES LFN AND USERD LOCAL TO
COMMENT. TO THE JOB.
CALL(GENJOB(JOB=GENSYS, S1=x, S3=y, MT=track)
6/7/8/9
```

The options available with the CALL statement are described in section 1.1.6.

4.1.6 VERIFICATION

Run a job similar to the following to verify the correct installation of various products.

```
Job statement
USER(usernumber,password,familyname)
CALL(GENJOB(JOB=vdeck)
6/7/8/9
```

vdeck is the verification deck for each product to be verified. Table III-4-1 contains the deck names for the verification decks; not all products have verification decks. If Table III-4-1 does not list a verification deck name for a particular product, then the deck does not exist.

4.2 SPECIAL INSTALLATION INFORMATION FOR PRODUCTS AND OPERATING SYSTEM MODULES

4.2.1 TIME-SHARING MODULE 1.2

No special information is needed to install Time-Sharing Module.

4.2.2 EXPORT/IMPORT MODULE 1.2

No special information is needed to install Export/Import Module.

4.2.3 TRANEX 1.3

ADDITIONAL PROCEDURES

When installing TRANEX, note the following additional requirements.

1. When creating the user validation file, assign a user index, a user number, and a password to TRANEX.
2. When modifying installation parameters, include these assignments in TRANEX by modifying the three parameters described in Default User Number/Password and User Index in this section.
3. After TRANEX is installed on the deadstart tape but before it is used, create a task library permanent file containing the five tasks required by TRANEX. These tasks, which are on the release tape REL1B, are:

ITASK	Task supervisor
KDIS	TRANEX K display driver
MSABT	Message sender for abnormally terminating tasks
OFFTASK	Inactive task controller
SYSMSG	Message task for system origin messages

4. If TRANEX will be used in a multimainframe environment, concurrent access to the same data base cannot be allowed. The copy of TRANEX in each machine must have its own user number/user index or default family.

To create a task library containing these tasks, run a job similar to the following.

```
JOB, CM55000.
USER(usernumber, password, familyname)                     Use the user number and
                                                           password assigned to
                                                           TRANEX previously.
VSN(REL1A=REL1A)
LABEL(REL1A, R, FI=NOS1P1*419420, MT, D=HY, F=I)
COPYEI(REL1A, OPL)
UNLOAD(REL1A)
RETURN(REL1A)
VSN(REL1D=REL1D)
LABEL(REL1D, R, FI=TRX1P3*419420, MT, D=HY, F=I)  TRANEX OPL
COPYBF(REL1D, TRNPL)
UNLOAD(REL1D)
RETURN(REL1D)
MODIFY(LO=E)
COMPASS(I, L=0)
DEFINE(TASKLIB/CT=PU)
PERMIT(TASKLIB, usernumber=W)                              Usernumber must be same
LDSET(LIB=TRANLIB)                                         as specified on the USER
LOAD(LGO)                                                  statement.
NOGO(LGOB)
LIBTASK(CR)
7/8/9
*OPLFILE        TRNPL
*EDIT           ITASK, KDIS, MSABT, OFFTASK, SYSMSG
7/8/9
*ITASK          S.
6/7/8/9
```

INSTALLATION PARAMETERS

The following parameters may be changed to fit the requirements of a specific installation. For jobs used to change installation parameters, refer to section 4.1.4.

COMKCBT PARAMETERS

Parameter and Released Default Value			Significance
CBTA	EQU	24	Number of the bit of the user area which signifies a binary terminal; a value of 24 implies that no bit is used.
CBTB	EQU	0	Maximum number of characters for binary input; a value of 0 implies the default value 1.
CBTC	EQU	4000B	Terminating character for binary input; a value of 4000B implies that no character has been selected.

Parameter and Released Default Value			Significance
USNM	MICRO	1,7,/KB100DC/	TRANEX user number, TRANEX.699
PWDM	MICRO	1,,//	TRANEX password, TRANEX.700
TRUI	EQU	16B	User index for TRANEX, TRANEX.701

The following parameters specify the default initialization K display options.

Parameter and Released Default Value			Significance
NSCP	EQU	12	Number of subcontrol points, TRANEX.815
NCMB	EQU	4	Number of communication blocks, TRANEX.823
SCMFL	EQU	100000B	Maximum field length, TRANEX.831
IFL=	EQU	50000B	Initialization field length, TRANEX.833
IMDM	EQU	3	Number of sets of data manager buffers (used by TRANEX to limit the number of transactions making data manager requests), TRANEX.851
ECSFL	EQU	0	ECS field length, TRANEX.842

The following specify the default DSDUMP parameters.

Parameter and Released Default Value			Significance
DFWA	EQU	100B	Default FWA for task dump, TRANEX.786
DLWA	EQU	300B	Default LWA for task dump, TRANEX.788
DEXP	EQU	1	Default exchange package dump, TRANEX.790
DDMB	EQU	0	Default data manager buffer dump, TRANEX.792
DORT	EQU	0	Default origin type for task, TRANEX.794
DORC	EQU	BCOT	Default origin code, TRANEX.796
DQDS	EQU	0	Default queue destination value, TRANEX.797

The following specify default time dependencies. Although these values are expressed in milliseconds, they are accurate only to 1 second.

Parameter and Released Default Value			Significance
CORTL	EQU	1*1000	Time to look for FL to release, TRANEX.717
RRTTL	EQU	1*1000	Time to evict a reusable task, TRANEX.718
TACTL	CON	2*60*1000	If no transaction input has arrived, an internal call is generated to ITASK, TRANEX.725
ITRTL	CON	1500	Time to wait for input before rollout of TRANEX field length, TRANEX.727
TROTL	CON	10*60*1000	Duration of rollout, TRANEX.728

The following specify default task rollout parameters.

Parameter and Released Default Value			Significance
RTDNL	EQU	2*1000	Number of milliseconds a task is allowed to remain in core waiting for a CALLRTN to complete, TRANEX.737
NESTL	EQU	16	Nest limit for CALLRTN (must be less than 64), TRANEX.736
DWITL	EQU	8*60	Default time in seconds a task is allowed to wait for terminal input, TRANEX.740
DMRTL	EQU	2	Time in seconds in which a task is rolled out after a data manager detected conflict, TRANEX.741
ROLTO	EQU	40	Number of words one task can send to the communication subsystem before it is rolled out, pending completion of the terminal output

The following specify default miscellaneous constants.

Parameter and Released Default Value			Significance
DTSTL	EQU	16	Default number of time slices for a task, TRANEX.746
TSKTL	EQU	120	Millisecond duration of a task CPU time slice, TRANEX.747
MAXCW	EQU	240	Cumulative word count limit of terminal output from one task, TRANEX.750
MAXJL	EQU	2500	Maximum word count on one journal request, TRANEX.751
MAXRA	EQU	500	Default task limit for RA+1 requests, TRANEX.745
DTYM	MICRO	1,2,/DI/	Default device type, TRANEX.782
MAXTO	EQU	400	Maximum number of words one task can send to the communication subsystem

The following parameters are for use with the Total Data Base Manager. For information on the installation of Total, refer to the Application Installation Handbook.

Parameter and Released Default Value			Significance
TMAXDB	EQU	31	Maximum number of Total data bases which can be initialized, TRANEX.767
TIMDM	EQU	10	Maximum number of tasks which can have outstanding requests to Total at one time; additional tasks are put in recall, TRANEX.766
TMAXFIL	EQU	100	Maximum number of files per data base, TRANEX.768
DBIDF	EQU	1	TRANEX data manager load flag, TRANEX.776 If 0, TRANEX data base manager is loaded only if DBID file is present If 1, TRANEX data base manager is loaded. If file DBID is missing, abort with error occurs. If -1, TRANEX data base manager is not loaded even if DBID is present.
TDBIDF	EQU	-1	Total data manager load flag, TRANEX.777 If 0, Total data base manager is loaded if TDBID file is present. If 1, Total data base manager is loaded. If file TDBID is missing, abort with error occurs. If -1, Total data base manager is not loaded even if TDBID is present.

4.2.4 PRODUCT TEXTS AND PRODUCT TEXTS I/O

INSTALLATION PARAMETERS

General installation parameters related to the common product set are defined within the COMDECK IPARAMS, included in Product Texts.

The default values of the IPARAMS configuration parameters are defined with the CEQU or CMICRO macros so that an installation can insert all modifications at one place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications should precede them. (For jobs used to change installation parameters, refer to section 4.1.4).

The following list constitutes the extent of installation changeable symbols in IPARAMS.

Parameter	Released Default Value	Significance
IP.CMU	0	If nonzero, compare/move unit hardware is present. The following common products reference IP.CMU. CRM 1 COBOL 4 COBOL 5 Sort/Merge 4
IP.CSET	IP.C64.1	Defines character set to be used throughout the system. The following common products reference IP.CSET. COBOL 4 Sort/Merge 4 BASIC 3 SIMULA 1 FORTRAN 2.3 CRM 1 Update 1 COMPASS
MODEL	74	Micro whose value is the CDC CYBER 70/170 model number corresponding to the type of central processor to be used for optimal code generation. Acceptable values are 72, 73, 74, 172, 173, 174, or 175. Most common products reference the MODEL micro.
OS.ID	NOS 1.1	System identification micro for displaying the operation name and version number in generated program binaries Most common products reference the OS.ID micro.

4.2.5 CYBER LOADER 1.2

INSTALLATION PARAMETERS

The following parameters may be set at LDRCOM. 13 in the update of REL1E.

Parameter	Released Default Value	Significance
IP. PSET	1	Core presetting options: 0 No presetting 1 preset to 0000 0000 0000 0000 0000B 2 preset to 7777 7777 7777 7777 7777B 3 preset to 1777 0000 0000 0000 0000B 4 preset to 3777 0000 0000 0000 0000B 5 preset to 6000 0000 0000 0000 0000B 6 preset to 4000 0000 0000 00 addrB 7 preset to 2525 2525 2525 2525 2525B 10 preset to 5252 5252 5252 5252 5252B NOTE addr is the core address; this is specified in lower 18 bits.
IP. REW	1	1 File is rewound prior to beginning of load 0 No rewind takes place
IP. LDER	1	Error processing by the loader may be one of the following. 0 Abort on all errors (ERR=ALL) 1 Abort on fatal errors (ERR=FATAL) 2 No abort if possible (ERR=NONE)
IP. FLINC	4000B	Amount of field length increase if additional field length is required for table construction by the loader. Acceptable values are multiples of 100B.
LINP	60D	Specifies the number of lines per page in a load map.
IP. LRT	0	If nonzero, a message given various time and memory measurements is issued to the dayfile. If IP. LRT ≥ 1000B, the value IP. LRT-1000B is placed in bits 29-18 of the MSG call.
IP. LDBG	0	If nonzero, conditional code to aid in debugging the loader is assembled.

Parameter	Released Default Value	Significance	
IP.FLMSG	0	If nonzero, a dayfile message giving the field length required for loading and execution is issued. This applies for relocatable loads when no map is specified.	
IP.MAP	3†	Default loader map option	
		0=MAP(OFF)	No map
		3=MAP(PART)	S, B options
		13B=MAP(ON)	S, B, X options
		17B=MAP(FULL)	S, B, E, X options
		S	Loader statistics and error messages only
		B	Block name, addresses and lengths
		E	Entry point list
		X	Cross reference list of external references

4.2.6 COMPASS 3.3

No special information is needed to install COMPASS.

† If CYBER Loader is installed using the program library released on REL1E, the default value is 3. However, if the binary on REL1E is used, the default value is 0.

4.2.7 UPDATE 1.2 AND CYBER UTILITIES

INSTALLATION PARAMETERS

The following Update features are available or unavailable through assembly options and may be modified by deleting the appropriate entry in the range UPDATE.703 through UPDATE.711. For jobs used to change installation parameters, refer to section 4.1.4.

An attempt to use features when the option is not assembled causes Update to issue error messages. For example, when PMODKEY is not set, the PULLMOD statement is not recognized as a legal directive.

Parameter	Released Default Value	Significance
DECLKEY	Enabled	Enables DECLARE directive
PMODKEY	Enabled	Enables PULLMOD statement and G option
AUDITKEY	Enabled	Allows audit functions
EDITKEY	Enabled	Allows merge and edit
OLDPLKEY	Enabled	Enables Update to read both old-style and new-style old program libraries
EXTOVLP	Enabled	Enables detection of four types of overlap involving two or more cards in a correction set
DYNAMFL	Enabled	Declares dynamic table expansion. When this option is assembled, Update automatically expands tables as required and dynamically requests NOS to change the user field length to accommodate the additional table area. At the end of the run, the field length is reduced to that requested by the user.

Update also uses the symbol IP.CSET and the micro MODEL defined in IPTEXT.

4.2.8 CYBER RECORD MANAGER 1.4

CYBER Record Manager modules are released in two parts, Basic Access Module and Advanced Access Module.

BASIC ACCESS MODULE

Basic Access Module performs the sequential and word addressable I/O functions. In addition, linkage modules are included to handle the interfaces to Advanced Access Module (IS, DA, AK).

INSTALLATION PARAMETERS

The installation parameters described below permit a certain amount of tailoring. To facilitate writing the Update statements, each installation parameter has a unique mnemonic Update identifier. To change the parameter from its default value to a user value, the following process is required. (For jobs used to change installation parameters, refer to section 4.1.4.)

```
*DELETE            <mnemonic>.1
=<parameter>=      EQU <user-value>
```

Use the following Update *IDENT format to set installation parameters.

```
*IDENT       URM<mmddyy>
 mm          month
 dd          day
 yy          year
```

Parameter	Mnemonic Update ID	Released Default value	Significance
DBG	DBG	Off	Causes extensive debug code to be assembled if defined
MCTL	MCTL	100 decimal	Memory catalog table length. MCTL/2 is the maximum number of files allowed open at one time.
LBLIM	LBLIM	10 decimal	Length of label buffer. Size limit of a user label string. Each user label requires nine words. LBLIM should be n*9+1, where n is the maximum number of labels permitted (HDR1-9,...).

One feature is sensitive to IPARAMS. This is the use of the compare move unit (CMU) hardware. If IP.CMU is nonzero, CRM routine MOVE.RM assembles using the CMU hardware. For records over 40 characters, the CMU hardware reduces CP time of a program using CRM.

The IPARAM (IP.CMU) can be overridden as follows:

To turn CMU on:

```
*DELETE CMU.1
≡CMU≡ MICRO 1,,/1/
```

To turn CMU off:

```
*DELETE CMU.1
≡CMU≡ MICRO 1,,/0/
```

ADVANCED ACCESS MODULE

Advanced Access Module (IS, DA, AK) provides all the routines necessary to create, Update, and access random files on mass storage. These files can be retrieved by key as well as sequentially.

A key analysis utility routine is available to aid in the selection of a hashing routine.

A create utility is available for efficiently creating DA files.

There are two utility routines called by control statements for indexed sequential files. SISTAT prints the statistics for an existing IS file; ESTMATE produces estimates of block and buffer sizes from input statements containing IS file descriptions.

DEFICIENCIES

If either DA or AK is used in a program with overlays, it should be included in the main overlay. Neither DA nor AK routines can be overlayed. SIS processing can be done outside the 0, 0 overlay but the SIS processing must not span overlays. A file processed outside the 0, 0 overlay must be closed before another overlay is called.

The create utility requires that Sort/Merge be installed. If Sort/Merge is not available, comparable DA files can be created through explicit Record Manager calls at the expense of appreciably greater creation time.

INSTALLATION PARAMETERS

The system contains parameter values that are effective when the user does not supply settings. The default parameters are defined on the program library tape REL3B in the common decks SDACOM (for DA parameters), SISCOMM (for IS parameters), and SAKCOM (for AK parameters). For jobs used to change these parameter settings, refer also to **section 4.1.4.**

1. IS PARAMETERS

Parameter	Released Default Value	Significance
DAHDRSZW EQU	1	Specifies data block header size in words; range is 1 through 31; location is SISCOMM.12
DFBKGFAC EQU	2	Specifies default data record blocking factor; range is 1 through 4095; location is SISCOMM.16
DFDAPADP EQU	0	Specifies data block padding factor; range is 0 through 99; location is SISCOMM.17
DFERRLIM EQU	26	Specifies maximum number of trivial errors +1; for IS 1.0 programs only; range is 1 through 32,767; location is SISCOMM.18
DFIBKSZW EQU	511	Specifies default index block size in words; range is 1 through 23,767; location is SISCOMM.19
DFINPADP EQU	5	Specifies default index padding factor; range is 0 through 99; location is SISCOMM.20
DFNRLVLS EQU	1	Specifies default number of index levels; range is 1 through 63; location is SISCOMM.21
KEYLIMIT EQU	255	Specifies maximum key size in characters; range is 1 through 511; location is SISCOMM.31
TOTFILES EQU	10	Specifies maximum number of active IS files per run; defines an internal table size in words; no practical limit; location is SISCOMM.338

2. DA PARAMETERS

Parameter	Released Default Value	Significance
BLKHDL EQU	1	Specifies block header length (CM words range is 1 through 31 If the value specified is not within this range, the following message is issued. INSTALLATION PARAM BLKHDL OUTSIDE 1-31 RANGE When modifying this parameter, the update sequence number is SDACOM.38.

3. AK PARAMETERS

Parameter	Released Default Value	Significance
PRCNTBL EQU	25	Specifies percent required for random probing; range is 0 through 100
PROBLEM EQU	4	Specifies number of random probes limit (no limit)
DBLKFK EQU	8	Specifies default blocking factor; limited by user field length
BLKHDL EQU	1	Specifies block header length in words; limited by FLK size
NMOPNFL EQU	10	Specifies number of allowable opened files

4.2.9 8-BIT SUBROUTINES 1.C

No special information is needed to install 8-Bit Subroutines.

4.2.10 FORTRAN EXTENDED 4.6 AND FORTRAN EXTENDED 4.6 WITH INTERACTIVE OPTION

SPECIAL NOTES

FORTRAN Extended with Interactive Option is released separately on REL4B. It contains an additional time-sharing mode. When TS is selected on the FTN control statement, FTN operates as a one-pass compiler instead of a two-pass compiler. It uses less field length and less CPU time than OPT=0, 1, or 2. No overlay reloading is done during compilation; thus, I/O time is reduced. The object code produced is not optimized and executes more slowly than that produced by OPT=1 or 2. FORTRAN Extended with Interactive Option is also used under the FTNTS subsystem.

The following discussion applies to both FORTRAN Extended and FORTRAN Extended with Interactive Option.

1. Selection of the debug mode for compilation (D control statement parameter requires approximately 17K (octal) additional central memory field length above that required for an OPT=0 compilation.

2. Selection of the full cross reference map feature (R=2 or R=3) may require additional central memory for compilation, depending upon program size and complexity. Refer to the FORTRAN Extended Reference Manual for further information.

3. The intrinsic function SHIFT does not accept two-word arguments (types DOUBLE PRECISION or COMPLEX).

4. FORTRAN Extended is designed to produce efficient object code. The rate of compilation tends to be higher on program units that avoid lengthy sequences of complicated arithmetic replacement statements (such as contiguous statements with no branching entries or exits). Up to a certain point, however, long sequences may result in faster object code, because register contents can be optimized over several statements.

5. If FORTRAN Extended is installed on a CDC CYBER 70/Model 72 or 73 or on a CYBER 170/Model 172, 173, or 174 with the MODEL installation parameter correspondingly set, the object code produced may not work properly on a model 74 or 175 because of optimization and timing considerations. The reverse is not true; if MODEL is set to 74 or 175, the compiled object code will execute correctly on a CDC CYBER 70/Model 72, 73, or 74 or on a CYBER 170/Model 172, 173, 174, or 175 (although it will be slightly suboptimal for the model 72, 73, 172, 173, or 174).

6. When the FTN control statement specifies either the C or E option, the compiled object code is produced as symbolic COMPASS source language rather than executable binary. Since a library set cannot be specified in a COMPASS source program, a user must place the NOS control statement LDSET(LIB=FORTRAN/SYSIO) at appropriate points in the control statement section of the card deck. FTN routinely issues a precautionary 8-type warning reminder in the symbolic program when the C or E option has been selected.

INSTALLATION PARAMETERS

The FORTRAN Extended compiler program library, REL4A is distributed with installation parameters properly set for a normal installation in a NOS operating system on a CDC CYBER Model 74, 175, or 6600 computer. If the product is being installed on a CDC CYBER Model 72, 73, 172, 173, or 6200/6400 computer, more optimal object code may be obtained (at the expense of model 74 or 175 compatibility) by adding the following change to the FTN deck. (For jobs used to change all installation parameters, refer to section 4.1.4.)

```
*DELETE OPTIONS.60
 MODEL        MICRO        1,,$72$        (or $73$, $172$, $173$, or $174$ as
                                           appropriate)
```

If the machine is a 175, MODEL must be set to 175 or 74.

The compiler source listing is formatted with 60 lines per page. This may be revised for different sizes of printer paper as follows:

```
*DELETE OPTIONS.240
 L.PAGE       EQU          nnD            nn is maximum lines per page.
```

Other less frequently revised installation options are located in the common deck OPTIONS and the deck FTN. OPTIONS is called by FTNMAC and FTNTEXT because of its global nature.

Current UPDATE sequence numbers for installation option may be obtained by assembling FTNMAC or FTNTEXT (the FTNMAC listing is much shorter) and/or FTN, depending on the parameters of interest. FTN contains the installation parameters for default control statement settings, control statement error processing, default file names, input/output buffer length, overlay library names, and reduce mode field length increments. The remaining parameters are in OPTIONS (FTNMAC/FTNTEXT).

4.2.11 COBOL 4.5

INSTALLATION PARAMETERS

The COBOL compiler uses symbol definitions from IPTEXT for IP.CMU, and MODEL. To override these installation parameter values, make the following changes in the COMDECK ASSEMOP when COBOL is assembled. (For jobs used to change installation parameters, refer to section 4.1.4.)

Generate code optimized for a CDC CYBER 74 type CPU	*B ASSEMOP.4 SKIP 1
Generate code optimixed for a CDC CYBER 73 type CPU	*I ASSEMOP.4 SXFR EQU 1
Generate CMU instructions	*D ASSEMOP.9,10
Generate non-CMU instructions	*D ASSEMOP.9

To create a compiler and object library which contain the CDCS interface feature activated, the COBOL compiler should be installed with the following code. During installation, this code should be available on file USER (refer to section 4.1.4). †

```
*IDENT CDCSIN
*INSERT,ASSEMOP.23
 DB1.0      EQU      1
```

4.2.12 CONVERSION AIDS SYSTEM 1.2

LANGUAGE CONVERSION AIDS SYSTEM (LCAS)

HARDWARE REQUIREMENTS

LCAS can be maintained on a 49K hardware configuration for NOS. If installation parameters other than the default parameters are chosen, more central memory is required.

DEFICIENCIES

1. The input must be valid, processable MASTER batch jobs and diagnostic-free source programs in compliance with the standard CDC 3000L product reference specifications.

2. A batch job cannot contain binary object decks or data.

3. Only FORTRAN and COBOL source can be processed in update mode.

4. Card data may not contain a $ in column 1 since this will be interpreted incorrectly as a MASTER control statement.

5. COBOL programs containing COPY statements can be processed only if the COPY library is made available during the conversion run.

6. Batch jobs and programs that have been partly converted or processed by the Language Conversion System cannot be submitted as input.

7. If a COBOL batch job using tapes or permanent files does not assign them through MASTER *DEF control statements, then the equivalent function to request the tapes or permanent files must be inserted manually into the converted control statement record for the job; otherwise, during job processing, only temporary scratch files are assigned.

8. COMPASS ($CMP) and tag sort ($LSRT) control statements, directives, and source decks will be deleted when input in job mode.

† The released program library for COBOL on REL5A does not contain this code. However, this code is included in the released binary.

INSTALLATION PARAMETERS

The tables of the FORTRAN and COBOL Language Conversion Processors (LCPs) may overflow while processing programs with large numbers of symbols or length statements.

The FORTRAN LCP name table contains a fixed-size entry for each name that appears in a declarative statement. The COBOL LCP name table contains a variable-size entry for each special name, file name, and data name, except within an RD, SD, or the Report Section. COBOL name table entries are 4+(n+3)/4 words long, where n is the number of characters in the name.

The user may reinstall the Conversion Aids System, enlarging these tables by running the maintenance and load jobs supplied on REL5B and including either *DEFINE, LTAB or *DEFINE, LTAB, XLTAB in the UPDATE directive record of the maintenance procedure where noted.

Table sizes are:

	No *DEFINE (default)	*DEFINE, LTAB	*DEFINE, LTAB, XLTAB
FORTRAN LCP	300 entries	600 entries	-----------------------
COBOL LCP	2500 words	5000 words	7500 words
Minimum Central Memory Required	74K words	104K words	114K words

FILE CONVERSION AIDS SYSTEM (FCAS)

HARDWARE REQUIREMENTS

FCAS can be maintained on a 49K hardware configuration for NOS.

DEFICIENCIES

1. FORTRAN File Conversion Processor (FCP)

   The FORTRAN FCP converts the amount of data transferred to an external medium by the processing of one FORTRAN input/output statement, according to the description and sequence which the user specifies, and so will not handle blocked records.

   Data organized on a bit basis (data created by the FORTRAN TABLE statement) will be converted on a word basis and may require a pack/unpack subroutine or function in the converted program.

   The maximum logical record length of the file to be converted is 33,999 6-bit characters.

2. COBOL File Conversion Processor (FCP)

   The maximum block size that can be handled is 131,067 characters.

   BCD files cannot contain computational binary items (COMP-1, COMP-2).

Processing of standard labels is accomplished via control statements.

Universal records are converted to type D records (the record contains a to b characters depending on dataname, where dataname is a 6-character, unsigned integer field located in the first word).

3. COSY UPDATE File Conversion Processor (FCP)

COSY/UPDATE converts sequential type COSY libraries only. COSY libraries residing on mass storage have to be copied to tape using the COSY COPY directive; they may not be transferred to tape in the original mass storage format.

4.2.13 SORT/MERGE 4.4

HARDWARE REQUIREMENTS

If the tape sort option is used, additional magnetic tape units are required (polyphase requires three and balanced requires four).

INSTALLATION PARAMETERS

Sort/Merge assembles to use the CMU hardware, depending upon IP.CMU from IPTEXT. (Refer to section 4.2.4 for definition of IP.CMU.) To override this parameter, make the following changes (refer to section 4.1.4 for jobs to change parameters).

To install Sort/Merge without CMU code:

```
*I  FEAT64.42
    BDP.INST     EQU      BDP.NO
```

To install Sort/Merge with CMU code:

```
*I  FEAT64.42
    BDP.INST     EQU      BDP.YES
```

4.2.14 ALGOL 4.1

DEFICIENCIES

A job must be in REDUCE(-) mode, since ALGOL programs use the space following the program as the buffer area or stack area.

INSTALLATION PARAMETERS

The following installation options are available. For jobs used to change installation parameters, refer to section 4.1.4.

1. The type of computer on which the compiler and run-time system must be assembled is established on the OLDPL by an UPDATE DEFINE directive placed in the YANK$$$ deck.

   The release deck contains the directives

   ```
   *IDENT MACHINE
   *DEFINE COMPUTER6
   ```

   in the maintenance decks for CDC CYBER 70/Models 72, 73, 74, CDC CYBER 170/Models 172, 173, 174, 175, and 6000 systems. The program library does not contain these directives.

2. A macro, DEFAULT, establishes which control error or compilation options are active by default. The release tape contains the following call.

   ```
   DEFAULT B,L,I,N,E,X
   ```

   To change these values, the parameters must be modified:

   ```
   *DELETE ALGO.119
       DEFAULT        new parameters
   ```

3. A macro, DEFAULT, establishes which run-time options are to be active by default. No options are default enabled on the release tape. To set run-time default options, introduce statements of the type:

   ```
   *INSERT OPENALG.129
       DEFAULT        new parameters
   ```

4. The default number of significant input characters is set by default to 72 but this value can be changed to a new value:

   ```
   *DELETE ALGO.83
   INPUT EOU        new value
   ```

   $1 \leq$ new value ≤ 126

   The K option of the ALGOL control statement may change that value dynamically.

4.2.15 BASIC 3.1

INSTALLATION PARAMETERS

For jobs used to change installation parameters, refer to section 4.1.4.

The default character set for BASIC 3 is the same as the default NOS character set. To create different versions of BASIC 3, use the following installation options.

Old character set

```
*DELETE LIPARAM.8
CHARSET        EQU      OLDCSET
```

NOTE

A corresponding change is required when installing the RESEQ processor. Refer to section 2.4.15 for a description of that change.

Print lines per inch default

```
*DELETE LIPARAM.2
IP.PD          CEQU     8          8 lines per inch
```

Printable lines per page default

```
*DELETE LIPARAM.3
IP.PS          CEQU     n          n lines per page
```

Burstable listing default

```
*DELETE LIPARAM.4
IP.BL          CEQU     1          Make listing burstable
```

AS flag default

```
*DELETE LIPARAM.5
IP.AS          CEQU     1          Make AS the default
```

Array base default

```
*DELETE BASCOMP.202
BDFLT          DATA     0          Make base 0
```

Time and memory utilization dayfile messages at compilation and execution time

```
*DELETE LIPARAM.9
MESSAG         EQU      1          Turn on CP time messages
```

4.2.16 APL 1.2

APL consists of five modules: APLSYS, APLB, APLT, APLTF, and APLTB. APLSYS is the overlay loader and is resident in the running system. APLB, APLT, APLTF, and APLTB are overlay files used to process batch and terminal input; they are direct access permanent files accessed under user number LIBRARY.

ADDITIONAL PROCEDURE

The jobs to install and modify APL (refer to section 4.1.4) must be run from the console (system origin job) to install APLB, APLT, APLTF, and APLTB as direct access permanent files on the usernumber LIBRARY, user index 377776. Care must be taken to ensure that these four permanent files are available on the production system under the usernumber LIBRARY.

The deck VAPL must also be run as a system origin job. This job initializes two indirect access files needed for APL.

4.2.17 SIMULA 1.1

DEFICIENCIES

All code released after level 401 should not be installed. Perform the following.

```
*YANK SU40013
*YANK SU40026
*YANK SU40039
*YANK SU40050
```

Code procedures and direct files are not implemented.

Jobs must be in REDUCE(-) mode when SIMULA programs are executed because the space following the program is used as the stack area for variables.

Segment mode loading (SIMULA control statement options S, U, R, and G) is deactivated; attempts to use these options result in job termination.

4.2.18 QUERY UPDATE 2.1

INSTALLATION PARAMETERS

Assembly options are defined within Query Update. At the time of release, the assembly options are set to values deemed most convenient or practical (for example, the default report page size is set at 136 columns by 60 lines). To obtain an up-to-date listing of the assembly options, run the following job.

```
Job statement
USER(usernumber,password,familyname)
VSN(REL11A=REL11A)
LABEL(REL11A,F=I,D=800)REL11A
UPDATE(P=REL11A,Q)
COPYSBF(COMPILE,OUTPUT)
7/8/9
*IDENT PARAM
*DEFINE KRONOS
*BEFORE CWEOR1.1
*DECK OPT
*CALL OPTIONS
*C OPT,CWEOR1
6/7/8/9
```

4.2.19 DATA DESCRIPTION LANGUAGE 1.1

No special information is needed to install Data Description Language.

4.2.20 CYBER DATABASE CONTROL SYSTEM 1.0

ADDITIONAL PROCEDURE

To activate the interface between CDCS and COBOL, refer to section 4.2.11.

4.2.21 DATABASE UTILITIES 1.0

No special information is needed to install Database Utilities.

4.2.22 QUERY UPDATE 3.0

No special information is needed to install Query Update 3.

4.2.23 DATA DESCRIPTION LANGUAGE 2.0

No special information is needed to install DDL 2.

4.2.24 FORM 1.0

DEFICIENCIES

Problems occur when executing FORM from within a load sequence, such as the following.

```
    .
    .
    .
LDSET(FILES=LIBNAB)
FORM.
    .
    .
    .
```

To overcome these problems, a copy of FORM must be assigned to the user's control point. This can be accomplished by the following control statements.

```
COMMON(SYSTEM)
GTR(SYSTEM,A)REL/FORM
DEFINE,FORM/CT=PUBLIC.
COPYEI(A,FORM,V)
```

It is suggested, however, that a site install FORM on the user number LIBRARY if this product will be used frequently in load sequences. There is no problem in executing FORM and no additional steps are necessary if it is not used in a load sequence.

4.2.25 MULTIMAINFRAME MODULE 1.0

No special information is needed to install Multimainframe 1. Refer to part III, section 8 for information concerning system operation in a multimainframe environment.

4.2.26 COBOL 5.0

DEFICIENCIES

The ANSI Communications Facility is not available in COBOL 5.

The CALL data-name/CANCEL Facility is not available in COBOL 5. (The CALL literal is available but the program cannot be CANCELLed.)

Most user programs written for COBOL 4 must be translated by Conversion Aids, COBOL 4 to COBOL 5 before they will compile and execute properly under COBOL 5.

The compiler supports only 64-character collating sequences.

INSTALLATION PARAMETERS

The COBOL 5 compiler uses symbol definitions from IPTEXT which are filtered through CB5TEXT. No direct references to any IPTEXT symbols are contained in the compiler or the object routines. This allows sites to be more flexible in changing their normal installation parameters for COBOL 5.

Symbols governing machine type, character set, and CMU option are obtained from IPTEXT, while those governing CDCS, default page size, print density, and default error termination must be changed within the product. To override the system defaults in one or more of these areas, the following changes should be made and placed during installation on file USER (refer to section 4.1.4).

Include the following code to create a compiler that will generate code for a CMU machine.

```
*INSERT CB5TEXT.134
 OP.BDP    CEQU   OP.YES
```

To change the default error termination level to T, W, F, or C, include the following.

```
*DELETE ASSEMOP.37
              DEF        CB5$ET       ≡level≡;
```

In this case, level may be specified as 1, 2, 3, or 4 to denote T, W, F, or C, respectively.

To activate CDCS processing, include the following; both deletions must be made or the results will be unpredictable.

```
*DELETE CB5TEXT.161
 OP.DCS     CEQU    OP.DCS1
*DELETE ASSEMOP.25
              DEF      CB5$CDCS          ≡"YES"≡;
```

To change the default number of lines per page (not including a total of 6 lines for top and bottom headers and margins), include the following.

```
*DELETE ASSEMOP.48
             DEF       CB5$LINP           ≡number≡;
```

To change the default print density, include the following.

```
*DELETE ASSEMOP.61
             DEF       CB5$PDENS          ≡number≡;
```

The acceptable values for number are 3, 4, 6, and 8 lines per inch.

To change the CPU type that code is generated for and object routines are assembled for, include the following.

```
*INSERT CB5TEXT.186
 OP.MODEL   CEQU   OP.machine
```

machine should be set to 6400 for a machine with a unified CPU and 6600 for a machine with a nonunified CPU.

4.2.27 CONVERSION AIDS, COBOL 4 TO COBOL 5 1.0

This conversion aid consists of two utility control processor programs which assist the user in converting COBOL 4 source programs to COBOL 5 source programs. The two utilities are a language conversion processor (LCP) and a COPY utility.

DEFICIENCIES

Input to the conversion aid must be valid diagnostic-free COBOL 4 source programs which comply with the COBOL 4 Reference Manual specifications.

Programs containing COPY (from library) statements can only be processed if the conversion aids library file from the COPY utility is available during conversion. Also, the COPY version of the LCP must be used.

*DECK directives are the only directives allowed in the Update source file.

INSTALLATION PARAMETERS

To generate the COPY version, specify *DEFINE CBLCOPY in the C4C5 job. This increases the central memory requirements by 20K.

The LCP may cause tables to overflow during the conversion of programs with large numbers of symbols or lengthy statements. Each name table entry contains a variable length entry of 4+(n+4)/10 words for each user defined name of n characters.

The LCP can be installed with enlarged tables by including LTAB or LTAB,XLTAB in the C4C5 job.

The name table sizes are as follows:

	No *DEFINE (default)	*DEFINE,LTAB	*DEFINE,LTAB,XLTAB
Name Table	3200	6500	14000
Minimum Central Memory Required	77K	84K	124K

SYSTEM ACCOUNTING 1

The user validation and user accounting capability of NOS is based on two special system files, VALIDUs and PROFILa.

VALIDUs is used to control user validation.

- Who can use the system
- What resources they can use
- Limits on job resource usage

PROFILa is used to control user accounting.

- Who can be accounted for
- What accounting parameters are assigned
- Limits on time of day access to the system
- Limits on total resource usage

As the structure of the user validation file and user accounting file change, concurrent file name changes may also occur to make it easier for sites to convert from one system to another. Although the file names VALIDUs, VALINDs, and PROFILa are referenced throughout this manual, the validation and accounting file names corresponding to the appropriate systems are listed in Table IV-1-1.

TABLE IV-1-1. VALIDATION AND USER ACCOUNTING FILE NAMES

System	User Validation File	User Indices File	User Accounting File
KRONOS 2.1.1	VALIDUX	VALINDX	PROFILO
NOS 1.0	VALIDUS	VALINDS	PROFILA
NOS 1.1	VALIDUZ	VALINDZ	PROFILA

VALIDUs contains user numbers and PROFILa contains charge-project numbers. User numbers identify the user, his set of permanent files, and his resource limitations. Charge-project numbers control and record billing charges. Entry of subsequent user numbers during a job session affects only permanent file usage and does not alter the billing procedure. Entry of subsequent charge-project numbers causes a new sequence of billing computations to be initiated.

The billing unit which reflects the resources used by the system during a job or a session is called the system resource unit (SRU). The calculation of this unit provides the flexibility of weighing the usage of resources against one another. The following resources are included in the calculation of this unit.

- CM field length
- ECS field length
- CPU usage
- Mass storage usage
- Magnetic tape usage
- Permanent file usage

The parameters for SRU computation are related to the charge-project numbers. The PROFILa file contains indexes used to determine which SRU parameters are to be used for computation while a charge-project number is in effect for the job or session.

1.1 USER VALIDATION (VALIDUs† — SPECIAL SYSTEM FILE)

A special system file is one which contains data necessary to control various aspects of system activity. (As a rule, this is privileged information requiring secrecy.) These files are maintained as direct access permanent files under the system user index 377777B. (User index is defined in section 2.1.) These files can only be accessed by special system jobs.

Special system jobs are system routines which can only be initiated by the analyst at his control point. They execute with benefit of file security and access to resources and system functions without user validation restrictions.

The system-bound security of VALIDUs and PROFILa† ensures that they will not be accessed by individual users either for curiosity or tampering.

System file security does not preclude the customer from establishing operational parameters for his own users. The analyst in creating and updating PROFILa can establish master users who may add, modify, and delete charge and project admissions for thier own people. The master users are customer supervisors who manage the projects involved and are in the best position to determine what should be used and by whom it should be used.

The VALIDUs file is created and managed by MODVAL. The PROFILa file is created and managed by PROFILE. These two managers are system program modules that, with input directives, constitute special system jobs used as system file processors. Special system jobs call the special file supervisor (SFS) which provides routines, table management, data manipulation, and I/O processing.

An input directive contains the parameters used to define access information. The basic format of each parameter designation is:

identifier = data

identifier	Specifies the control field within the file to be set
data	Specifies what value is to be set in that field

A brief overview of the creation and use of validation files is given in Figure IV-1-1.

---

† In this manual the user validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

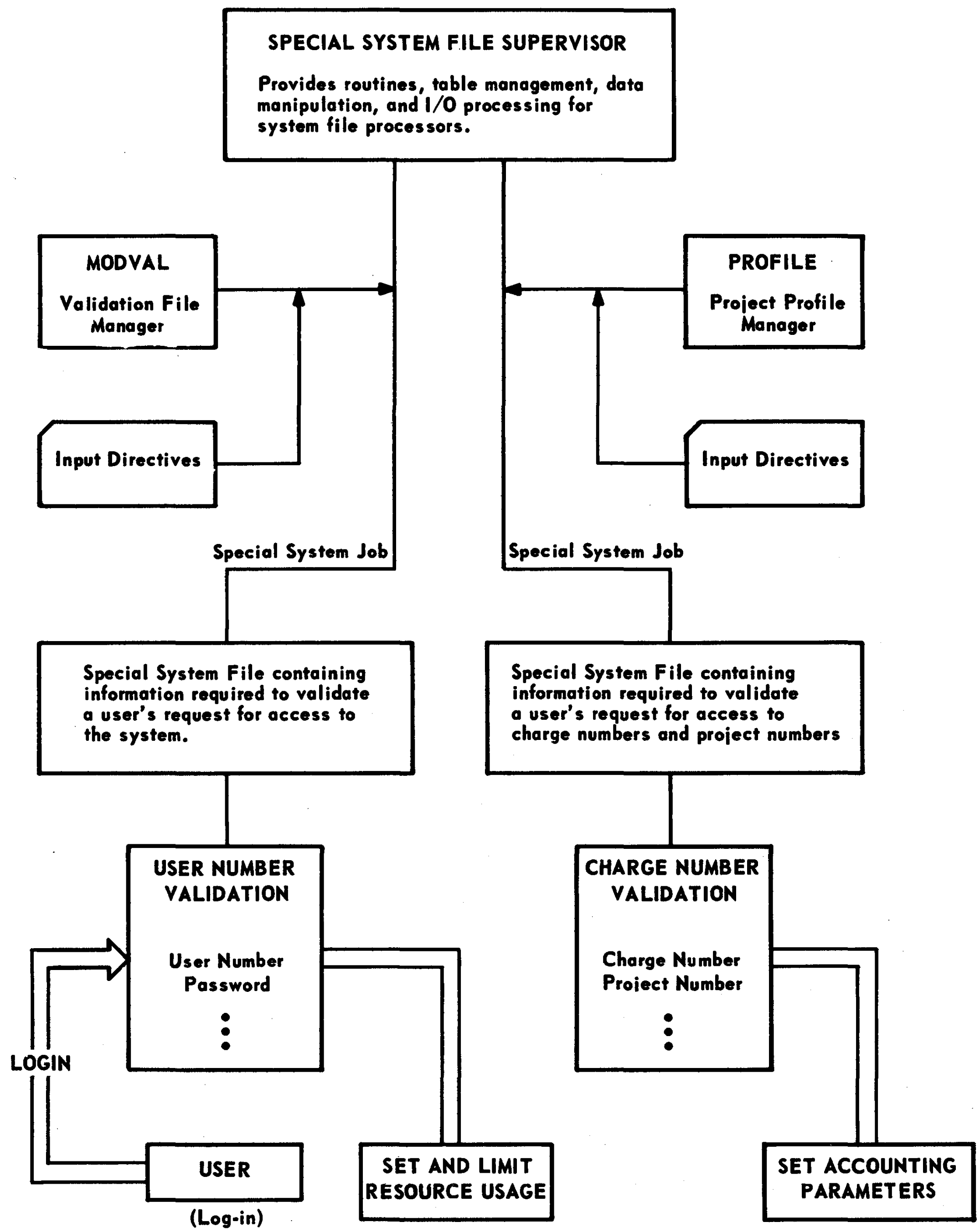


Figure IV-1-1. Creation and Use of Validation and Accounting Files

The VALIDUs special system file contains the user numbers that validate individual user access to the operating system. A second level of security, the password, can also be specified. Associated with each user number defined in VALIDUs is a set of limits which specify the system resources available to the user in a single job. This set includes the following.

- Equipment usage
  - Maximum number of magnetic tapes this user may use
  - Maximum number of disk packs this user may use
  - Maximum number of mass storage PRUs this user may additionally allocate
  - On what master device the permanent files and catalog for each user will be stored
  - Maximum number of lines printed or cards punched per output file
- File usage
  - Maximum number of permanent files allowed
  - Maximum number of files while running
  - Maximum number of disposed output (such as print or punch)
  - Ability to create direct access permanent files
  - Ability to create indirect access permanent files
  - Maximum size of direct or indirect access permanent files or cumulative size of all direct or indirect access permanent files
  - Maximum number of dayfile messages written, and control statements processed
- Machine usage
  - Maximum CPU time
  - Maximum CM space
  - Maximum ECS space
- System usage
  - System origin capability
  - May access system files
- Terminal usage
  - Which terminals are valid for individual users
  - Terminal type
  - Transmission mode
  - Parity
  - Delay count

The number of deferred batch jobs can also be specified.

1.1.1 MODVAL VALIDATION FILE MANAGER

The validation file manager, MODVAL, is used to create and manage the VALIDUs file. This can be done either from the system console (system origin job) or via a batch job.† Refer to section 1.1.7 (example 14) for a description of the creation of initial VALIDUs and VALINDs†† files.

VALIDUs is a direct access permanent file. From the console, it can be created and updated via the K display. Batch input must first establish local files and then copy onto the direct access file.

1.1.2 MODVAL CONSOLE INPUT

All batch input directives (section 1.1.3) are available for console input. In addition, the following are provided specifically for console input.

K.I,usernum	Inquire option relative to the user identified by the user number usernum. Information is given on the second MODVAL display (Figure IV-1-3). This option cannot be used for data entry.
K.U,usernum	Update modification for the user number usernum which is on the existing VALIDUs file. Modification data is input following the input of this directive.
K.C,usernum	The user number usernum is entered on a VALIDUs file that is being created. The following input line(s) can contain identifiers for this user number.
K.D,usernum	The user number usernum is deleted from the existing VALIDUs file.

Information for the previous options is provided on two MODVAL displays (Figures IV-1-2 and IV-1-3) for each user number specification. For the option types, identifiers must be entered on separate lines following the user number specification.

K./usernum,ident=data	This slash (/) directive is used to update usernum as with the U option; however, if usernum is not found, a new user number is created automatically. It is possible under this option to switch control directly from one user number to another without returning to the initial MODVAL display (Figure IV-1-2). If the / directive is used exclusively, data entry is analogous to batch input; that is, the card image can be entered with user number and identifier on the same line.
K.+	The plus (+) is used to toggle the user display pages (K display).

† While MODVAL may create the VALIDUs file from a batch job, the file thus created cannot be saved. Explanation of this restriction is given in section 1.1.3.

†† User validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

```
                          VALIDUs

                  CREATED      74/10/03.
                  UPDATED      75/04/13.


   INPUT DIRECTIVES ARE THE SAME AS BATCH INPUT DIRECTIVES.
   THE FOLLOWING DIRECTIVES ARE ALSO PROVIDED --

        /AN    - TERMINATE INPUT FOR PRESENT ACCOUNT NUMBER
                 IF ANY, AND UPDATE VALIDUS FILE.  INITIATE ACTION
                 ON *AN*.
        I,AN   - INQUIRE OPTION.  THIS DISPLAY ONLY.
        C,AN   - CREATE OPTION.  THIS DISPLAY ONLY.
        U,AN   - UPDATE OPTION.  THIS DISPLAY ONLY.
        D,AN   - DELETE OPTION.  THIS DISPLAY ONLY.
        +      - TOGGLE USER ACCOUNT DISPLAY PAGES.
        END    - COMPLETE UPDATE OR INQUIRE OF ACTIVE USER.
                 FROM THIS DISPLAY, END RUN.
        DROP   - TERMINATE INPUT FOR ACTIVE USER.
        STOP   - TERMINATE INPUT FOR ACTIVE USER, IF ANY.  END RUN.
            AN - 1-7 CHARACTER ACCOUNT NUMBER.

  DATA ENTRY FORMAT IS OF THE FORM MT=XX, PR=XX, TL=XX, ETC.
  ALL NUMERIC FIELDS ARE ASSUMED TO BE DECIMAL UNLESS A POST-
  RADIX IS SPECIFIED.  FOR EXAMPLE -
             4000B

  ACCESS WORD IDENTIFIERS STARTING AT BIT 0 ARE -
      CPWC CTPC CLPF CSPF CSOJ CASF CAND CCNR CSRP CSTP CTIM
```

Figure IV-1-2. First MODVAL Display

```
                 USER                 USER1        PAGE 1 OF 2

       CREATED     75/04/14.       UPDATED      75/04/14.

CONTENTS                                 DESCRIPTION

  PW  =                     PASSWORD (0-7 CHARACTERS).
  UI  =           0B        USER INDEX (1-377777B).
  MT  =            0        MAGNETIC TAPES (0-7).
  RP  =            0        REMOVABLE PACKS (0-7).
  MS  =           1B   (   4608 )     MASS STORAGE PRUS (0-77B).
  CM  =          14B   (   610B)      CENTRAL MEMORY FL (0-77B).
  EC  =           0B   (     0B)      ECS FIELD LENGTH (0-77B).
  TL  =           0B   (    10B)      TIME LIMIT (0-77B).
  NF  =            2   (     24 )     NUMBER OF LOCAL FILES (0-7).
  OF  =            1   (      4 )     DISPOSED OUTPUT (0-7).
  DB  =            0   (      0 )     DEFERRED BATCH (0-7).
  DS  =       SYSTEM                  DA FILE SIZE (0-7).
  FS  =       SYSTEM                  INDIRECT FILE SIZE (0-7).
  FC  =       SYSTEM                  PERMANENT FILE COUNT (0-7).
  CS  =       SYSTEM                  INDIRECT SPACE (0-7).

  AW  =  00000000000000000215          ACCESS WORD (3-4 CHARACTERS).
```

Figure IV-1-3. Second MODVAL Display (Page 1 of 2)

```
                    USER               USER1        PAGE 2 OF 2

        CREATED       75/04/14.     UPDATED      75/04/14.

  CONTENTS                                 DESCRIPTION

   LP  =              0B (    512)         LINES PRINTED (0-77B).
   CP  =              0B (      0)         CARDS PUNCHED (0-77B).
   CC  =             34B (    512)         CONTROL CARDS (0-77B).
   DF  =              0B (     64)         DAYFILE MESSAGES (0-77B).
   SL  =              0B (      8)         SRU LIMIT (0-77B).
   CN  =  CHG1                 CHARGE NUMBER (1-10 CHARACTERS).
   PN  =  PROJECTNUM           PROJECT NUMBER (1-20 CHARACTERS).


   AB  =  ANSWERBACK1          ANSWERBACK (0-10 CHARACTERS).
          ANSWERBACK2
          ANSWERBACK3
          ANSWERBACK4
   PA  =  EVEN                 TERMINAL PARITY (3-4 CHARACTERS).
   RO  =        SYSTEM         RUBOUTS (0-37B).
   PX  =  HALF                 TRANSMISSION (4 CHARACTERS).
   TT  =  TTY                  TERMINAL TYPE (3-7 CHARACTERS).
   TC  =  ASCII                CHARACTER SET (5-6 CHARACTERS).
   IS  =  NULL                 INITIAL SUBSYSTEM (4-8 CHARACTERS).

   AW  =  00000000000000000215         ACCESS WORD (3-4 CHARACTERS).
```

Figure IV-1-3. Second MODVAL Display (Page 2 of 2)

These five console options can be used by the analyst to access VALIDUs with the MODVAL manager according to the following general procedure.

1. The analyst types:

   A, B. CR

   X. MODVAL. CR

2. The B display will appear on the right screen. This display gives a listing of the control points and what is assigned to each. The analyst notes the control point to which MODVAL is assigned. To the right of this entry will appear the flashing message.

   REQUEST K DISPLAY

3. The analyst types in

   K, n. CR

   where n is the control point number noted on the B display.

4. The first MODVAL display will appear on the left screen. This is a listing of the options for manipulating the validation file VALIDUs (Figure IV-1-2).

5. The analyst types in one of the five console options. This is either

   K. option,usernum CR

   or

   K. /usernum, ident=data CR

   If a delete (K. D, usernum) is entered, the user number usernum is deleted from the validation file VALIDUs at this point. No further action is needed for this option.

6. For a create, update, or inquire option, the second MODVAL display (Figure IV-1-3) replaces the first on the left screen.

   For a create (either by C or /), the new user number will appear with default values for the parameters. If the ident is included with the slash (/), it will appear on the display but will not, at this point, be entered on the file.

   For an update (either by U or /), the existing user number will appear with current parameters. If a modification identifier is included with a / input, the new value will appear on the screen but will not be entered on the file. From here the analyst would have to go to step 7.

   For an inquire, the display contains the information requested and the procedure would stop at this step.

7. For a C or U option, the ident is now typed in with the format:

K. ident=data CR

8. To initiate action on the create or update entry, the analyst types:

END CR

If the analyst does not want this entry on the file, he can either type

DROP CR

and erase the entry without terminating this run, or he can type

STOP CR

and terminate the run without action on this entry.

In the case of DROP, the first MODVAL display (Figure IV-1-2) will return to the left screen and the analyst can enter more user numbers and their associated parameters.

To terminate any run without erasure, the operator types:

END. CR

Table IV-1-2 summarizes the basic input coding for console options.

TABLE IV-1-2. INPUT CODING FOR CONSOLE OPTIONS

Create	Update	Inquire	Delete
K. C, usernum K. ident=data K. END	K. U, usernum K. ident=data K. END	K. I, usernum K. END	K. D, usernum
K. /usernum, ident=data K. END			

1.1.3 MODVAL BATCH INPUT

Batch jobs that call the MODVAL validation file manager cannot make use of direct access permanent files, such as VALIDUs that are under the system user index 377777B. Accordingly, batch input to VALIDUs requires the user of user permanent files and local copies. The local versions are ultimately copied onto the direct access VALIDUs file via the console as a system origin job.

The following files are used by MODVAL in batch processing.

Default Name	Use
INPUT	Contains the input data directives that will be used to create or update the validation file VALIDUs
NEWVAL	The interim copy of the new validation file that is to be created or reformatted
VALIDUs	The old validation file that is to be updated or reformatted
SOURCE	Receives the source input for each user number
VALINDs	Contains all the available user indexes (definition in section 2.1) for the present VALIDUs file. is always used in conjunction with one of the validation files, new=NEWVAL or old=VALIDUs.
OUTPUT	File to receive output listings

For a batch create under MODVAL, two states of operation are necessary.

1. Input at the card reader is used by MODVAL to create a local copy of a new validation file. This copy is saved as an indirect or direct access permanent file.

2. From the console, the analyst gets or attaches the new validation file, defines the direct access file VALIDUs, and copies the new version onto the old direct access file.

For an update and other operations that deal with a preexisting VALIDUs file, three stages of operation are necessary.

1. The VALIDUs file is attached via the console. A local copy is made. The copy is saved, or defined and copied, as a permanent file.

2. The batch input gets or attaches this copy of VALIDUs. The MODVAL control statement and input parameters are executed. Either the local copy is modified and then replaced (indirect) or the modifications are entered on the attached file (direct).

3. From the console, the modified version is retrieved and then copied onto the direct access VALIDUs file.

While MODVAL batch input is more involved than input made exclusively at the console, it will prove faster and more convenient when a long list of user numbers with many identifiers will be entered.

MODVAL CONTROL STATEMENT

Batch input accesses the MODVAL validation file manager by means of the MODVAL control statement. The following is the format of the statement.

MODVAL(p_1, p_2, ..., p_i, ..., p_n)

The p_i's are specific combinations of the following identifiers.

I = input	Local file name of the file that will contain input data or source data; default is INPUT
P = validus	Local file name of the copy of the old validation file that is to be updated or reformatted; default is VALIDUs
N = newval	Local file name of the interim file that will become the newly created validation file; default is NEWVAL
S = source	File that will receive source data for each user number; default is SOURCE
U = valinds	File containing the available user indices of the current VALIDUs file; default is VALINDs
L = output	File to receive list output; default is OUTPUT
CV	KRONOS 2.1.1 to NOS 1 conversion (OP=C option), or NOS 1 to KRONOS 2.1.1 conversion (OP=S option). During a create run (OP=C), the automatic creation of the system and library user numbers is suppressed, and the creation of two user numbers with the same user index is allowed. Any identifier not recognized by NOS 1 MODVAL is ignored. During a source run (OP=S), the generation of identifiers not recognized by KRONOS 2.1.1 MODVAL is suppressed. **NOTE** When a KRONOS 2.1.1 VALIDUX file is converted to NOS 1 VALIDUs format that is subsequently converted to source, VALIDUs fields identified MT, RP, TL, CM, NF, MS, and DB are converted from range indices to actual values. These values may not be identical to the original VALIDUX file values.
D	If specified, MODVAL will not abort when directive errors are detected.
FA	Forces an attach of VALIDUs and VALINDs for system origin type jobs (for options OP=S, U, or R).
FN = name	name indicates the family name the user wishes MODVAL to access. To use this option, the user must have system origin privileges.

OP=C	Create option. Processes the input file and creates the interim validation file (N=NEWVAL) and the file of associated user indices (U=VALINDs).
OP=C, LO=E	Initiates the create as above, and then lists errors encountered in processing (refer to Error Messages to Output File, section 1.1.7).
OP=U	Update option. Updates the local copies of VALIDUs and VALINDs with data on input file. This option can be used with certain other options (for example, OP=URS). It is the default for jobs other than system origin jobs.
OP=U, LO=E	Initiates the update as above, and then lists the errors encountered in processing (refer to Error Messages to Output File, section 1.1.7).
OP=Z	Statement update option. This is like the update option except that directives are included on the MODVAL statement. The Z parameter in this option must be used alone.
OP=Z, LO=E	Initiates the statement update as above, and then lists the errors encountered in processing (refer to Error Messages to Output File, section 1.1.7).
OP=R	Reformats the validation file by purging all files of each deleted user. Until this option is selected, all files of deleted users remain in the permanent file system even though they cannot be accessed. This allows redefinition of a user (with UI identifier on data input directive) if an error was made in deleting him.
OP=S	Specifies a source run that returns the validation file specified by the P identifier (default=VALIDUs) to source format (directive images) on the file specified by the S keyword (default=SOURCE).
OP=K	K display option. All other options are cleared and instructions must be entered via the K display. K is the default value for a system origin job. Hence, if no parameters are specified and the call statement is simply MODVAL. the K option is automatically selected.

OP=I

Inquire option. Gives a listing of validation parameters for the user specified by the last USER statement or by terminal login. This option must occur alone. The following is a sample list.

```
  AB =,
  AB =,
  AB =,
  AB =,
MT =                 2,
RP =                 3,
TL =              1290,
CM =               730,
NF =                18,
DB =                 8,
FC =                16,
CS =              2560,
FS =                16,
PA = EVEN             ,
RO =            SYSTEM,
PX = HALF             ,
TT = TTY              ,
TC = STANDARD         ,
IS = NULL             ,
MS =         UNLIMITED,
DF =               298,
CC =               266,
OF =                16,
CP =               522,
LP =              4106,
EC =                 0,
SL =              1034,
 CN =73CD65,
 PN =KN               ,
DS =                16,
AW = 00000000000000000215
```

OP=L
or
OP=L,LO=A

Reads the validation file, sorts the copy by user number, and writes it to the output file for listing according to the format in Figure IV-1-4.

OP=L,LO=N

Reads the validation file, sorts the copy by user index, and writes it to the output file for listing according to the format in Figure IV-1-4.

OP=L, LO=L	Reads the information on the local file identified in the parameter list, sorts by user number, and writes it to the output file for listing according to the format in Figure IV-1-4.
OP=L, LO=AL	Same as LO=L since A is a default value
OP=L, LO=NL	Reads the information on the local file identified in the parameter list, sorts the copy by user index, and writes it to the output file for listing according to the format in Figure IV-1-4.
OP=L, LO=EN or OP=L, LO	File will be sorted by user index.
OP=C, LO=EN U Z or OP=C, LO U Z	Produces a list of errors for the C, U, or Z processing. In this case, MODVAL will use whichever applies.

```
USER       USER     CREATION  LAST MOD
NUMBER     INDEX      DATE      DATE

USERAAA        1    yy.mm.dd. yy.mm.dd.
USERBBB        2    yy.mm.dd. yy.mm.dd.
USERCCC        3    yy.mm.dd. yy.mm.dd.
   .        .          .         .
   .        .          .         .
   .        .          .         .
LIBRARY   377776    yy.mm.dd. yy.mm.dd.
```

Figure IV-1-4. Format of VALIDATION File Listing

INPUT DIRECTIVE

The input directive is used to enter user numbers under a create run (OP=C) and to modify existing user numbers under an update run (OP=U). Format of the input directive is

/usernum, ident1=data1, ident2=data2, ...

where usernum is the one- to seven-character user number being referenced and each ident=data is a system usage definition for this number. The user number statement must begin with a / in column 1. The user number and all directives must be terminated by a separator.

In addition to the standard separators (period, comma, parenthesis, dollar-sign, etc.), end-of-card and end-of-line are recognized. Thus, if an input directive item occurs last on a statement, none of the standard separators are needed.

All directives relative to a user number must appear before another user number or the end of the input stream is encountered.

All data within a user number entry is free format to column 72. A directive cannot be split between cards or lines. Blanks are ignored.

To allow sequencing and/or identification of input directives, all data past column 72 is ignored.

Example: The following is acceptable

/ROBERTR, AW=CSPF

AW=CLPF

However, data cannot lap from statement to statement.

Example: The following is not acceptable.

/ROBERTR, AW=CSPF, AW=

CLPF.

The following is a list of identifiers and their descriptions.

Identifier	Description
PW = passwrd	passwrd is a one-to seven-character alphanumeric password. Blanks are significant. If this identifier is omitted, the system assigns a password of all blanks. In the latter case, the user will have to enter a null password at login.
UI = nnnnnnr	This identifier specifies the user index to be assigned to this user. If this entry is not supplied, the system assigns the next available user index. nnnnnnr consists of six numeric characters followed by a radix. Blanks are suppressed. The maximum value is 377777B. This identifier cannot be used with the K display or update option.

Identifier	Description
AB = answerback	answerback is a 1- to 10-character answerback code. Blanks are significant. The answerback code restricts the user to a particular terminal. Up to four answerback entries are permissible per user number. If this identifier is omitted, the system supplies an answerback code of all blanks which gives this user access through any terminal.
MT = n	This identifier specifies the number of magnetic tapes allowed. n consists of one numeric character. The maximum value is 7, which specifies unlimited tapes. If this identifier is omitted, the system supplies a value of 0.
RP = n	This identifier specifies the number of removable disk packs allowed. n consists of one numeric character. The maximum value is 7, which specifies unlimited disk packs. If this identifier is omitted, the system supplies a value of 0.
TL = nnr	This identifier specifies a range index representing the maximum CPU time that a user may run. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum index value is 77B, which specifies unlimited CPU time. If this identifier is omitted, the system supplies an index value of 0. The system utilizes the formula maximum CPU time = (index * 100B) + KTLI† to convert the index value into the maximum CPU time (units of 10B seconds) that a user may run.
DF = nnr	This identifier specifies a range index representing the maximum number of MESSAGE requests the user can issue to the system and/or job dayfiles. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum index value is 77B which specifies unlimited MESSAGE requests. If this identifier is omitted, the system supplies an index value of 0. The system utilizes the formula maximum MESSAGE requests = (index * 20B) + KDFI† to convert the index value to an actual limit of job MESSAGE requests.
CC = nnr	This identifier specifies a range index representing the maximum number of batch control statements processed for a user. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum index value is 77B which specifies unlimited batch control statement processing. If this identifier is omitted, the system supplies an index value of 34B. The system utilizes the formula maximum control statements = (index * 20B) + KCCI† to convert the index value to an actual limit of control statements processed.

† Refer to part III, section 2.4.2 for a description of these COMSACC parameters.

Identifier	Description
OF = n	This identifier specifies a range index representing the maximum number of job print and punch files the user can dispose to the output queue. n consists of one numeric character. The maximum index value is 7, which specifies an unlimited number of disposed output files. If this identifier is omitted, the system supplies an index value of 1. The system utilizes the formula maximum disposed output=(index * 4) to convert the index value to an actual limit of disposed output files.
CP = nnr	This identifier specifies a range index representing the number of cards that can be punched from a user's disposed punch file. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B, which specifies unlimited punched output. If this identifier is omitted, the system supplies an index of 0. The system utilizes the formula cards punched = (index * 100B) + KCPI† to convert the index value to an actual limit of cards punched from a disposed file.
LP = nnr	This identifier specifies a range index representing the number of lines that can be printed from a user's disposed print file. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B, which specifies unlimited print output. If this identifier is omitted, the system supplies an index value of 0. The system utilizes the formula lines printed = (index * 2000B) + KLPI† to convert the index value of an actual limit of lines printed from a disposed output file.
EC = nnr††	This identifier specifies a range index representing the maximum ECS memory space a user is allowed. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B, which specifies all ECS memory space of the machine. If this identifier is omitted, the system supplies an index value of 0. The system utilizes the formula ECS limit = (index * 100B) + KECI† to convert the index value to an ECS memory limit expressed in units of 2000B words.
SL = nnr	SRU limit. Refer to section 1.4 for information concerning SRU.

† Refer to part III, section 2.4.2 for a description of these COMSACC parameters.

†† This feature is not currently utilized by the system but is provided for future expansion of validation control.

Identifier	Description
CM = nnr	This identifier specifies a range index representing the maximum central memory space a user is allowed. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B, which specifies all available CM space of the machine. If this identifier is omitted, the system supplies an index value of 14B. The system utilizes the formula CM limit = (index * 40B) + KCMI† to convert the index value to a CM space limit expressed in units of 100B words.
NF = n	This identifier specifies a range index representing the maximum number of files a user can concurrently utilize. n consists of one numeric character. The maximum index value is 7, which specifies an unlimited number of concurrent files. If this identifier is omitted, the system supplies an index of 2. The system utilizes the formula concurrent files = (index * 10B) + KNFI† to convert the index value to an actual limit of concurrent files.
MS = nnr	This identifier specifies a range index representing the maximum number of mass storage PRUs the user is allowed to additionally allocate. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B which specifies unlimited additional mass storage PRUs. If this identifier is omitted, the system supplies an index value of 1. The system utilizes the formula PRU limit = (index * 10000B) + KMSI† to convert the index value to a PRU equivalent of the actual mass storage tracks additionally allocated to the job files.
DB = n	This identifier specifies a range index representing the maximum number of deferred batch jobs the user is allowed to have in the system concurrently. The maximum index value is 7, which specifies an unlimited number of deferred batch jobs. If this identifier is omitted, the system supplies an index value of 0. The system utilizes the formula deferred batch jobs = (index * 2) to convert the index value to an actual limit of deferred batch jobs. If the user has the access bit CSOJ and DEBUG is set on the system display console, the user has system origin,or DB=0, the DB parameter is ignored and the user may submit as many jobs as he wishes.

† Refer to part III, section 2.4.2 for a description of these COMSACC parameters.

Identifier	Description
AW = xxxx	xxxx is a four-character designation that toggles a particular bit in the access word. For each bit that is set, special permission is allowed to that user. The bit is set when the first identifier is encountered and cleared if the identifier is used again. A maximum of 60 entries per record is allowed. Blanks are suppressed.

There are 11 access bits defined in the system.

• CPWC (bit 0)	User may change his password
• CTPC (bit 1)	User may use the ACCESS commands (terminal use only)†
• CLPF (bit 2)	User may create direct access permanent files
• CSPF (bit 3)	User may create indirect access permanent files
• CSOJ (bit 4)	User may have system origin capability from any job origin if the debug option is turned on by the operator User may also assign a device by specifying its EST ordinal. This does not require that the debug option be turned on. User may also call the PPU/hardware diagnostics of the 881/883 pac reformatting utility FORMAT, if ENGINEERING mode is enabled.
• CASF (bit 5)	User may access system files (library)
• CAND (bit 6)	User may request nonallocatable devices (for example, magnetic tape units)
• CCNR (bit 7)	Allows use of system without entry of charge or project number
• CSRP (bit 8)	User may issue auxiliary device commands
• CSTP (bit 9)	User may access special transaction functions
• CTIM (bit 10)	User is not logged off because of timeout

†Refer to appendix E of the operator's guide for a description of the ACCESS commands.

Identifier	Description
	The default values are CPWC, CLPF, CCNR, and CSPF. To set or clear all bits in the access word, the following commands can be entered. • ALL — Sets all 60 bits in the access word • NUL — Clears all 60 bits in the access word
CAB = oldab, newab	Indicates that the answerback code is to be changed. The entry consists of two fields: the first (oldab) indicates the answerback code that is to be changed and the second (newab) indicates the new code. An entry consists of one to ten alphanumeric characters. Blanks are not suppressed. Four or less answer-back changes are permitted per record.

The following identifiers can only be used in update and K display options.

Identifier	Description
DAC= usernum	Deletes the user number, usernum, from the VALIDUs file. This user number must match the current user number as specified after the most recent /.
FUI=nnnnnnr	Forces the user index to be inserted or changed. Parameters are the same as for the UI=nnnnnnr.

The following four specifications control permanent file access for the individual user. Ordinarily this is specified by origin type.

Identifier	Description
FC=n	File count indicating the maximum number of indirect access permanent files allowed the user. n may assume values in the ranges shown. The user is validated for the corresponding upper limit.

n	Upper Limit Allowed (Octal)
0	Use job origin control
1	10
2	20
3	30
4	40
5	50
6	100
7	Unlimited

If FC is not specified, n=0 will be assumed.

Identifier	Description
CS=n	Cumulative size of all indirect access files for this user. n may assume any value. The user is validated for the corresponding upper limit.

n	Upper Limit Allowed (Octal Count of PRUs)
0	Use job origin control
1	1000
2	2000
3	5000
4	10000
5	50000
6	100000
7	Unlimited

If CS is not specified, n=0 is assumed.

Identifier	Description
FS=n	Maximum file size allowed for an individual indirect access permanent file. n may assume the values listed. The user is validated for the corresponding upper limit.

n	Upper Limit Allowed (Octal Count of PRUs)
0	Use job origin control; no controls are enacted
1	10
2	20
3	30
4	40
5	50
6	60
7	Unlimited

If FS is not specified, n=0 will be assumed.

Identifier	Description
DS = n	File size allowed for an individual direct access permanent file. n may assume the following values.

n	Upper Limit Allowed (Octal Count of PRUs)
0	Use job origin control
1	1000
2	2000
3	5000
4	10000
5	50000
6	100000
7	Unlimited

If DS is not specified, n=0 is assumed.

The following six specifications manipulate fields describing the user's terminal.

Identifier	Description
PX = xxxx	Specifies transmission mode. Only one entry should occur per user number record. Since the terminal operates in full or half duplex mode, either of the following values is available for xxxx.

Value	Explanation
FULL	System enters echoplex mode automatically.
HALF	System does not enter echoplex mode automatically.

Identifier	Description
RO = nnr	nnr consists of two numeric characters followed by a radix. Blanks are suppressed. This is the rubout count which is the character count delay associated with the user's terminal. One value from 0 to 37B may be entered for each user number record. A value of 37B denotes that the system will use the default number for the user's terminal type.
PA = xxxx	xxxx specifies terminal parity. The terminal operates with even or odd parity. One of the following two values may be entered for each user number record.

EVEN

ODD

Identifier	Description
TT = xxxxxx	xxxxxx specifies the terminal type. One of the following values may be specified for each user number.

Value	Explanation
TTY	Teletype or other ASCII compatible terminal
MEMAPL	Memorex 1240 with APL print
COR	Correspondence with standard print
CORAPL	Correspondence with APL print
BLKEDIT	Block mode terminal
713	CDC 713 display terminal

Identifier	Description
TC = xxxxx	xxxxx specifies the character set to be used by the terminal. One of the following values is available for each user number.

Value	Explanation
STANDARD	63- or 64-character code set
ASCII	128 full ASCII code set

Identifier	Description
IS = xxxxx	xxxxx specifies the initial subsystem for the terminal. One of the following values may be specified for each user number.

Value	Explanation
NULL	NULL subsystem
BASIC	BASIC subsystem
FTNTS	FORTRAN subsystem
TRANSACT	TRANSACT subsystem
EXECUTE	EXECUTE subsystem
BATCH	BATCH subsystem

Identifier	Description
CN = chargenumber	This identifier specifies the charge number associated with the user. This field is not checked or updated by the system; it serves only to provide information when OP=I is specified. To maintain correct information, this parameter should be updated when the user's charge number information is updated.
PN = projectnumber	This identifier specifies the project number associated with the user. This field is not checked or updated by the system; it serves only to provide information when OP = I is specified. To maintain correct information, this parameter should be updated when the user's project number information is updated.

1.1.4 EXAMPLES OF MODVAL USAGE

The examples in this section give representative commands for exercising the MODVAL options both at the console and by batch input. System files are under index 377777B. Refer to section 1.1.8 for the definition of the ISF control statement.

Example 1:

Example 1 is a create at the console with the C,usernum format. MODVAL is called and the B display indicates the control point for the job (n). This is entered via the K display. Following this is an entry of three user numbers with a password ident for each.

X. MODVAL.

K,n.

K. C. USER201

K. PW=ADMIT1

K. END

K. C, USER202

K. PW=ADMIT2

K. END

K. C, USER203

K. PW=ADMIT3

K. END

K. END.

Example 2:

In example 2, the previous parameters are entered at the console with the / format.

X. MODVAL.

K, n.

K. /USER201, PW=ADMIT1

K. /USER202, PW=ADMIT2

K. /USER203, PW=ADMIT3

K. END

K. END.

Example 3:

In example 3, the same entries are made as a batch job with default values used for the file names. The following is the statement input.

Job statement

USER(usernumber, password, familyname)

MODVAL(OP=C)

SAVE(NEWVAL)

SAVE(VALINDs=VAL)

7/8/9

/USER201,PW=ADMIT1

/USER202,PW=ADMIT2

/USER203,PW=ADMIT3

6/7/8/9

This produces indirect access permanent files. These will be made direct access permanent files in the system from the console. The system index (377777) must be specified. The third parameter on the COPY command initiates verification of the files after execution of the copy. Input at the console is:

X.DIS

USER(usernumber,password,familyname)

or SUI,xxxx. where xxxx is the user index for usernumber.

GET,NEWVAL.

GET,VAL.

SUI,377777.

If a VALIDUs file already exists on the system, it will be necessary, at this point, to enter

ISF(R=VALIDUs).

PURGE(VALIDUs, VALINDs).

DEFINE,VALIDUs.

DEFINE,VALINDs.

COPY,NEWVAL,VALIDUs.

COPY,VAL,VALINDs,V.

Example 4:

In example 4, the previous create is run with file names supplied. It is assumed that the following indirect access file is on mass storage before the batch deck is submitted.

FILE PUTIN

/USER201,PW=ADMIT1

/USER202,PW=ADMIT2

/USER203,PW=ADMIT3

Input at the card reader is:

Job statement

USER(usernumber, password, familyname)

GET, PUTIN.

MODVAL,OP=C,I=PUTIN,N=VALNEW.

SAVE,VALNEW.

SAVE,VALINDS=VALX.

6/7/8/9

After this job is executed, the following entries are made at the console.

X.DIS.

USER (usernumber, password, familyname)

or SUI,xxxx, where xxxx is the user index for usernumber.

GET,VALNEW.

GET,VALX.

SUI,377777.

If a VALIDUs file already exists on the system, it will be necessary, at this point, to enter

ISF(R=VALIDUs).

PURGE(VALIDUs,VALINDs).

DEFINE, VALIDUs.

DEFINE,VALINDs.

COPY,VALNEW,VALIDUs,V.

COPY,VALX,VALINDs,V.

Example 5:

Example 5 is an update at the console with the U,accnumb format. The first two user numbers entered via the previous creates have their passwords changed.

X.MODVAL.

K,20.

K.U,USER201

K.PW=ENTER1

```
K.END

K.U,USER202

K.PW=ENTER2

K.END

K.END.
```

Example 6:

In example 6, the previous parameters are entered at the console with the / format.

```
X.MODVAL.

K,20.

K./USER201,PW=ENTER1

K./USER202,PW=ENTER2

K.END

K.END.
```

Example 7:

In example 7, the previous update is entered by means of batch input. First, the direct access permanent files VALIDUs and VALINDs are copied to permanent files (direct or indirect) that can be accessed by the batch input and used in the MODVAL control statement. This is done at the console as follows:

```
X.DIS.

SUI,377777.

ISF(R=VALIDUs)

ATTACH(VALIDUs,VALINDs)

COPY(VALIDUs,VAL)

COPY(VALINDs,VALX)

RETURN(VALIDUs,VALINDs)

ISF.

USER(ANLST)

SAVE(VAL,VALX)
```

The batch input is:

```
Job statement

USER(usernumber,password,familyname)

GET(VAL,VALX)

MODVAL(OP=U,P=VAL,U=VALX)

REPLACE(VAL,VALX)

7/8/9

/USER201,PW=ENTER1

/USER202,PW=ENTER2

6/7/8/9
```

The modified files are returned to the system at the console. The M=W in the ATTACH is needed to establish write permission relative to the direct access files.

```
X.DIS

USER(usernumber,password,familyname)

GET(VAL,VALX)

SUI,377777.

ISF(R=VALIDUs)

ATTACH(VALIDUs,VALINDs/M=W)

COPY(VAL,VALIDUs)

COPY(VALX,VALINDs)

ISF.
```

If the OP=Z option is used, it is not necessary to provide an input file and SAVE it under 377777. The Z option makes the changes directly as follows:

```
X.DIS.

MODVAL(OP=Z)/USER201,PW=ENTER1

     (one user at a time)
```

Example 8:

In example 8, a delete is done from the console only.

```
X.MODVAL

K,20.

K.D.USER203

K.END.
```

Example 9:

In example 9, reformatting of the validation file is initiated from the console. The DIS is used and statements are typed in (no K display). The OUTPUT file will have a listing of the purged indices.

```
X.DIS.

SUI,377777.

ISF(R=VALIDUs)

ATTACH(VALIDUs,VALINDs/M=W)

MODVAL(OP=R)

REWIND(VALIDUs,NEWVAL)

COPY(NEWVAL,VALIDUs)

OUT.

ISF.
```

Example 10:

In example 10, to reformat the validation file with batch input, the direct access files have indirect access copies made via the console.

```
X.DIS.

SUI,377777.

ISF(R=VALIDUs)

ATTACH(VALIDUs,VALINDs)

COPY(VALIDUs,VAL)

COPY(VALINDs,VALX)

RETURN(VALIDUX,VALINDs)
```

```
ISF.
USER(ANLST)
SAVE(VAL,VALX)
```

Then, from the card reader:

```
Job statement
USER(usernumber,password,familyname)
GET(VAL,VALX)
MODVAL(OP=R,P=VAL,U=VALX)
SAVE(NEWVAL)
REPLACE(VALX)
6/7/8/9
```

Then, from the console:

```
X,DIS.
USER(usernumber,password,familyname)
GET(NEWVAL,VALX)
SUI,377777.
ISF(R=VALIDUs)
ATTACH(VALIDUs,VALINDs/M=W)
COPY(NEWVAL,VALIDUs,V)
COPY(VALX, VALINDs, V)
ISF.
```

Instead of the ATTACH, in which each COPY would write over an old file, it would be possible to use

```
PURGE(VALIDUs,VALINDs)
DEFINE(VALIDUs,VALINDs/M=W)
```

and then copy onto the empty files.

Example 11:

In example 11, the validation file is returned to source code via the console.

```
X.DIS.

SUI,377777.

ISF(R=VALIDUs)

ATTACH(VALIDUs,VALINDs)

MODVAL(OP=S)

SAVE(SOURCE)
```

Later, this source code file could be used to create a new VALIDUs file with:

```
GET(SOURCE)

MODVAL(OP=C,I=SOURCE)
```

Example 12:

In example 12, to return the validation file to source code via batch, the following procedure is followed.

From the console:

```
X.DIS.

SUI,377777.

ISF(R=VALIDUs)

ATTACH(VALIDUs,VALINDs)

COPY(VALIDUs,VAL)

COPY(VALINDs,VALX)

RETURN(VALIDUs, VALINDs)

ISF.

ATTACH(VALIDUs,VALINDs)

COPY(VALIDUs, VAL)

COPY(VALINDs,VALX)

USER(ANLST)

SAVE(VAL, VALX)
```

From the card reader:

Job statement

USER(usernumber,password,familyname)

GET(VAL,VALX)

MODVAL(OP=S, P=VAL, U=VALX)

SAVE(SOURCE)

From the console:

X.DIS.

USER(usernumber,password,familyname)

GET(SOURCE)

SUI,377777.

SAVE(SOURCE)

Example 13:

In example 13, a KRONOS 2.1.1 VALIDUX file is converted to NOS 1 format. The analyst first deadstarts a KRONOS 2.1.1 system and then enters the following sequence of commands at the console.

X.DIS.

SUI(377777)

DEFINE(SOURCE)

MODVAL(OP=S,P=VALIDUX,FA)

DROP.

When the VALIDUX file is successfully converted to source, the analyst should deadstart a NOS 1 system. After bringing up a DIS package, the following sequence of commands should be entered.

X.DIS.

SUI(377777)

ATTACH(INPUT=SOURCE)

PURGE(VALIDUs,VALINDs)

DEFINE(VALIDUs,VALINDs)

```
MODVAL(OP=C,N=VALIDUs,CV)

DROP.

X.ISF.
```

The CV parameter allows use of a source dump of VALIDUX to create a new NOS VALIDUs file. KRONOS 2.1.1 identifiers not recognized by NOS MODVAL are ignored when the VALIDUs file is created and are diagnosed as errors in the output file listing.

If it is desired to reverse the conversion sequence (that is, starting from NOS and converting to 2.1.1), the CV parameter should be used on the NOS MODVAL call to SOURCE to suppress NOS source identifiers not recognizable by KRONOS 2.1.1 MODVAL. After a NOS deadstart, the following sequence of commands should be entered.

```
X.DIS.

SUI(377777)

DEFINE(SOURCE)

MODVAL(OP=S,CV,FA)

DROP.
```

Then, the analyst should deadstart a KRONOS 2.1.1 system and enter the following commands.

```
X.DIS.

SUI(377777)

ATTACH(INPUT=SOURCE)

PURGE(VALIDUX,VALINDX)

DEFINE(VALIDUX,VALINDX)

MODVAL(OP=C)

SAVE(NEWVAL=VALIDUX)

DROP.

X.ISF.
```

Example 14:

In example 14, a VALIDUs file and a VALINDs file are created when there are no VALIDUs and VALINDs files already present; that is, an initial VALIDUs file and an initial VALINDs file are created.

The local file PUTIN contains input directives for three user numbers.

```
/USER201,PW=ADMIT1

/USER202,PW=ADMIT2

/USER203,PW=ADMIT3
```

The following are entered at the console.

```
X.DIS

SUI(377777)

DEFINE,VALIDUs,VALINDs

MODVAL(I=PUTIN,N=VALIDUs,OP=C)

RETURN,VALIDUs,VALINDs.

ISF.
```

1.1.5 PASSWOR CONTROL STATEMENT

If the access word (section 1.1.3) for a particular user has the zero bit set (AW=CPWC), this user may change his password by using the PASSWOR control statement. The format of the statement is:

PASSWOR (oldpass,newpass)

The passwords are one to seven characters.

PASSWOR ERROR MESSAGES

ERROR IN PASSWOR ARGUMENTS.

Invalid control statement arguments.

Action: Correct and rerun.

ILLEGAL CONTROL CARD.

User is not permitted to change password.

Action: User must acquire validation.

1.1.6 LIMITS CONTROL STATEMENT

Each user can obtain a listing of all the limiting parameters that apply to his user number by means of the LIMITS control statement. MODVAL processes this statement exactly as the OP=I option. The format of the statement is:

LIMITS.

The explanation of the inquire option under MODVAL Control Statement (section 1.1.3) gives a sample listing.

LIMITS ERROR MESSAGE

ERROR IN LIMITS ARGUMENTS.

Invalid control statement arguments. This statement has no arguments associated with it.

1.1.7 DIAGNOSTICS FOR MODVAL

DAYFILE ERROR MESSAGES

DATA BASE ERROR.

Error in a control word in the validation file is in error. If error persists, call an analyst.

SYSTEM ERROR.

Internal malfunction due to either software or hardware. Consult an analyst immediately.

ILLEGAL PASSWORD

Old password not found. Correct and rerun.

ERROR IN MODVAL ARGUMENTS.

Invalid control statement arguments. Correct and rerun.

MODVAL ABORTED.

Control point error flag is set. Consult dayfile listing for reason.

DIRECTIVE ERRORS.

Errors were encountered on the input file for either a create or an update. Check the output file for specific errors.

ILLEGAL CONTROL CARD.

User is not validated to change password via PASSWOR statement.

ERROR MESSAGES TO OUTPUT FILE

Corrective action is governed by the severity of the errors. The job may be rerun or the new validation file corrected at the user's discretion.

**** ERROR IN USER NUMBER.

Illegal data was encountered where user number was expected.

Action: Illegal data is disregarded and MODVAL goes to the next user entry.

**** DUPLICATE USER NUMBER.

The user number encountered is a duplicate of a user number previously entered. This can only happen on a create run.

Action: The first entry is used.

**** ERROR IN IDENTIFIER.

Illegal parameter identifier encountered.

Action: That particular user number is disregarded. If entry is from K display, only that line of input is disregarded.

**** ERROR IN NUMERIC DATA.

Indicates any of the following:

1. Numeric data not numeric
2. Data exceeds maximum
3. No data present

Action: This user number entry is disregarded. If entry is from K display, only that line of input is disregarded.

**** ERROR IN ALPHANUMERIC DATA.

Indicates any of the following:

1. No data present
2. Data for AW identifier unrecognized
3. Number of characters exceeds maximum allowed

Action: This user number entry is disregarded. If entry is from K display, only that line of input is disregarded.

**** USER INDEX PREVIOUSLY DEFINED.

No more than one user number can be assigned to any user index with the UI identifier.

Action: This user number entry is disregarded unless the CV parameter (suppression of automatic creation of system and library user indices) is selected. Then the duplication is flagged on the output file and processing continues normally.

**** KRONOS INPUT DIRECTIVE IGNORED.

KRONOS input identifier (MS or PR) was encountered during a MODVAL (CV,OP=C) create run.

Action: PR identifier is ignored and/or the MS field of the user number record is set to its default value.

**** NO USER INDICES AVAILABLE.

No more user indices are available for automatic assignment. The FUI directive (force user index to be inserted or changed) must be used to specify user indices.

Action: This user number is disregarded. If entry is from K display, only that line of input is disregarded.

INFORMATIVE MESSAGES

CATALOG COMPLETE.

A list run is complete.

nnn USER INDICES PURGED.

All files under nnn user indices were purged via the reformat option. This can occur only with a system origin job.

LEVEL-1 INDEX BLOCKS LINKED.

If the validation file can be reformatted to eliminate block linkage, searches will be faster for user numbers residing in linked blocks and nonexistent user numbers which would have resided in linked blocks.

CREATING usernum.

User number usernum is being created.

UPDATING usernum.

User number usernum is being updated.

The following informative messages are self-explanatory.

CREATION COMPLETE.

UPDATE COMPLETE.

REFORMAT COMPLETE.

CONVERSION TO SOURCE COMPLETE.

INQUIRY COMPLETE.

1.1.8 ISF CONTROL STATEMENT

The ISF control statement is used to initialize the fast-attach system files VALIDUs, PROFILa, RSXDid, and RSXVid (where id is the machine id). A fast-attach file is a special direct access file that is initialized with the E parameter on the ISF control statement rather than attached with the ATTACH control statement. It is released with the R parameter in the ISF control statement rather than with the RELEASE control statement.

Because the fast-attach files are system files, the ISF control statement must appear in a system origin job. Processing of the statement causes a search of the special permanent file catalog (UI=377777) for files with the predefined names listed above. They are defined in a table internal to ISF. Format of the ISF control statement is

$$\text{ISF,} \quad \left\{ \begin{array}{l} \text{E=filename} \\ \text{R=filename} \end{array} \right\} \quad \text{,FM=family.}$$

where the parameters have the following significance.

E=filename	Initializes the system file identified by filename. If E=0 or no filename is specified (neither E nor R appear), then all files defined in the ISF table are initialized. Refer to the matrix on Table IV-1-1.
R=filename	Releases the currently active system file specified by filename from fast-attach status. If R=0, all of the files in the ISF table for the specified family that are currently active are released.

NOTE

Since initialization and release are mutually exclusive, E and R cannot appear on the same control statement. Also, the ISF control statement has no effect if the device containing the file has the initialize bit set.

FM=family	Designates a specific family of devices. This is necessary only when a family other than the default is desired. Such a case would be the initialization of a new VALIDUs file for an alternate system.

ISF can be entered as a command from the console. The format is:

$$\text{X.ISF,} \quad \left\{ \begin{array}{l} \text{R=filename} \\ \text{E=filename} \end{array} \right\} \quad \text{,FM=family.}$$

The fast-attach file mechanism provides a method to be used by special system jobs for files which are to be retained as permanent files but have a high enough access rate to make permanent file ATTACHs very time-consuming. When a permanent file is activated as a fast-attach file, an FNT entry is made which retains the basic data normally kept in the catalog entry and system sector of the file (interlocks and filename). This dispenses with the catalog search and system sector read normally necessary to attach a permanent file.

If the file is a shared (global) fast-attach file for a multimainframe network, additional information is also maintained in the FAT table on the link device. The type of file determines whether it is entered as global fast-attach or as local fast-attach. This criterion is kept internal to ISF. Basically, VALIDUs and PROFILa are entered as global fast-attach files if they reside on a shared device. The resource files are always entered as local fast-attach. A limit of 77_8 exists on the number of files that can be entered as global fast-attach in a multimainframe environment.

If the ISF control statement is entered with any parameters [for example, X.ISF(R=RESEXid] , the procedure file SYSJOB is not submitted nor is SYSPROC called. They are submitted/ called only with the entry of X.ISF. alone.

The matrix in Table IV-1-3 shows the action of initialize (E) and release (R) relative to the individual fast-attach files.

TABLE IV-1-3. INITIALIZE AND RELEASE OF FAST-ATTACH FILES

Name of Fast-Attach System File	Option	
	Initialize (E)	Release (R)
VALIDUs PROFILa	Make global fast attach.	Return from fast attach to normal direct access.
RSXDid† RSXVid†	Make local fast attach. If file does not exist in catalog (UI=377777), ISF will create the file and attach as a fast-attach file.	Return from fast attach to normal direct access. If RSXDid is specified, RSXVid will also be returned automatically.

One use of the R option is to release fast-attach files activated on a device that is to be initialized. Device initialization is not initiated so long as any direct access files are active on the device (an activated fast-attach file is treated the same as an active direct access file). Until these files are released, MSI will reply to an attempted device initialization with the error message:

ACTIVE FILES ON DEVICE

† The resource files are generated and maintained uniquely for each machine id in a multi-mainframe or single mainframe system by appending the machine id to the file name (for example, RSXVid becomes RSXVAB on the machine with an id of AB).

1.2 USER ACCOUNTING

The special system file PROFILa† contains the information required to control a user's accounting and access to the system. This access is defined not only by charge numbers and project numbers, but also by time in, time out, SRUs, accumulated and connect time. In addition, all exercise of this access by individual users is written by the system to the accounting dayfile, thereby affording the customer a time-log as a basis for his account billing.

Programs that access PROFILa must run as special system jobs.

PROFILa affords three levels of job accounting.

Charge Number	This is the primary division of the customer's job structure. It is a 1- to 10-character billing identifier. Charge numbers can only be entered onto PROFILa by the analyst in a system origin job.
Project Number	This is an optional second level division of the charge number. It is a 1- to 20-character identifier of a particular company project. The project number can be followed by time-access parameters to this project. Project numbers and their associated parameters can be entered and changed by users who have been declared master users in the creation of the current PROFILa file.
User Number	The third level is a 1- to 7-character identifier of the individual user who is allowed access to a designated company project. This is the same user number that VALIDUs† furnishes to verify system access (section 1.1). User numbers are entered and deleted by the master user. A user can be validated for more than one project in the same charge category or in different ones.

Although charge, project, and user numbers can be entered by the analyst at the console, practical dictates of an industrial situation usually require the analyst to create only a shell of the PROFILa file; that is, he enters just the charge numbers and the associated master users. This relatively constant information is furnished by the customer. Following this, the master users will update the PROFILa file with the projects and users that are under their direct cognizance. Then, if a user's validation for system access includes bit 7 of the access word being clear (AW=CCNR, section 1.1.3), this user must enter valid charge and project numbers.

This two-stage structuring of a PROFILa file is illustrated in Figure IV-1-5.

---

† In this manual, the user validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

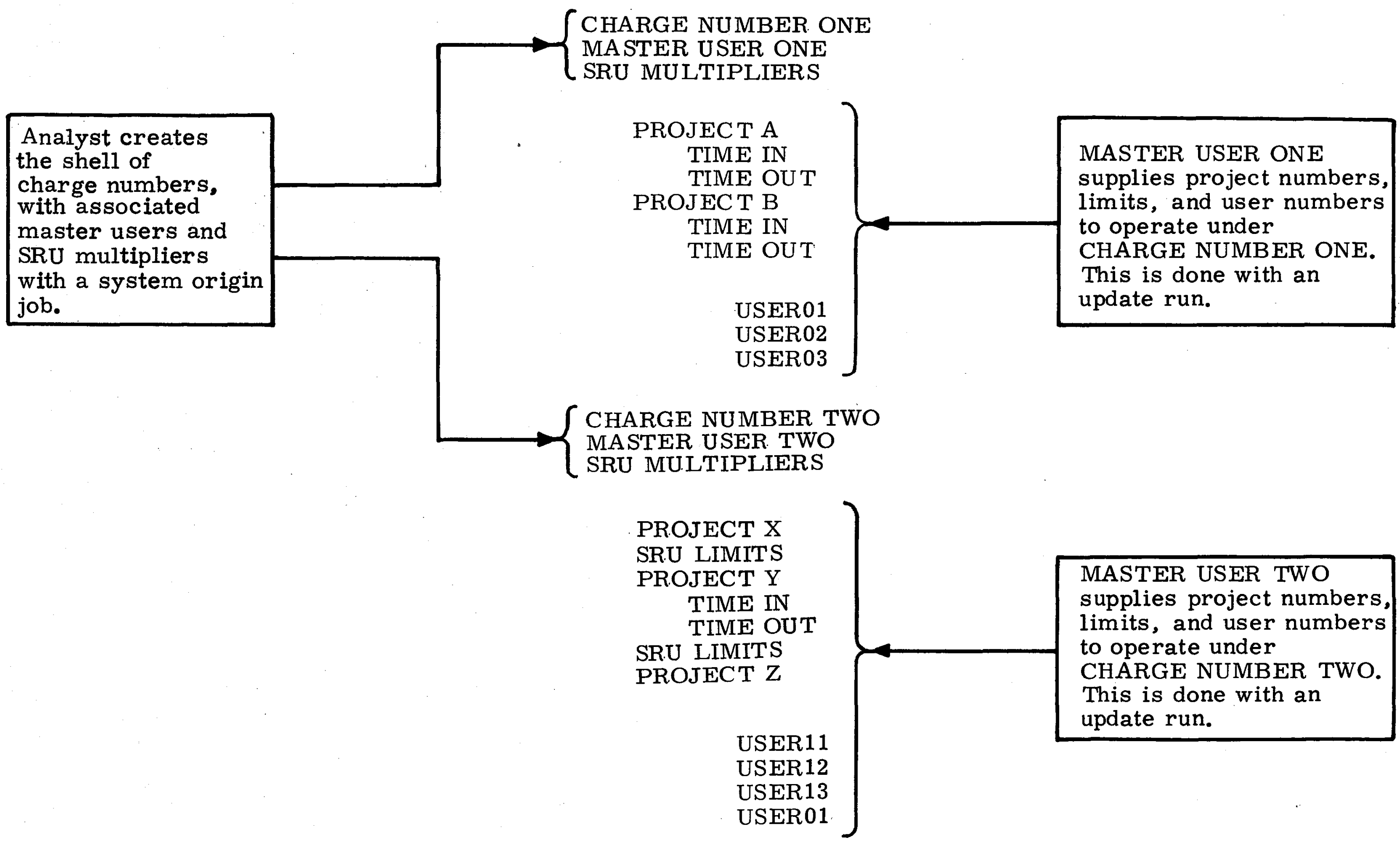


Figure IV-1-5. Representative Structure of a PROFILa File

1.2.1 PROFILa PROJECT PROFILE MANAGER

The program PROFILE uses the special system file supervisor (SFS) to create and manage the project profile file, PROFILa .Creation of PROFILa can only be done from the console via a system origin job (section 1.2.3). Access and modification of an existing PROFILa file can be done from console, card reader (section 1.2.4), or from a time-sharing terminal (section 1.2.5), depending on the option. In all cases, options are called into execution with the PROFILE control statement. The format of the statement is

PROFILE(p_1, p_2,..., p_i,..., p_n)

where the p_i's are the identifications used in defining project profile operations and files.

IDENTIFICATIONS ONLY THE ANALYST CAN USE

Identification	Description
OP=C	C indicates a create option. Input directives are processed so as to create a new PROFILa file. Directives are entered either through the K display or input file.
OP=K	K display option. All other options are cleared and instructions must be entered via the K display. K is the default value for a system origin job. Hence, if no parameters are specified and the call statement is simply PROFILE. the K option is automatically selected.
OP=R	R indicates a restructure run that rebuilds a copy of the current PROFILa file, discarding any deleted entries and reconstructing the directory to reduce file access. The existing PROFILa file is replaced with this restructured file.
OP=S	S indicates a source run that returns the PROFILa file to source format (directive images) and places this source code on the source file (either S=source or SOURCE). This source file is used as the input for a later create or update.

CV	Converts KRONOS 2.1.1 to NOS or NOS to KRONOS 2.1.1. This parameter is meaningful only with OP=C or OP=S options. CV with OP=C Kronos 2.1.1 directives are assumed as source input. Warning messages are generated when obsolete KRONOS input directives (CP or AP) are encountered. The NOS directives M1, M2, M3, M4, AD, SR, and AS may be physically inserted into the KRONOS source input file before the NOS PROFILE create run occurs, if other than the default values are desired, or a later update run may be used to set these new NOS directives. CV with OP=S A source file is generated which is later used for a KRONOS 2.1.1 PROFILE create run. NOS PROFILE suppresses the generation to source of the directives unrecognizable by KRONOS PROFILE (M1, M2, M3, M4, AD, SR, and AS).
S=source	source identifies the file to receive PROFILa source data for the option OP=S. Default is SOURCE.
FN=name	name indicates the family name the user wishes PROFILE to access. To use this option, the user must have system origin privileges.

Identification	Description
OP=L, LO=F	Produces a full listing of the whole PROFILa file; Figure IV-1-6 is an example.
OP=L, LO=C	Produces a listing of charge numbers only for the whole PROFILa file; Figure IV-1-7 is an example.
OP=L, LO=P	Produces a complete charge number and project number listing for the whole PROFILa file; Figure IV-1-8 is an example.
OP=L	Default is LO=F.

IDENTIFICATIONS THE ANALYST AND MASTER USER CAN BOTH USE

Identification	Description
I=Input	Identifies the file that contains input data for a create (OP=C) and an update (OP=U); default is INPUT
L=output	Identifies the file to receive output listings; default is OUTPUT
P=profile	Project profile file; default is PROFILa.
OP=U	Updates the project profile file with directives supplied by the input file
OP=T	Time-sharing update. Processing is the same as OP=U but preliminary instructions are suppressed at the terminal.
OP=I, CN=xxxx.	Charge number inquire. All project numbers valid for charge number CN are written to the output file; Figure IV-1-9 is an example.
OP=I, CN=xxxx, PN=yyyy.	Project number inquire. The control values and all valid user numbers for project number yyyy are written to the output file; Figure IV-1-10 is an example.

```
PROFILa  FULL FILE LIST            OF FULL FILE.                  PAGE 1
         CHARGE NUMBER             MASTER USER           75/02/12.  10.48.43.

    1.     CHGNODD                    MUSERD
             CONTROLS FOR CHARGE NUMBER   CHGNODD

              M1 =         77B          (      1.000      )
              M2 =         77B          (      0.100      )
              M3 =         77B          (      0.003      )
              M4 =         77B          (      0.000      )
              AD =         77B          (      0.000      )

             CONTROLS FOR PROJECT NUMBER PJDD
          CREATE =       75/02/12.                  SR  =          64
          MODDED =       75/02/12.                  AS  =           0
              TI =       00.00.00.                  CT  =          64
              TO =       00.00.00.                  AT  =           0

    2.     CHGNUMA                    MUSERA
             CONTROLS FOR CHARGE NUMBER   CHGNUMA

              M1 =          6B          (      0.500      )
              M2 =         10B          (      0.062      )
              M3 =          5B          (      0.005      )
              M4 =         15B          (      0.013      )
              AD =         26B          (     22.000      )

             CONTROLS FOR PROJECT NUMBER PRJN001A
          CREATE =       75/02/12.                  SR  =         160
          MODDED =       75/02/12.                  AS  =          80
              TI =       11.00.00                   CT  =         210
              TO =       17.00.00                   AT  =          56

             USER NUMBERS VALID TO USE PRJN001A
          UN0101A      UN0201A     UN0301A    UN0401A


    3.     CHGNUMB                    MUSERC
             CONTROLS FOR CHARGE NUMBER   CHGNUMB

              M1 =         21B          (      0.700      )
              M2 =         30B          (      0.087      )
              M3 =         37B          (      0.031      )
              M4 =         74B          (      0.060      )
              AD =         12B          (     10.000      )

             CONTROLS FOR PROJECT NUMBER PRJN001B
          CREATE =       75/02/12.                  SR  =         250
          MODDED =       75/02/12.                  AS  =          20
              TI =       13.00.00.                  CT  =         180
              TO =       21.00.00                   AT  =          72

             USER NUMBERS VALID TO USE PRJNOO1B
          UN0101B     UN0201B
```

Figure IV-1-6. Full File List (OP=L, LO=F)

```
PROFILa CHARGE NUMBER LIST OF FULL FILE.                  PAGE     1
        CHARGE NUMBER       MASTER USER         75/10/02.  00.27.11.

   1.   CHARGNULLY          NULL
   2.   CHARGNULL2
   3.   CHARG0001           PROFILE
   4.   CHARG0002           PROFILE
   5.   CHARG0003           PROFILE
```

Figure IV-1-7. Full File Charge Number List (OP=L, LO=C)

```
PROFILa PROJECT NUMBER LIST OF FULL FILE.                 PAGE     1
        CHARGE NUMBER       MASTER USER         75/10/02.  00.27.06.

   1.   CHARGNULL1          NULL
   2.   CHARGNULL2
   3.   CHARG0001           PROFILE

            VALIDATED PROJECT NUMBERS ARE -
        PN10                                PN11
        PN12                                PROJECTNUM0001

   4.   CHARG0002           PROFILE

            VALIDATED PROJECT NUMBERS ARE -
        PN20                                PN21
        PN22                                PROJECTNUM0002

   5.   CHARG0003           PROFILE

            VALIDATED PROFECT NUMBERS ARE -
        PROJECTNUM0003
```

Figure IV-1-8. Full File Project Number List (OP=L, LO=P)

```
VALIDATED PROJECT NUMBERS ARE -
     PROJ01                                      PROJ02
     PROJ03                                      PROJ04
     PROJ05                                      PROJ06
     PROJ07
```

Figure IV-1-9. Charge Number Only List (OP=I, CN=xxxx)

```
   CONTROLS FOR PROJECT NUMBER PROJ02
CREATE  = 75/04/11.          CP =          200
MODDED  = 75/04/12.          AP =           10
    TI  = 08.00.00.          CT =          100
    TO  = 12.00.00           AT =            5

   USER NUMBERS VALID TO USE PROJ02
          USERABC     USER123
```

Figure IV-1-10. Project Number List (OP=I, CN=xxxx, PN=PROJ02)

Identification	Description
OP=L, LO=FM	Full list of everything accessible on the PROFILa file by the master user (A sample listing is given in Figure IV-1-11.)
OP=L, LO=CM	Charge number list of all charge numbers accessible on the PROFILa file by the master user (A sample listing is given in Figure IV-1-12.)
OP=L, LO=PM	Project number list of all project numbers accessible on the PROFILa file by the master user (A sample listing is given in Figure IV-1-13.)

1.2.2 CHARGE CONTROL STATEMENT

The system routine CHARGE provides validation of a user's charge and project number for access to defined segments of his resources within the system. A call to CHARGE will be required for either a master user or a project-level user if bit 7 (AW=CCNR) in his access word is not set (section 1.1.1). Operation of the CHARGE routine is flowcharted in Figure IV-1-14.

If validation fails, the job is aborted and an appropriate error message is issued to the dayfile (section 1.2.6). If the user is at a terminal, this message is returned to the terminal.

If validation is successful, the following three events occur.

1. Accounting information is written to the accounting dayfile (refer to section 1.5.4 for message formats).
2. The accounting parameters associated with the user's charge-project number are inserted into the accounting formula (refer to section 1.4.1). They are used in calculating the billing unit until the end of job/session or until another charge-project number is entered.
3. The SRU accumulator is set to zero. The CP, MS, MT, and PF accumulators are not altered in any way. If the minimum charge installation option has been selected and if the accumulated SRUs are less than the minimum charge amount, the minimum charge value is entered into the account dayfile (section 1.4 describes the parameters and the minimum charge installation option).

The following is the ordered list of the account dayfile and user's dayfile messages issued whenever a new charge number is entered.

```
yy.mm.dd.  hh.mm.ss.jobnameo.  UDCI, xxxxxx.xxx KCHS.  (Telex users only)
yy.mm.dd.  hh.mm.ss.jobnameo.  UDCI, xxxxxx.xxx KCHS.  (Telex users only)
yy.mm.dd.  hh.mm.ss.jobnameo.  UDND, xxxxx.xxx UNTS.
yy.mm.dd.  hh.mm.ss.jobnameo.  UDCP, xxxxx.xxx SECS.
yy.mm.dd.  hh.mm.ss.jobnameo.  UDMS, xxxxx.xxx KUNS.
yy.mm.dd.  hh.mm.ss.jobnameo.  UDMT, xxxxx.xxx KUNS.
yy.mm.dd.  hh.mm.ss.jobnameo.  UDPF, xxxxx.xxx KUNS.
yy.mm.dd.  hh.mm.ss.jobnameo.  ACSR, xxxxxx.xxx UNTS.
yy.mm.dd.  hh.mm.ss.jobnameo.  ACCN, charge,project.
```

Section 1.5 describes in detail the account dayfile messages.

```
PROFILa FULL FILE LIST        OF MASTER USER          PROFILE
        CHARGE NUMBER                                  75/10/02.  00.35.38.

     1.  CHARG0001

           CONTROLS FOR PROJECT NUMBER PN10
          CREATE  =  74/10/02.          SR   =          64
          MODDED  =  74/10/02.          AS   =           0
              TI  =  00.00.00           CT   =          64
              TO  =  00.00.00           AT   =           0

           CONTROLS FOR PROJECT NUMBER PN11
          CREATE  =  74/10/02.          SR   =          64
          MODDED  =  74/10/02.          AS   =           0
              TI  =  00.00.00.          CT   =          64
              TO  =  00.00.00.          AT   =           0

           CONTROLS FOR PROJECT NUMBER PN12
          CREATE  =  74/10/02.          SR   =          64
          MODDED  =  74/10/02.          AS   =           0
              TI  =  17.00.00.          CT   =          64
              TO  =  24.00.00.          AT   =           0

             USER NUMBERS VALID TO USE PN12
         USER1       USER2

           CONTROLS FOR PROJECT NUMBER PROJECTNUM0001
          CREATE  =  74/10/02.          SR   =          64
          MODDED  =  74/10/02.          AS   =        2048
              TI  =  08.00.00.          CT   =         200
              TO  =  17.00.00.          AT   =       20000

             USER NUMBERS VALID TO USE PROJECTNUM0001
         ABCUSER     USERA        1234567


     2.  CHARG0002

           CONTROLS FOR PROJECT NUMBER PN20
          CREATE  =  74/10/02.          SR   =          64
          MODDED  =  74/10/02.          AS   =           0
              TI  =  00.00.00.          CT   =          64
              TO  =  00.00.00.          AT   =           0

           CONTROLS FOR PROJECT NUMBER PN21
          CREATE  =  74/10/02.          SR   =          64
          MODDED  =  74/10/02.          AS   =           0
              TI  =  06.40.00           CT   =          64
              TO  =  07.04.00.          AT   =           0
```

Figure IV-1-11. Master User Full File List (OP=L, LO=FM)

```
/PROFILE, OP=L, LO=CM

    PROFILa CHARGE NUMBER LIST OF MASTER USER        PROFILE

           CHARGE NUMBER                              75/10/02.  00.33.38.

        1.  CHARG0001
        2.  CHARG0002
        3.  CHARG0003
 LIST COMPLETE.
/
```

Figure IV-1-12. Master User Charge Number List (OP=L, LO=CM)

```
PROFILa PROJECT NUMBER LIST OF MASTER USER        PROFILE

       CHARGE NUMBER                              75/10/02.  00.34.19.

     1.   CHARG0001

               VALIDATED PROJECT NUMBERS ARE -
          PN10                                   PN11
          PN12                                   PROJECTNUM0001

     2.   CHARG0002

               VALIDATED PROJECT NUMBERS ARE -
          PN20                                   PN21
          PN22                                   PROJECTNUM0002

     3.   CHARG0003

               VALIDATED PROJECT NUMBERS ARE -
          PROJECTNUM0003

 LIST COMPLETE.
/
```

Figure IV-1-13. Master User Project Number List (OP=L, LO=PM)

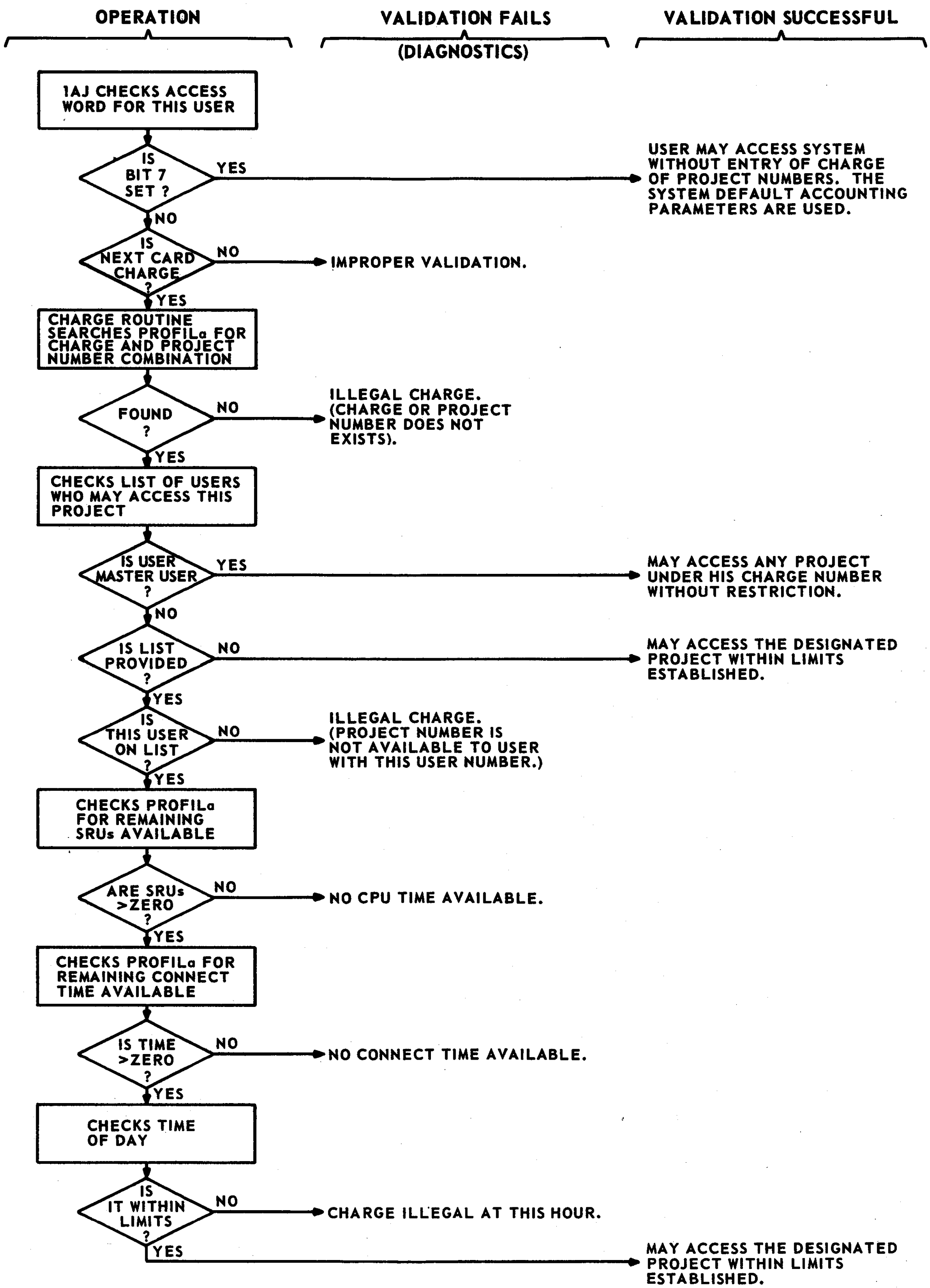


Figure IV-1 14. CHARGE Routine

The CHARGE routine is called from a terminal as either a response to the terminal login request

RECOVER/CHARGE:

or after login by typing in the CHARGE control statement.

The CHARGE routine is called from a batch job by the appearance of the terminal control statement after a user statement.

The format of the CHARGE control statement is:

CHARGE(chargenmbr, projectnumber)

chargenmbr	Charge number (1 to 10 alphanumeric characters)
projectnumber	Project number (1 to 20 alphanumeric characters)

BATCH USAGE OF CHARGE

When a batch job is submitted, the second statement is the user statement which gives the user number used in validating a user for system access. Associated with this number is the access word which determines whether or not this user must supply charge and project numbers. If validation control indicates that these are required, the statement following the user statement must be a CHARGE statement with acceptable charge and project numbers. If additional projects and/or charges are referenced in the job, these references must be accomplished by use of another CHARGE statement. If additional user statements (new users) appear in the input, each one that identifies a user requiring job profile validation must be followed by a CHARGE statement with the requisite charge and project numbers.

CHARGE APPLICATION AT A TERMINAL

When any user logs in at a terminal, the access word associated with his user number is checked to see if he is required to enter a charge number and a project number. If not, the teletype will print out at login:

```
TERMINAL      nn
RECOVER/SYSTEM:
```

The user then types in the subsystem he wishes to use and proceeds with access to resources as defined by his user number.

If charge and project are required, the terminal will print out at login:

```
TERMINAL       nn
RECOVER/CHARGE:
```

This user will not be allowed to enter any commands (except CHARGE, LOGIN, HELLO, RECOVER, GOODBYE, and BYE) until he enters valid charge and project numbers. He will then be allowed access to that particular project under that particular charge under whatever restrictions are currently in force for that project.

If later, while still logged-in, this user wishes to access another project (under this charge or another) for which he is validated, he enters the CHARGE control statement with the appropriate charge and project numbers.

DIAGNOSTICS FOR CHARGE (ERROR MESSAGES)

ILLEGAL CHARGE.

1. Charge or project number does not exist.
2. Project number is not available to a user with this user number.

DATA BASE ERROR.

Error in accounting file structure detected. Contact an onsite analyst immediately.

CHARGE ABORTED.

External job abort.

ILLEGAL CONTROL CARD.

No terminator was found on control statement.

CHARGE FILE BUSY.

File is not available for charge purposes. Wait and try again.

CUMULATIVE SRU LIMIT EXCEEDED.

Accumulated SRUs have exceeded the maximum allowed. †

CONNECT TIME LIMIT EXCEEDED.

Terminal connect time allowed under this project number has been expended. †

† The system does not currently provide a mechanism to update these accumulators in PROFILa.

CHARGE ILLEGAL AT THIS HOUR.

This project number cannot be used at this time of day.

1.2.3 PROFILE CONSOLE INPUT

A **PROFILa** file can only be created by calling PROFILE from the console (system origin job). Likewise, an existing **PROFILa** file can be restructured, returned to source, or read to an output file as a full-file listing only from the console. The remaining operations (update, inquire, and master user listings) can be executed from console, batch (section 1.2.4), or a terminal (section 1.2.5).

A new **PROFILa** file can be created via the console by means of the DIS display. A pre-established input file of control values is called and the PROFILE command with OP=C is entered. The following example, given an input file (INPUT) with control values for structuring a new **PROFILa** file, is a create run from the console.

X.DIS.

SUI(377777)

GET(INPUT)

PROFILE(OP=C)

The K display can be used for both a create and an update. With the K option, directives are entered directly via the console instead of from an input file. If directives are entered for an existing charge and project number, the control values are changed according to the directive; if the charge number and project number are not found, they are considered new ones and are added to the PROFILa file **if it exists or are used to create a new one if it does not exist.**

The K display is called with:

X.PROFILE.

The B display will indicate the control point to which PROFILE is located. A flashing message on that line will be:

REQUEST K DISPLAY

The analyst types in

K,n.

where n is the control point number for PROFILE. This brings the K display for **PROFILa** to the left screen (Figure IV-1-15). The analyst is now ready to create or update. The following example illustrates an update input stream to follow the above.

```
            PROFILa
CREATED   75/02/20.        LAST MOD    75/04/14.

ALL INPUT DIRECTIVES ARE THE SAME AS THE
REGULAR UPDATE DIRECTIVES.  THE FOLLOWING
COMMANDS ARE ALSO PROVIDED.
/CN - BEGINS ACTION ON CHARGE NUMBER *CN*
END -   TERMINATES INPUT OF DIRECTIVES FOR THE
        SELECTED CHARGE NUMBER, AND UPDATES
        PROFILa IF SO DIRECTED.
DROP-   TERMINATES INPUT OF DIRECTIVES FOR THE
        SELECTED CHARGE NUMBER, AND PREVENTS
        ANY UPDATE OF INFORMATION FOR THE
        CURRENT PROJECT NUMBER.
STOP-   END OF PROCESSING.


            CHARGE NUMBER    CHARG

            PROJECT NUMBER   PROJ

            MASTER USER
                CREATED         75/04/14.
                LAST MOD        75/04/14.

OPT     VALUE       DESCRIPTION
 TI =     00.00.00.   TIME IN  - 4 CHARS NUMERIC
 TO =     00.00.00.   TIME OUT - 4 CHARS NUMERIC
 SR =            64 MAXIMUM ACCUMULATED SRU
 AS =             0 Accumulated SRU
 CT =            64 CONNECT TIME
 AT =             0 ACCUMULATED CONNECT TIME
 M1 =   77B     (  1.000    )      SRU MULTIPLIER
 M2 =   77B     (  0.100    )      SRU MULTIPLIER
 M3 =   77B     (  0.003    )      SRU MULTIPLIER
 M4 =   77B     (  0.000    )      SRU MULTIPLIER
 AD =   77B     (  0.000    )      SRU CONSTANT
```

Figure IV-1-15. PROFILA K Display (Left Screen Only)

```
K./CHARJNUM1          (The / is used when PROFILE is updating a
                      particular charge number; otherwise, it is
K.MU=MUSE1            not used.)
K.END
K./CHARJNUM2
K.MU=MUSE2
K.END
K.STOP
```

INPUT DIRECTIVES

The input stream for a PROFILE create or update is divided into charge numbers and associated charge number entries. All directives relative to a charge number must appear within the range of that charge number, that is, until another charge number appears or the end of the input stream is encountered. A charge number can appear only once in an input stream on a create run.

The directives within a charge number entry are further divided into master user and SRU multipliers, project number, and associated project number entries. The project number entry contains the data identifiers that establish the control values for this project and the list of user numbers that may access this project. The occurrence of duplicate project numbers under the same charge number entry in the input stream is not allowed on a create run.

Figure IV-1-16 diagrams a typical input stream.

The statement for a charge number must begin with a / in column 1. The one-to ten-characters following the / are the charge number name. This name is terminated with a separator. Separators consist of all special characters (except /), end-of-line, and end-of-card. Directives may follow the separator.

All data within a charge number entry is free format to column 72. Directives cannot be split between cards or lines. Blanks are ignored. To allow sequencing and/or identification of input directives, all data past column 72 is ignored.

The format of a data identifier is

ident = data

where ident is the two-character designation of the limiting parameter and data is the value applied to the project under which this identifier appears. The following are the available identifiers for PROFILE input.

Identifier	Description
MU	Master user number which has the ability to update, inquire, and make listings for the projects entered under the same charge number as this master user This master user number must be specified on the USER statement for batch input (section 1.2.4) or when logging in (for time-sharing, section 1.2.5) in order for the master user to exercise the project-oriented privileges specified.

Charge Number - - - - - - - - - - /CHNUMBER01

Charge Number Entry

Master User - - - MU=USEAAA

Project Number -- PN=PROJECTNUMBER001

Project Number Entry

TI=1200

TO=1700

SR=200

UN=USERA11

UN=USERA12

Project Number -- PN=PROJECTNUMBER002

Project Number Entry

TI=0800

TO=1200

UN=USERA21

UN=USERA22

UN=USERA23

Charge Number - - - - - - - - - - /CHNUMBER02

Charge Number Entry

Master User - - - MU=USERBBB

Project Number -- PN=PROJECTNUMBER011

Project Number Entry

TI=0000

TO=0800

SR=500

UN=USERB11

UN=USERB12

Figure 1-16. Typical Input Stream for Use with PROFILE

Identifier	Description
	The master user number is 1 to 7 alphanumeric characters. There can be only one master user per charge number.
M1	Index for the SRU multiplier to weight calculated system resources used against those not directly measurable for this charge number. This may be 1 or 2 numeric digits. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. The maximum and default value of 77B gives system default. M1 can only be set from a system origin job. (Refer to section 1.4.5 for more information on the usage of this parameter.)
M2	Index for the SRU multiplier to weight input/output usage for this charge number. This may be 1 or 2 numeric digits. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. The maximum and default value of 77B gives system default. M2 can only be set from a system origin job. (Refer to section 1.4.5 for more information on the usage of this parameter.)
M3	Index for the SRU multiplier to weight central memory field length usage for this charge number. This may be 1 or 2 numeric digits. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. The maximum and default value of 77B gives system default. M3 can only be set from a system origin job. (Refer to section 1.4.5 for more information on the usage of this parameter.)
M4	Index for the SRU multiplier to weight extended core field length usage for this charge number. This may be 1 or 2 numeric digits. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. The maximum and default value of 77B gives system default. M4 can only be set from a system origin job. (Refer to section 1.4.5 for more information on the usage of this parameter.)
AD	Index for the SRU constant used to charge for resources not directly measurable by the system for this charge number. This may be 1 or 2 numeric digits. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. The maximum and default value of 77B gives system default. AD can only be set from a system origin job. (Refer to section 1.4.5 for more information on the usage of this parameter.)

Identifier	Description
PN	Project number. This is a 1-to 20-character alphanumeric designation of a particular customer activity. Every project number must be associated with a charge number.
UN	User number. A 1- to 7-alphanumeric identification of the individual or individuals who will have access to the project or projects under which this user number is entered. It is also the identification used by VALIDUs to establish system access (section 1.1.1). Under NOS, all files are cataloged by user number (section 2.1). If no user numbers are specified for a project, then all user numbers are allowed to use it.
TI	Time of day before which the validated user cannot use this project number. This is expressed in four-digit military time notation. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. Maximum value is 2400D.
TO	Time of day after which the validated user cannot use this project number. This is expressed in four-digit military time notation. A radix may follow to indicate decimal (D) or octal (B). If the radix is omitted, decimal is assumed. Maximum value is 2400D.
CT †	Total amount of accumulated time in seconds that users are allowed to be connected to this project. Time may be 1- to 10-numeric digits. A radix may be added. Default value is 100B.
AT †	Connect time in seconds that this project number has accumulated. Time may be 1- to 10-numeric digits. A radix may be added. If this value is entered on a create or update run, it will establish an arbitrary origin from which further counting will start. Appearing on a display, it indicates the actual time recorded.

† These fields are not currently checked or updated by the system.

Identifier	Description
SR †	Total amount of system resource usage this project is allowed to accumulate. This may be 1 to 10 digits. If the radix is omitted, decimal is assumed. The default value is 100B.
AS †	System resource usage this project number has accumulated. This may be 1 to 10 digits. If the radix is omitted, decimal is assumed.
DC	Delete charge number. This directive may appear only on an update run from the console (system origin job). It removes the specified charge number from the current PROFILa file. This automatically removes all projects and users under this charge. It does not affect a user's access under another charge number.
DP	Delete project number. This directive can appear only in an update run. It removes the designated project from the charge number entry under which it was entered.
DU	Delete user number. This directive can appear only on an update run. It removes the designated user number from the specific project under which it was entered. It does not affect any access it may have under other projects.

Default on all directives is zero unless specified otherwise.

1.2.4 PROFILE BATCH INPUT

A master user can initiate an update, inquire, or listing of the projects under his charge number via card reader input. The following is an example of a batch update in which master user MASTR1 adds a time-in (TI) and a time-out (TO) to one of his projects (PROJ2).

```
JOBUPDA.
USER(MASTR1)
PROFILE(OP=U)
7/8/9
```

† These fields are not currently checked or updated by the system.

/CHARJNUM1

PN=PROJ2

TI=1400

TO=1800

6/7/8/9

1.2.5 PROFILE EXECUTION FROM A TERMINAL

A master user can initiate an update, inquire, or listing of the projects under his charge number from a time-sharing terminal. To do this he must have his master user status validated at login. After this validation is affirmed, the master user must enter the BATCH subsystem in order to use the PROFILE directive.

The following is an example of a login by master user MASTR1. †

```
  74/12/25
CDC MULTI-MODE OPERATING SYSTEM              NOS 1.
USER NUMBER: MASTR1    CR
PASSWORD
[redacted]   CR
TERMINAL:          32, TTY
```

At this point, the system will either respond with

```
RECOVER/CHARGE:
```

or with

```
RECOVER/SYSTEM:
```

For further illustration, assume this master user wants to add limiting identifiers to project PROJ2 which is one of the projects under CHARJ1. If the system had responded with the first reply, the sequence that follows is:

```
RECOVER/CHARGE: CHARGE, CHARJ1, PROJ2
READY.
BATCH  CR
$RFL, 20000
/
```

If CHARGE was not required, the sequence is:

```
RECOVER/SYSTEM: BATCH  CR
READY.
/CHARGE(CHARJ1, PROJ2)  CR           [only if needed]
/
```

The master user is now ready to enter the identifiers under an update.

---

† In the representation of terminal printout, user responses are underscored. Also, each carriage return CR is by the user.

UPDATE FROM A TERMINAL

The time-sharing master user can initiate an update by issuing the command PROFILE(OP=U). Once initiated, the following block of information is output.

```
FOLLOWING ARE THE VALID INPUT DIRECTIVES
FOR UPDATE -
   PN = PROJECT NUMBER
   UN = USER NUMBER
   TI = TIME IN
   TO = TIME OUT
   SR = SRU MAXIMUM USAGE
   AS = ACCUMULATED SRU USAGE
   CT = MAXIMUM CONNECT TIME
   AT = ACCUMULATED CONNECT TIME
   DC = CHARGE NUMBER TO DELETE
   DP = PROJECT NUMBER TO DELETE
   DU = USER NUMBER TO DELETE
A BLANK INPUT LINE TERMINATES THE
UPDATING OF THE GIVEN CHARGE NUMBER.
```

This informative printout can be suppressed by using the control statement option OP=T instead of OP=U. In all other regards, the operation of T is identical to U.

If charge and project numbers are required, the example would run as follows:

```
ENTER CHARGE NUMBER
? CHARJ1  CR
ENTER DIRECTIVES
? PN=PROJ2  CR
? TI=0800  CR
? TO=1200  CR
?  CR
ENTER CHARGE NUMBER
?  CR
PROFILa UPDATED.
/
```

INQUIRE FROM A TERMINAL

The master user can request information on his current charge number and its project numbers by means of an inquire from a terminal. To do this, he enters the BATCH subsystem and issues the command

PROFILE(OP=I, CN=charge, PN=project)

where charge is his charge number and project is the particular project for which he wants information.

If the charge number has not been supplied, PROFILE outputs

ENTER CHARGE NUMBER

when it processes the command. The user must type in the appropriate charge number. If a null line is entered, it is treated as an end-of-file (end processing).

After output of the desired information or if a charge number has not been supplied in the PROFILE control statement, PROFILE outputs

ENTER CHARGE NUMBER

to the terminal and waits for the user to enter the desired charge number. If a null line is entered, it is treated as an end-of-file (end processing).

If a project number has not been entered on the control statement or if a charge number has just been supplied in response to ENTER CHARGE NUMBER, PROFILE outputs

ENTER PROJECT NUMBER

to the terminal and waits for the user to enter the desired project number. If a null line is entered, output consists of a list of valid project numbers under this charge number, and PROFILE again responds

ENTER PROJECT NUMBER

If a project number is entered, output consists of a list of the controls for and valid users of this project number. PROFILE again responds

ENTER PROJECT NUMBER

until a null line is entered to indicate end of processing for the current charge number.

LIST FROM A TERMINAL

The master user can request an FM, CM, or PM listing from a terminal. These listings are explained in section 1.2.1. Sample listings are given in Figures IV-1-11, IV-1-12, and IV-1-13.

As with an update and an inquire, the master user must enter the BATCH subsystem to use this option.

1.2.6 DIAGNOSTICS FOR PROFILE

PROFILE SYSTEM ERROR MESSAGES

DATA BASE ERROR.

Erroneous control word on attached PROFILA file

LEVEL-3 DATA BASE ERROR.

Erroneous control word in format of file on data level

PROFILE DAYFILE MESSAGES

ERROR IN PROFILE ARGUMENTS.

Error on control statement

NO INPUT FILE.

No directives present

PROFILE ABORTED.

Error flag set at control point

DIRECTIVE ERRORS.

Examine output file to determine reason for error. Check diagnostics to output file listed in the next section.

ILLEGAL PROFILE INQUIRE.

User not allowed to access control information for charge number supplied

PROFILA CREATED.

Creation run completed

PROFILa UPDATED.

Update run completed

FILE NOT AVAILABLE CURRENTLY.

PROFILa file busy; try again later.

INQUIRY COMPLETE.

Inquire run completed

REFORMAT COMPLETE.

Reformat completed

PROFILA TO SOURCE.

Source run completed

LIST COMPLETE.

List of PROFILa completed

PROFILE DIAGNOSTICS TO OUTPUT FILE

**** ERROR IN CHARGE NUMBER

Format error in charge number; does not have 1- to 10-alphanumeric characters.

Action: Charge number entry is disregarded and PROFILE skips to next charge number entry in the input stream.

**** DUPLICATE CHARGE NUMBERS

This error can occur only on a create run. Two charge entries referring to the same charge number have been found. (Two charge number entries with the same charge number are legal for an update run.)

Action: All charge number entries after the first one with that charge number are disregarded. All other charge number entries that do not have duplicate charge numbers are processed normally.

**** DIRECTIVE ERROR

This error can occur only on a create run. A delete directive (DC, DP, or DU) was encountered on this create run.

Action: The delete directive is ignored while all other directives are processed.

**** DUPLICATE PROJECT NUMBER

This error can occur only on a create run. Two or more project number entries within a charge number entry refer to the same project number. (It is illegal to enter the same project number more than once for a particular charge number during a create run.)

Action: All duplicate project number entries after the first one are disregarded. All other project number entries that do not have duplicate project numbers are applied normally.

**** DUPLICATE USER NUMBER

This error can occur in two possible situations.

1. The same user number (UN) directive appears more than once with a project number.

2. An update attempts to add a user number that already exists under the project number.

Action: The entire project number entry containing the duplicate user number is disregarded.

**** ERROR IN IDENTIFIER

An unrecognized directive identifier is encountered.

Action: Depends on the position of the error within the input stream of directives. There are four possible conditions. The following sample input stream is used to illustrate the four possible actions.

```
/CHARGENUMB - - - - - - - - - - - - - - - - - - - - - -  |
 MU=MASTERU                                              |
 PN=PROJECTNUMBER1 - - - - -                             |
      SR=0777B              }  Project                   |
      UN=USERA              }  Number                    |   Charge
      UN=USERB    - - - - - }  Entry                     |   Number
 PN=PROJECTNUMBER2 - - - - -                             |   Entry
      SR=0377B              }  Project                   |
      UN=USERC              }  Number                    |
      UN=ESERD    - - - - - }  Entry                     |
                 - - - - - - - - - - - - - - - - - - - - |
```

1. If the error occurs within a project number entry, that entire project number entry is disregarded.

2. If the error occurs in one of the directives MU, DC, or DP (but not PN) within the first charge number entry, only the erroneous directive is disregarded.

3. If the error is in the first PN directive, the entire project number entry will eventually be disregarded since the directives immediately following that PN will be treated as directives coming ahead of the first PN directive and will, accordingly, be ignored.

4. If the error occurs in any PN directive after the first one, it is treated as an error within the preceding project number entry. Both the project number entry under the erroneous project number and the project number entry under the preceding project number are disregarded.

**** USER NOT VALID TO UPDATE

The user number of the person attempting the PROFILE update run is not the master user number for that charge number entry. This error can only occur during an update run.

Action: The unacceptable charge number entry is disregarded and PROFILE skips to the next charge number entry in the input stream.

**** MULTIPLE MASTER USER NUMBER

More than one master user (MU) directive is encountered within a charge number entry in the input stream. This error can only occur during a create run.

Action: All master user number directives after the first one are disregarded.

**** ERROR IN NUMERIC DATA

This error occurs when numeric data exceeds the maximum limit for that value or when nonnumeric data is encountered.

Action: The entire project number entry containing the directive with the numeric data error is disregarded.

**** ERROR IN PROJECT NUMBER

The data field of the project number directive (PN) contains a format error (not 1- to 20-alphanumeric characters).

Action: The project number entry associated with the erroneous project number is disregarded.

**** ERROR IN USER NUMBER

The data field of the user number directive (UN) contains a format error (not 1- to 7-alphanumeric characters).

Action: The entire project number entry containing the erroneous UN directive is disregarded.

**** ERROR IN MASTER USER NUMBER

The data field of the master user directive (MU) contains a format error (not 1- to 7-alphanumeric characters or an asterisk).

Action: Entire charge entry is disregarded and PROFILE skips to next charge entry.

**** ILLEGAL UPDATE, DIRECTIVE IGNORED.

The master user does not have the necessary permission to access or change the following input directives.

M1, M2, M3, M4, AD, NU

Action: This input directive is ignored; processing can continue normally.

**** INPUT DIRECTIVE IGNORED.

This warning message is generated on the run (with OP=C, CV specified) when the obsolete KRONOS 2.1 input directive CP (maximum CP time) is encountered.

Action: This input directive is ignored; processing can continue normally.

1.3 USER NUMBERS AND CHARGE NUMBERS

System access (remote/local) privileges are given to a user through the assignment of a user number and a password (optional). Once system access is attained, all billable activity is associated with a charge number. The system provides the central site with the flexibility of equating a charge number to a user number or maintaining them as separate entities. This flexibility is attained through the use of the CHARGE required flag (CCNR bit in the VALIDUs† access word) and the availability of a charge number in the PROFILa† file. The following table indicates the combinations possible and the resultant billing number.

	Charge Number Required	Charge Number Not Required
No PROFILa Entry	This option is not possible since a charge number is required.	The user number is used as the billing number.
PROFILa Entry Available	The resources used during job/session initialization (log-in and user validation) are accumulated and thus billed under the required charge number.	The user number is used as the billing number until such time that the user decides to enter the charge number.

† In this manual, the user validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

1.4 RESOURCE ACCOUNTING

The basic accounting unit for NOS is the system resource unit (SRU). The SRU is a measurement of the resources used by a job or a terminal session. The SRU algorithm combines measurements of the following resources into a single unit.

- Central memory field length
- ECS field length
- CPU time
- Mass storage usage
- Magnetic tape usage
- Permanent file usage

The SRU calculation is dynamic; that is, each time additional amounts of the above resources are utilized by the job or session, the SRU value is updated. The following sections describe the algorithm for calculating SRU values and a detailed description of SRU components.

1.4.1 SRU FORMULA

The following formula is used by the system for SRU computation.

SRU = M1(CP + M2 x IO + M3(CP + IO)CM + M4(CP + IO)EC) + AD

Parameter	Description
CP	Central processor usage expressed in milliunits. The value of this parameter is determined by the following formula. CP = S0 x CP0 + S1 x CP1 CP0 — Time accumulated on CPU 0 in milliseconds CP1 — Time accumulated on CPU 1 in milliseconds S0, S1 — Multipliers used to normalize CP time when the system is running on a dual CPU machine
IO	A measure of the accumulated input/output system activity for a user. This parameter, expressed in milliunits, is defined by the following formula. IO = S2 x MS + S3 x MT + S4 x PF MS — Mass storage activity. The components of this parameter are described in detail in section 1.4.2. MT — Magnetic tape activity. The components of this parameter are described in detail in section 1.4.2. PF — Permanent file activity. The components of this parameter are described in detail in section 1.4.2. S2, S3, S4 — Multipliers used to weight MS, MT, and PF activity against one another

Parameter	Description
CM	Central memory field length expressed in words/1000B
EC	ECS field length expressed in tracks (1 track=2000B words)
M1	Multiplier used to scale the overall SRU value
M2	Multiplier used to weight the I/O activity against CP time, CM field length, and ECS field length usage
M3	Multipliers used to weight CM field length against CP time, I/O activity, and ECS field length usage
M4	Multiplier used to weight ECS field length against CP time, I/O activity, and CM field length usage
AD	Incremental adder which is applied to the SRU value during accounting initialization

The multipliers S0 through S4, as well as the default values for units of MS, MT, and PF, are installation options which do not change during system execution. The multipliers M1 through M4 and the adder AD are also installation options, but they may change once system activity has begun. The default values for M1 through M4 and AD are set during job or session initialization. When a charge number is entered, different values for M1 through M4 and AD may be specified for use in the SRU calculation (refer to section 1.2.3, PROFILE Console Input). These parameters are retained in PROFILa and provide the central site with the flexibility of varying the billing unit for selected users.

1.4.2 SRU PARAMETERS

The common deck COMSSRU contains the definitions for the SRU multipliers and associated parameters. The absolute ranges for these values are also defined. To obtain a listing of COMSSRU, assemble CALLSYS; the default values are shown. To change any default values, modifications must be made in COMSSRU.

The following paragraphs describe the SRU parameters and list the absolute ranges and default values. The COMSSRU name is listed along with the default value for each parameter. When a site sets SRU parameter default values, these values must lie within the absolute range for each parameter.

S0, S1

The values used for S0 and S1 are selected by the system at deadstart time from a list of multipliers defined for each type of CPU detectable by NOS. S0 is the primary multiplier and is used for all single CPU machines. For dual CPU machines, S0 is used for the first CPU (CPU 0) and S1 is used for calculations involving the second CPU (CPU 1). For example, if a site is running a 6700, S0 is assigned the value defined for a 6600 CPU and S1 is assigned the value defined for a 6400 CPU. This allows a site with several systems to use different multipliers for different CPUs while using only one deadstart tape.

It is possible for a site to transform this selection at deadstart by the use of the IPRDECK entry CPM. Use of this entry allows the site to select any multiplier from the list in COMSSRU, which follows, to be used instead of the normally selected value. One advantage of this entry is that an installation may charge differently for the use of a 6200 CPU or a 6400 CPU although the software cannot normally detect the difference.

COMSSRU Name	Default Value	Description
CP62	1.0	6400/CDC CYBER 72 CPU
CP64	1.0	6400/CDC CYBER 73 CPU
CP66	1.0	6600/CDC CYBER 74 CPU
C172	1.0	CDC CYBER 172 CPU
C173	1.0	CDC CYBER 173 CPU
C175	1.0	CDC CYBER 175 CPU

Absolute range: 0.1 to 50.0

S2, S3, S4

These multipliers are used in the calculation of the IO parameter. In addition to providing weighting factors, these multipliers also convert units of resource usage (MS, MT, or PF) to milliunits of IO. For example, if the default value for S2 is used, 300 units of MS usage results in 300 milliunits of IO.

COMSSRU Name	Default Value	Description
S2SR	1.0	MS multiplier (S2)
S3SR	1.0	MT multiplier (S3)
S4SR	1.0	PF multiplier (S4)

Absolute range: 0.1 to 50.0

M1

This multiplier is used as a scaling factor to increase or decrease the overall SRU value. This value may be changed from the system default for each charge number when this charge number is entered (refer to sections 1.2.3 and 1.4.5 for further information).

COMSSRU Name	Default Value	Description
M1SR	1.0	M1 multiplier

Absolute range: 0.1 to 25.5

M2, M3, M4

These multipliers provide weighting of the various terms in the SRU calculation. These values may be changed from the system default values for each charge number when this charge number is entered (refer to sections 1.2.3 and 1.4.5 for further information).

COMSSRU Name	Default Value	Description
M2SR	0.100	M2 multiplier
M3SR	0.003	M3 multiplier
M4SR	0.0	M4 multiplier

Absolute range: 0.001 to 1.023

AD

The value assigned to this parameter is applied to the SRU value during accounting initialization of a job or session. It thus serves as an overhead increment. This value may be changed from the system default for each charge number when this charge number is entered (refer to sections 1.2.3 and 1.4.5 for further information).

COMSSRU Name	Default Value	Description
ADSR	0	Incremental adder (AD)

Absolute range: 1 to 1000

In addition to the parameters which make up the SRU formula, the follwing values are also defined in COMSSRU.

MINIMUM DISPLAY VALUE

This parameter defines the minimum value to be displayed at the end of each time-sharing job step. If the accumulated SRUs are less than this value, they are not displayed.

COMSSRU Name	Default Value	Description
MDSR	0.100	Minimum display value

Absolute range: 0.001 to 1.000

MINIMUM CHARGE VALUE

This parameter defines the minimum SRU value to be applied against a charge number. If accumulated SRUs are less than this value, then a charge equal to this value is applied.

COMSSRU Name	Default Value	Description
MCSR	1.000	Minimum charge value

Absolute range: 0.001 to 10.000

1.4.3 IO INCREMENTS

The IO parameter in the SRU formula is a measure of the accumulated input/output system activity for a user. It accounts for MS activity, MT activity, and PF activity. This parameter is controlled by central site defined increments. These increments are assigned to various functions performed by the system. These functions include data transfer as well as other operations such as file positioning.

This section describes the increments of MS, MT, and PF which make up the measurable portion of the IO parameter. The common deck COMSSRU contains the definitions, default values, and absolute ranges for these increments. To obtain a listing of COMSSRU, assemble CALLSYS. If a site desires to change any of the default increments, modifications must be made in COMSSRU. This section lists the IO increments with COMSSRU names, released default values, and absolute ranges.

MS INCREMENTS

The following increments are for each operation or for each PRU processed.

COMSSRU Name	Default Increment	MS Activity
IMRL	3	READ WITH LIST
IMPO	2	POSITION
IMCO	1	CLOSE/OPEN
IMRS	1	REWIND/SKIPEI
IMLL	1	LIBRARY LOAD

The increments assigned for CIO READ/WRITE operations are defined by the following.

$$\text{number of increments} = (\text{number of PRUs}) \times (2^{\text{IMRW}})$$

COMSSRU Name	Default Value	MS Activity
IMRW	2	CIO READ/WRITE. Using the IMRW released default value of 2, four increments per PRU are charged for CIO READ/WRITE

If the number of PRUs read in one CIO operation falls within the following ranges, an additional charge is made. This charge serves as a penalty for an inefficient I/O transfer.

COMSSRU Name	Additional Charge Increment	Range
IMSA	4	0 - 3 PRUs
IMSB	2	4 - 7 PRUs
IMSC	0	10 - infinite PRUs

Absolute range for MS increments: 0 to 63

The following describes the increments charged for positioning.

COMSSRU Name	Default Value	Description
IMPL	128	Chargable positioning interval for each READ WITH LIST operation. Using the IMPL released default value, one increment is charged for each 128 PRUs positioned from the current position.

MT INCREMENTS

The following increments are charged for each magnetic tape operation.

COMSSRU Name	Default Increment	MT Activity
ITRW	4	READ/WRITE
ITRL	5	READ L TAPE
ITPO	2	POSITION
ITCL	1	OPEN/CLOSE
ITWL	6	WRITE L TAPE

Absolute range for MT increments: 0 to 63

PF INCREMENTS

The following increments are charged for each permanent file operation.

COMSSRU Name	Default Increment	PF Activity
IPSV	1	SAVE
IPRP	1	REPLACE
IPGT	1	GET
IPAP	20	APPEND
IPDF	4	DEFINE
IPAT	4	ATTACH
IPPM	1	PERMIT
IPCG	1	CHANGE
IPPG	1	PURGE
IPCT	1	CATLIST
IPCS	4	CATALOG SEARCH
IPCE	0	CATALOG ENTRY RETURNED
IPVA	1	VALIDUs ACCESS
IPPA	1	PERMIT FILE ACCESS
IPAD	1	ALTERNATE DEVICE ACCESS

The following increment is charged each time a specified number of PRUs are transferred.

COMSSRU Name	Default Increment
IPPR	4

The following specifies the number of PRUs transferred before the IPPR increment is charged.

COMSSRU Name	Default Value	Description
IPPN	10	Using the default value, the IPPR increment is charged each time 10 PRUs are transferred.

Absolute range for PF increments: 0 to 63

1.4.4 EXAMPLE OF SRU CALCULATION

This section illustrates how an actual SRU value is arrived at. The SRU formula given in section 1.4.1 is:

SRU = M1 (CP + M2 x IO + M3 (CP + IO) CM + M4 (CP + IO)EC) + AD

The parameters are described in section 1.4.1. For this example, all default values are assumed, except AD which equals 1.0. Therefore, the following parameters are known.

S0 = 1.0	M1 = 1.0	AD = 1.0
S1 = 1.0	M2 = 0.100	
S2 = 1.0	M3 = 0.003	
S3 = 1.0	M4 = 0.0	
S4 = 1.0		

For the purpose of this example, it is assumed that the job or session accumulated the following amounts of the specified resources.

9.135 milliseconds of CP time on CPU 0 (CP0 = 9.135)
0.0 milliseconds of CP time on CPU 1 (CP1 = 0.0)
28.880 milliunits of mass storage (MS = 28.880)
No magnetic tape or permanent file usage (MT = 0.0, and PF = 0.0)
No ECS usage (EC = 0.0)
10500B CM field length

To solve for the CP parameter, the following formula is used.

CP = S0 x CP0 + S1 x CP1
= 1.0 x 9.135 + 1.0 x 0.0
= 9.135

so

CP = 9135 milliunits

To solve for the IO parameter, the following formula is used.

IO = S2 x MS + S3 MT + S4 x PF
= 1.0 x 28.880 + 1.0 x 0.0 + 1.0 x 0.0
= 28.880

so

IO = 28,880 milliunits

To solve for the CM parameter, the following formula is used.

CM = (CM field length = 777B)/1000B (the 777B is used as a roundup factor)
= (10500B + 777B)/1000B
= 11B
= 9

so

CM = 9 milliunits

To solve for the EC parameter, the following formula is used.

EC = ECS field length in tracks
= 0

so

EC = 0 milliunits

Since the AD parameter is specified in units and the remainder of the SRU equation is specified in milliunits, this part is computed first and converted into units. It can then be added to the AD parameter to obtain the number of SRUs.

M1 (CP + M2 x IO + M3 (CP + IO) CM + M4 (CP + IO) EC)
= 1.0(9135 + 0.100 x 28880 + 0.003(9135 + 28880)9+0)
= 9135 + 28.880 + 1026.405
= 10190.285
= 10190 milliunits
= 10.190 units

so

SRU = 10.190 + AD
= 10.190 + 1.0
= 11.190 units

So, during this job or session, 11.190 SRUs have been accumulated.

1.4.5 MULTIPLIER INDEX VALUES AND ACTUAL MULTIPLIER VALUES

When a site assigns charge numbers, it can specify certain multipliers (M1 through M4) and the adder (AD) in the SRU multiplier formula as other than the system default values. (The system default values are described in section 1.4.3.) In fact, each charge number may be assigned a unique set of multiplier and adder values. This is done using the PROFILE control statement and the M1 through M4 directives (refer to section 1.2.3). An index from 0 to 77B is specified with each directive which is converted to the actual multiplier or adder value. The actual multiplier or adder value must lie within the absolute range defined by the system for that parameter. However, each site can also specify a subrange for each multiplier or adder in which all multipliers or adders must lie. This is done by specifying upper and lower bounds for these parameters. The released values for these upper and lower bounds are contained in COMSSRU. To obtain a listing of COMSSRU, assemble CALLSYS. To change these values, modifications must be made in COMSSRU. The following lists the COMSSRU names and gives the released values.

COMSSRU Name	Released Value	Description
M1SL	0.5	M1 lower bound
M1SU	1.5	M1 upper bound
M2SL	.050	M2 lower bound
M2SU	.150	M2 upper bound
M3SL	.001	M3 lower bound
M3SU	.064	M3 upper bound
M4SL	.001	M4 lower bound
M4SU	.064	M4 upper bound
MASL	1	Adder (AD) lower bound
MASU	64	Adder (AD) upper bound

When a site specifies an index value for M1, M2, M3, M4, or AD under PROFILE, it is converted to the actual multiplier (or adder) value by the following formula.

MI = I(MISU - MISL)/64 + MISL

Parameter	Description
MI	Actual multiplier (or adder) obtained
I	Multiplier (or adder) index value entered with a PROFILE directive
MISU	Multiplier (or adder) upper bound
MISL	Multiplier (or adder) lower bound

The actual multiplier (or adder) value MI, not the index value I, is displayed on the PROFILa K display (refer to Figure IV-1-15).

The following example illustrates a conversion of an index value to an actual multiplier.

Example 1:

Assume that the released upper and lower bounds for M2 are used. That is:

M2SL = .050
M2SU = .150

The site specifies a PROFILE directive

M2 = 32

for a particular charge number. This implies that the index value I in the formula

MI = I(MISU - MISL)/64 + MISL

is equal to 32.

Upon substitution:

M2 = 32(.150 - .050) 64 + .050
= 3.200/64 + .050
= .050 + .050
= .10

Thus, the actual M2 multiplier used for this charge number is .10.

Two exceptions are:

1. If the index value 0 is entered in the PROFILE directive, an actual multiplier value of 0 is assigned.

2. If the upper index value 77B is entered or if no index value is entered in the PROFILE directive, the system default multiplier value is assigned.

After a site has chosen upper and lower bounds for its multipliers (and adder), it may desire to assign different actual multiplier values to certain charge numbers. In order to choose the proper index value to be specified on the PROFILE directives, the following formula is used. (This is merely the previous formula solved for I.)

I = 64(MI - MISL)/(MISU - MISL)

I, MI, MISL, and MISU are the same as those defined previously.

Example 2:

Assume that the released upper and lower bounds for M2 are used. That is:

M2SL = .050
M2SU = .150

The site wishes to assign an actual M2 multiplier value of .10 to a particular charge number. To determine the appropriate index value for the PROFILE directive, the following formula is used.

I = 64(MI - MISL)/(MISU - MISL)

MI = .10 in this case.

Upon substitution:

I = 64(.100 - .050)/(.150 - .050)
= 64(.050)/(.100)
= 64 x .5
= 32

Thus, the PROFILE directive M2 = 32 should be entered for the particular charge number to specify an actual M2 multiplier value of .10.

1.5 ACCOUNT DAYFILE MESSAGES

The purpose of the account dayfile is to provide a history of system usage over the life of the account dayfile. This history is used for the following two purposes.

1. It provides the information necessary to properly bill the users of the system.

2. It provides the necessary information to analyze the use of the system or any part of it by the installation. For example, the installation may want to determine the amount of magnetic tape usage.

Therefore, a standardized message format is provided to ease in the account dayfile analysis. The following is the general format of the account dayfile message. All account dayfile messages have this general format.

hh.mm.ss.jobnameo. geac, additional information

Message	Description
hh.mm.ss	Current time. This field begins in column 2 and ends with a period. The system always appends this field in this format to the beginning of the message at the time it is entered into the account dayfile.

Message	Description
jobname	The name of the job which caused the entry of this message into the account dayfile. The field begins in column 11 and ends in column 17. The system appends this field to the beginning of the message along with the time.
o	A single character in column 18 which describes the origin type of the job. The system automatically appends this character when the message is entered into the account dayfile. The following origin types are specified.

Character	Origin Type
S	System
B	Batch
T	Terminal
E	Remote batch
M	Multiterminal

Message	Description
geac	A unique four-character message identifier which defines the particular activity identified. The field begins in column 21 and ends with a comma-blank (,) . The first character identifies the information group, the second character identifies the event which caused the message to be entered into the account dayfile, and the last two characters identify the activity being recorded. This field is further described in section 1.5.1.
additional information	Information that gives further detail to the activity identified by geac. The field begins in column 27 and ends with a period. This field is further described in section 1.5.1.

These message lines are free format. That is, each field defined in a message ends in either a comma-space (,) , or a period (.) if it is the last field in the message. If a field is not used, it appears only as a comma-space, or it does not appear if it is the last field in the message. The separator immediately follows the last character of the field. The field size can be any length, depending on the information being supplied.

Example:

SPAT, filea, , packnam.

No alternate user access was specified during the permanent file attach operation.

SPAT, filename.

No pack name was required and no alternate user access was specified during the permanent file attach operation.

Refer to section 1.5.1 for a description of this message.

1.5.1 MESSAGE GROUPS

All account dayfile entries are grouped by a particular information type. Each group type is identified by the first character of the geac field (that is, g of geac) in the account dayfile message. The following are the group types.

Type	Description
Aeac	Accounting information
Seac	Statistical information
Ueac	Job usage information

ACCOUNTING INFORMATION

The A message group provides the information necessary for accounting purposes. These messages include information denoting the beginning and end of an accounting sequence, as well as all resources used. The message formats are:

Aeac, additional information.

The events (e character of Aeac) are defined as follows:

Character	Event Description
B	The beginning of a job or session
C	Denotes a change in the accounting activity
D	Denotes disk activity
E	Denotes the end of a job or session
M	Denotes magnetic tape activity
P	Denotes permanent file activity
Q	Denotes a queue operation
R	Denotes a recovery operation
S	Denotes the suspension of a session

The activity identifier (ac of geac) identifies the information being recorded and is event dependent.

The following is a list of the begin event (B) messages.

```
ABCN, charge, project.
ABLQ, qt.
ABUN, usernum, family.
```

Identifier	Description
ABCN, charge, project.	Denotes the beginning of a charge sequence: charge — 1- to 10-alphanumeric character charge number project — 1- to 20-alphanumeric character project number
ABLQ, qt.	Denotes the placing of the job into the I/O queue: qt — Type of queue: IN Input queue PN Output punch queue PR Output print queue
ABUN, usernum, family	Denotes the beginning of a user number usernum under the permanent file family family

The following is a list of the change event (C) activity messages.

```
ACCN, charge, project.
ACSR, sruunits.
ACUN, usernum, family.
```

Identifier	Description
ACCN, charge, project.	Denotes a change of charge with the charge number charge and the project number project
ACSR, sruunits.	Denotes the end of an accounting block that used sruunits of SRUs. The SRU accumulator is displayed and cleared and is associated with the entering of a new charge/project number.
ACUN, usernum, family.	Denotes the change to a user number usernum under the permanent file family family

The following is a list of the disk event (D) activities.

```
ADDR, es, family, dn.
ADPD, es, packnam, usernum.
ADPM, es, packnam, usernum.
```

Identifier	Description
ADDR, es, family, dn.	Denotes the recovery of a permanent file device with family name family and device number dn on equipment es
ADPD, es, packnam, usernum.	Denotes the dismounting of the auxiliary removeable disk pack with the user number usernum from the device es with pack name packnam. If the device is a public pack, the usernum field is not used.

ADPM, es, packnam, usernum.	Denotes the mounting and recovery of the auxiliary disk pack with user number usernum on the device es with pack name packnam. If the device is a public pack, the usernum field is not used.

The following is a list of the end event (E) activities.

```
AENR, usernum, family.
AEPQ, qt.
AESR, sruunits,
AEUN, usernum, family.
```

Identifier	Description
AENR, usernum, family.	Denotes the discarding of user usernum on family family from the TELEX recovery file
AEPQ, qt.	Denotes the purging of the job on the I/O queue: qt — Type of queue: IN Input queue PN Output punch queue PR Output print queue
AESR, sruunits.	Denotes the end of a job or session that used sruunits SRUs under the current charge number
AEUN, usernum, family.	Denotes the end of a user number activity (for Export/Import log-off operation)

The following is a list of the magnetic tape event (M) activities.

```
AMAS, es, vsnnum.
AMRT, es, onx.
AMRT, es, C1, cccccccc.
AMRT, es, C2, cccccccc.
```

Identifier	Description
AMAS, es, vsnnum.	The magnetic tape equipment es is assigned, with a volume serial number vsnnum. If the tape is unloaded, vsnnum is not used.
AMRT, es, onx.	Denotes magnetic tape equipment es returned from the user. onx specifies the type of drive returned and is specified as 657 or 659.
AMRT, es, C1, cccccccc. AMRT, es, C2, cccccccc.	Denotes magnetic tape equipment as returned from the user. These messages denote 667/669 tape equipment. Two lines are issued to display the cumulative status cccccccc, parts 1 and 2.

The following is a list of the permanent file event (P) activities.

```
APDF, filenam, usernum, packnam, size.
APPG, filenam, usernum, packnam.
APPN, packnam.
APRP, filenam, usernum, packnam, size.
APSV, filenam, usernum, packnam, size.
```

Identifier	Description
APDF, filenam, usernum, packnam, size.	Denotes a permanent file defined
APPG, filenam, usernum, packnam.	Denotes a permanent file purged
APPN, packnam.	Denotes entering the default pack name. If no pack name is specified, the message denotes the clearing of the default pack name.
APRP, filenam, usernum, packnam, size.	Denotes a permanent file replaced
APSV, filenam, usernum, packnam, size.	Denotes a permanent file saved

These parameters refer to the above permanent file event (P) activities.

Parameter	Description
filenam	The file name
usernum	The user number of an alternate user accessing a file. If there is no alternate user, this field is not used.
packnam	The name of the auxiliary device, if one was specified. If not specified, the field is not used.
size	The size of the file in PRUs. This field is not currently used, but is available for future expansion.

The following is a list of the recovery event (R) activities.

```
ARSY, ln, yy/mm/dd.
ARUN, usernum, family, tn.
```

Identifier	Description
ARSY, ln, yy/mm/dd.	Denotes the recovery of the system at recovery level ln on the date yy/mm/dd
ARUN, usernum, family, tn.	Denotes the recovery of a time-sharing session with user number usernum, family name family, and terminal number tn

The following is a list of the suspend event (S) activities.

```
ASSR, xxxxxx.xxx UNTS.
ASDx, usernum, family.
ASNx, usernum, family.
ASTx, usernum, family.
```

Identifier	Description
ASSR, xxxxxx.xxxUNTS.	Denotes the suspension of a job or session that used the indicated amount of SRUs under the current charge number
ASDx, usernum, family.	Denotes that the user number is duplicated in the time-sharing recovery file. The current session is ignored.
ASNx, usernum, family.	Denotes that the user has not been successfully entered into the time-sharing recovery file
ASTx, usernum, family.	Denotes that the user has been saved in the time-sharing recovery file

x	Description
H	The recovery operation was caused by a hang up.
R	The recovery operation was caused by the recovery of the time-sharing sybsystem.
T	The recovery operation was caused by a session timeout.

STATISTICAL INFORMATION

The S message group provides information relating to the various activities of the system. The message formats are:

Seac, additional information.

The events (e character of Seac) are defined as follows:

Character	Event Description
A	Subsystem abort
B	Subsystem begin
C	Accumulator displayed and cleared
D	Accumulator displayed and continued
E	Subsystem end
I	Informative message
P	Permanent file information
R	Subsystem recovery

The following is a list of the C event activities.

```
SCMT, ex, pppppppp.
```

Identifier	Description
SCMT, es, pppppppp.	Denotes the number of magnetic tape PRUs transferred from unit es

The following is a list of the D event activities.

```
SDCA, seconds.
SDCI, seconds.
```

Identifier	Description
SDCA, seconds.	Denotes system CPU time active
SDCI, seconds.	Denotes system CPU time idle

The following is a list of the informative (I) activities.

```
SIUN, usernum.
```

Identifier	Description
SIUN, usernum.	Denotes attempt to enter illegal user number usernum or secondary usernumber usernum while secondary user card feature is disabled

The following is a list of the permanent file event (P) activities.

```
SPAP, filenam, usernum, packnam.
SPAT, filenam, usernum, packnam.
SPCG, filenam, usernum, packnam.
SPCT, filenam, usernum, packnam.
SPDF, filenam, usernum, packnam.
SPGT, filenam, usernum, packnam.
SPPG, filenam, usernum, packnam.
SPPM, filenam, usernum, packnam.
SPRP, filenam, usernum, packnam.
SPSV, filenam, usernum, packnam.
```

Identifier	Description
SPAP, filenam, usernum, packnam.	Denotes permanent file APPEND operation
SPAT, filenam, usernum, packnam.	Denotes permanent file ATTACH operation
SPCG, filenam, usernum, packnam.	Denotes permanent file CHANGE operation
SPCT, filenam, usernum, packnam.	Denotes permanent file CATLIST operation
SPDF, filenam, usernum, packnam.	Denotes permanent file DEFINE operation

Identifier	Description
SPGT, filenam, usernum, packnam.	Denotes permanent file GET operation
SPPM, filenam, usernum, packnam.	Denotes permanent file PERMIT operation
SPPG, filenam, usernum, packnam.	Denotes permanent file PURGE operation
SPRP, filenam, usernum, packnam.	Denotes permanent file REPLACE operation
SPSV, filenam, usernum, packnam.	Denotes permanent file SAVE operation

The remainder of the S group messages are not used; they are reserved for future expansion.

USAGE INFORMATION

The U message group provides a breakdown of the usage of the system for a particular user. The message formats are:

Ueac, usage count descriptor.

The events (e character of Ueac) are defined as follows:

Character	Event Description
C	Accumulator displayed and cleared
D	Accumulator displayed and continued
E	Accumulator displayed and ended
S	Accumulator suspended

The activities (ac of Ueac) are defined as follows:

Characters	Activity Description
AD	SRU adder accumulator
CI	Characters transmitted into the system
CO	Characters transmitted out of the system
CP	CPU time
CR	Cards read
LP	Lines printed
MS	Mass storage activity
MT	Magnetic tape activity
PC	Cards punched
PF	Permanent file activity

Each accumulator is displayed in F10.3 format with a four-character unit descriptor after the value. The following are the descriptors.

Descriptor	Value
KCHS	Kilo-characters
KCDS	Kilo-cards
KLNS	Kilo-lines
KPRS	Kilo-PRUs
KUNS	Kilo-units
SECS	Seconds
UNTS	Units

The following is a list of the C event activities.

```
UCCI, xxxxxx.xxxKCHS.
UCCO, xxxxxx.xxxKCHS.
UCCR, es, xxxxxx.xxxKCDS.
UCLP, ex, xxxxxx.xxxKLNS.
UCPC, es, xxxxxx.xxxKCDS.
```

Identifier	Description
UCCI, xxxxxx.xxxKCHS.	The number of characters transferred into the system for a job (for time-sharing subsystem currently)
UCCO, xxxxxx.xxxKCHS.	The number of characters transferred out of the system for a job (for time-sharing subsystem currently)
UCCR, es, xxxxxx.xxxKCDS.	The number of cards read into the system for a job on equipment number es
UCLP, ex, xxxxxx.xxxKLNS.	The number of lines printed for a job on equipment number es
UCPC, ex, xxxxxx.xxxKCDS.	The number of cards punched for a job on equipment number es

The following is a list of the D event messages.

```
UDAD, xxxxxx.xxxUNTS.
UDCI, xxxxxx.xxxKCHS.
UDCO, xxxxxx.xxxKCHS.
UDCP, xxxxxx.xxxSECS.
UDMS, xxxxxx.xxxKUNS.
UDMT, xxxxxx.xxxKUNS.
UDPF, xxxxxx.xxxKUNS.
```

Identifier	Description
UEAD, xxxxxx.xxxUNTS.	Denotes the SRU adder accumulator for a job
UDCI, xxxxxx.xxxKCHS.	Denotes the number of characters transferred into the system for a job (currently available for time-sharing only)
UDCO, xxxxxx.xxxKCHS.	Denotes the number of characters transferred out of the system for a job (currently available only for time-sharing)
UDCP, xxxxxx.xxxSECS.	Denotes the CPU time for a job
UDMS, xxxxxx.xxxKUNS.	Denotes the mass storage activity accumulator for a job
UDMT, xxxxxx.xxxKUNS.	Denotes the magnetic tape activity accumulator for a job
UDPF, xxxxxx.xxxKUNS.	Denotes the permanent file activity accumulator for a job

The following is a list of the end job event (E) messages.

```
UEAD, xxxxxx.xxxUNTS.
UECI, xxxxxx.xxxKCHS.
UECO, xxxxxx.xxxKCHS.
UECP, xxxxxx.xxxSECS.
UEMS, xxxxxx.xxxKUNS.
UEMT, xxxxxx.xxxKUNS.
UEPF, xxxxxx.xxxKUNS.
```

Identifier	Description
UEAD, xxxxxx.xxxUNTS.	Denotes the SRU adder accumulator for a job.
UECI, xxxxxx.xxxKCHS.	Denotes the number of characters transferred into the system for a job (currently available only for time-sharing)
UECO, xxxxxx.xxxKCHS.	Denotes the number of characters transferred out of the system for a job (currently available only for time-sharing)
UECP, xxxxxx.xxxSECS.	Denotes the CPU time for a job
UEMS, xxxxxx.xxxKUNS.	Denotes the mass storage activity accumulator for a job
UEMT. xxxxxx.xxxKUNS.	Denotes the magnetic tape activity accumulator for a job
UEPF, xxxxxx.xxxKUNS.	Denotes the permanent file activity accumulator for a job

The following is a list of the suspend event (S) messages.

```
USAD, xxxxxx.xxxUNTS.
USCP, xxxxxx.xxxSECS.
USMS, xxxxxx.xxxKUNS.
USMT, xxxxxx.xxxKUNS.
USPF, xxxxxx.xxxKUNS.
```

Identifier	Description
USAD, xxxxxx.xxxUNTS.	Denotes the SRU accumulator for a job
USCP, xxxxxx.xxxSECS.	Denotes the CPU time for a job
USMS, xxxxxx.xxxKUNS.	Denotes the mass storage activity accumulator for a job
USMT, xxxxxx.xxxKUNS.	Denotes the magnetic tape activity accumulator for a job
USPF, xxxxxx.xxxKUNS.	Denotes the permanent file activity accumulator for a job

The usage summary for a user is always issued in the following order.

```
UeCI, xxxxxx.xxxKCHS. (time-sharing users only)
UeCO, xxxxxx.xxxKCHS. (time-sharing users only)
UeAD, xxxxxx.xxxUNTS.
UeCP, xxxxxx.xxxSECS.
UeMS, xxxxxx.xxxKUNS.
UeMT, xxxxxx.xxxKUNS.
UePF, xxxxxx.xxxKUNS.
geSR, xxxxxx.xxxUNTS.
```

g and e are the group and event identifiers described in this section.

1.6 SYSTEM DAYFILE DUMPING ROUTINES

Routines are available for dumping the account dayfile, the error log dayfile, and the system dayfile. A job using any of these routines must either be a system origin job, or the user must have system origin privileges and DEBUG must be set at the console. The following format is used when calling these routines.

name(lfn, strng, op, pd, pl)

or

name(L=lfn, FR=strng, OP=op, PD=pd, PL=pl)

The following values may be entered for each of these mnemonics.

Parameter	Description
name	Type of dayfile dump to be performed. The following are acceptable values. AFD Account dayfile is dumped. ELD Error log dayfile is dumped. DFD System dayfile is dumped.

Parameter	Description
L=lfn	This name specifies the file on which the dayfile dump is placed. If no file name is specified, the default value is L=OUTPUT. Pagination occurs if L=OUTPUT, or if the PD and PL parameters are set.
FR=strng	This specifies the search string for selective dayfile dumping. The dayfile is searched for this string in the starting position of the field specified in the OP parameter.
OP=op	Specifies the dump option. This value may be one of the following. T The time field in the dayfile is searched for the string specified in the FR parameter. The dump begins from that point. J The jobname field in the dayfile is searched for the string specified in the FR parameter. The dump begins from that point. M The message field in the dayfile is searched for the string specified in the FR parameter. The dump begins from that point. I Specifies an incremental dump. The dayfile is dumped starting from the point of the last dayfile dump. P Specifies an incremental dump by user number. The dayfile is dumped starting from the point of the last dayfile dump by this user. F Specifies full dayfile dump If neither the FR nor the OP options are specified, the default value is OP=F. If a string is specified in the FR parameter, the default value is OP=M.
PD=pd	Specifies the print density. This may be specified as either 3, 4, 6, or 8 lines per inch. If omitted, the default value is PD=6.

Parameter	Description
PL=pl	Specifies the page length. The default is based on the print density as shown below.

Print Density	Default Page Length
3	30
4	40
6	60
8	80

PERMANENT FILE UTILITIES 2

Five utility processors maintain the NOS permanent file system. This maintenance includes the dumping and loading of permanent files, the cataloging of files in the system and on an archive tape, and the copying of specific files to a control point. The designations and functions of the five utilities are as follows:

PFDUMP	Dump permanent files. A permanent file utility that copies files stored on a permanent file device to a backup storage file (archive tape). Files created by this dump may be reloaded by the PFLOAD utility.
PFLOAD	Load permanent files. Load files from a backup storage file (archive tape) onto a permanent file device.
PFCAT	Catalog permanent file device. Catalogs permanent file catalog tracks and generates a report. The possible reports are: • Listing of catalog file with files grouped by user index • Statistical report of device usage
PFATC	Catalog archive tape. Catalogs permanent file archive file(s).
PFCOPY	Copy archive file(s) to control point. Extracts files from an archive file and copies them to a control point.

These five utility processors are overlays called by the permanent file supervisor (PFS). This supervisor cracks the parameters in the utility command and loads the correct processing overlay. The overlay performs its specific operation in interaction with the permanent file utility routine (PFU) which manages the catalogs, permits, data allocation on a device, and the data transfer between device and overlay. An overview of this procedure is given in Figure IV-2-1.

Operation of these utilities can be initiated from console, batch, or a terminal. However, in all cases the job containing the utility control statement must be system origin type or have system origin privileges with DEBUG on.

The call and operation of each of these utilities are explained in the following sections. These explanations require some familiarity with the following terms.

Archive File (Tape)

User Index

Device Mask

Mass Storage Table

Catalog Track

Catalog Entry

Track Reservation Table

Minimal definitions are given in the next section.

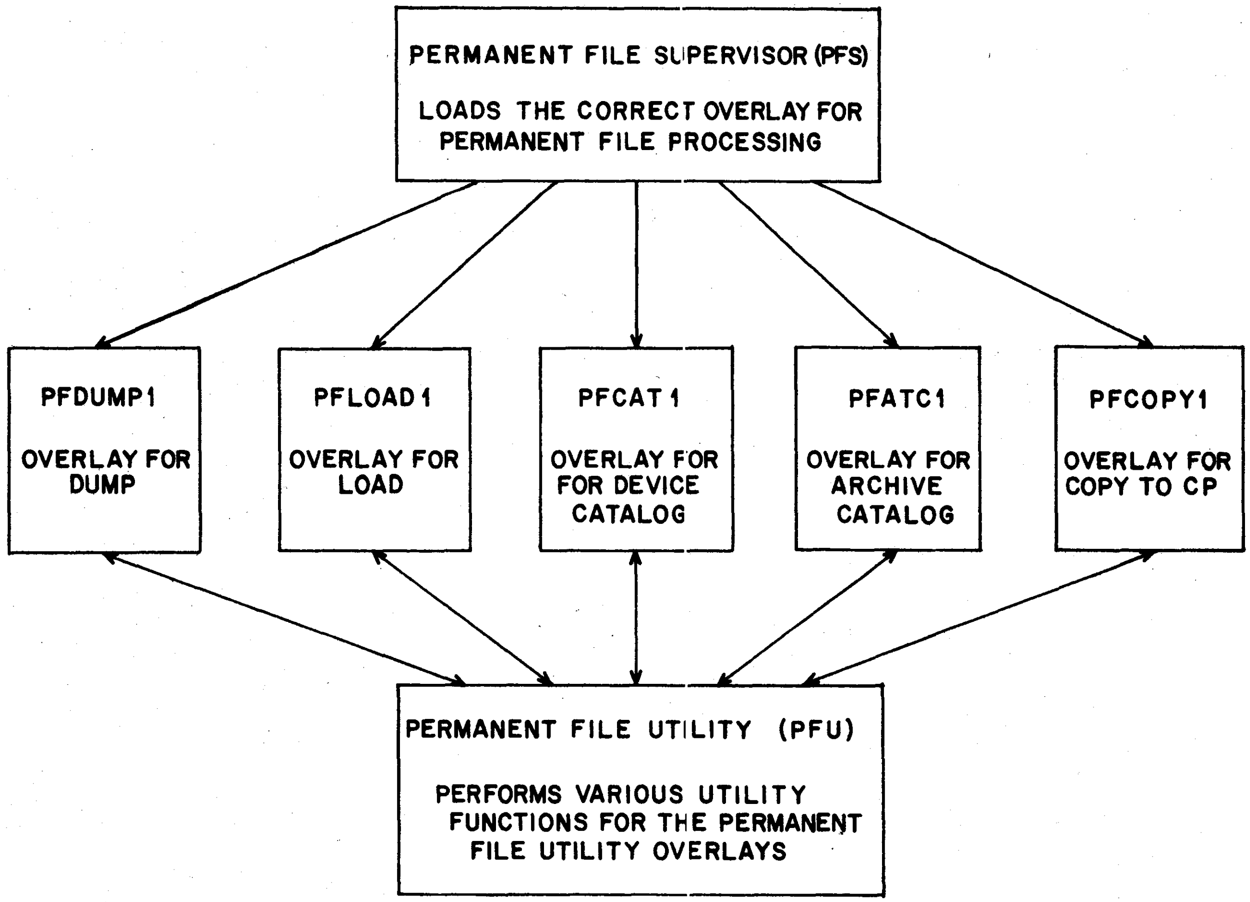


Figure IV-2-1. Functional Overview of the Permanent File Utility Overlays

2.1 DEFINITIONS

2.1.1 ARCHIVE FILE (TAPE)

The permanent files accumulated on mass storage can be dumped in whole or in part to a backup tape (or other mode of backup gear) as insurance against a device going down or to free a device temporarily as a scratch file for preventive maintenance. Each dump of permanent files is made to an archive file. Each archive file is part of a multirecord file in which each logical record represents one of the permanent files that was part of the dump that created that archive file.

If two or more archive files are created on one type of backup device (for example, tape), these archive files constitute a multifile archive file and can be so referenced by the parameters in the permanent file utility commands (Figure IV-2-2).

The archive file can be loaded back onto the permanent file system in whole or in part. It can also be used to generate reports, or the individual files can be referenced for selective use.

2.1.2 USER INDEX

A 17-bit user index is associated with each user number created on the VALIDUs† file. This index is entered through MODVAL (section 1.1) with the UI identifier on the user number input directive (/usernum, UI=xxxxxx) or MODVAL, by default supplies the next available index.

Whenever this user submits a job, the related user index is placed in the control point area along with the user number, job name, and other parameters that link hardware, files, and job. The permanent file manager uses the index to identify the master device and catalog track for this user. It does this with two masking operations involving the index and two sets of device parameters obtained from the mass storage table in CMR. The device parameters are device mask and number of catalog tracks.

One operation correlates the rightmost character in the user index (bits 0 through 2) with the bit settings of the device mask for each device in the configuration. The other operation performs a logical AND between the remaining portion of the index and the number of catalog tracks on the device to determine which track contains this user's catalog.

The identification of the master device is covered in the device mask definition; the identification of the catalog track on that device is covered in the following section.

The number of tracks on a device is established when the device is initialized or by default. Default values are the following.

† In this manual, the user validation and accounting files are referred to as VALIDUs, VALINDs, and PROFILa. Refer to Table IV-1-1 for a list of file names that correspond to the appropriate operating system levels.

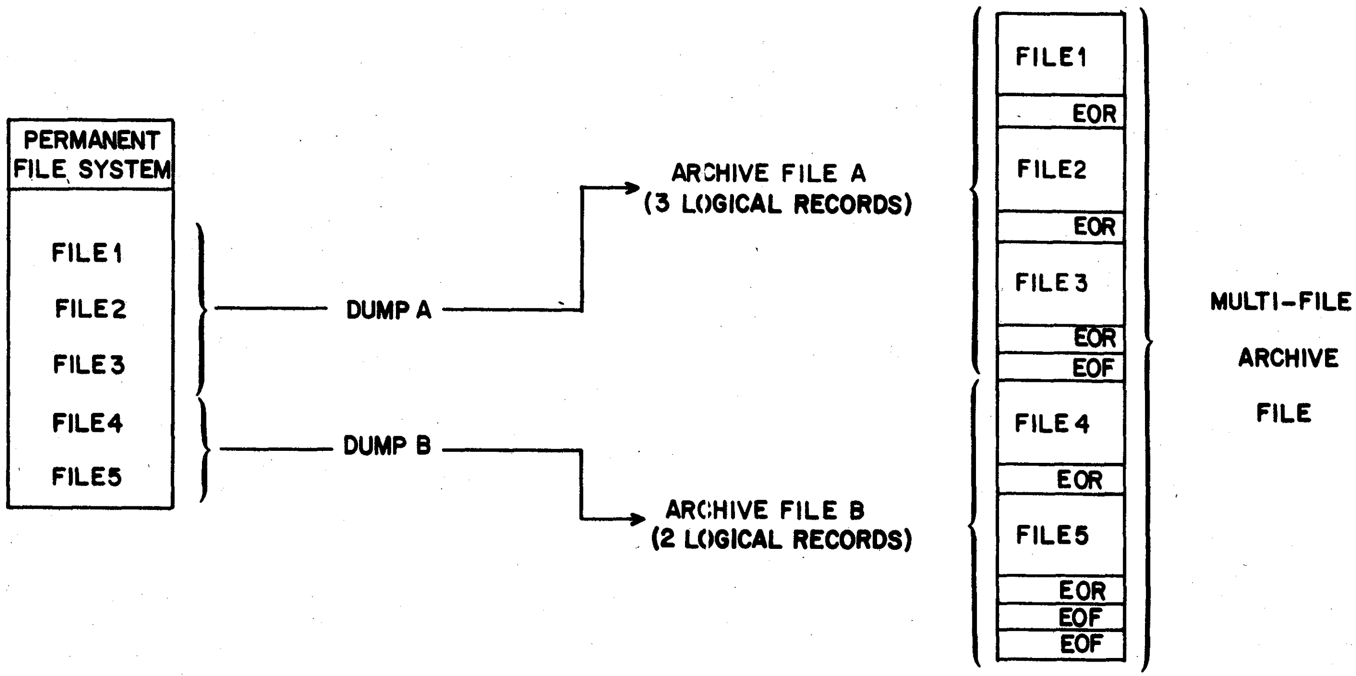


Figure IV-2-2. Example of Multifile Archive File Structure

Type of Device	Default Number of Catalog Tracks
DE ECS	2
DI 844	20
DP DDP path to ECS	2
MD 841	20
Private device	1

Whether initialized or defaulted, this number is a power of 2 up through 200B. Thus, the following are admissible numbers of tracks.

1, 2, 4, 10B, 20B, 40B, 100B, 200B

When these numbers are reduced by one, the following octal values are produced.

0, 1, 3, 7, 17B, 37B, 77B, 177B

In binary form, these values are:

```
        0
        1
       11
      111
    1 111
   11 111
  111 111
1 111 111
```

These serve as masks with all bits set (except for the first), and when ANDed with bits 3 through 9 of the user index, produce the track number for that particular user on that particular device.

Example:

Given a configuration of two devices with device masks 221B and 156B, and with 40B and 20B tracks respectively, a user who has the user index 14224 will have the assignment of master device and catalog track as illustrated in Figure IV-2-3.

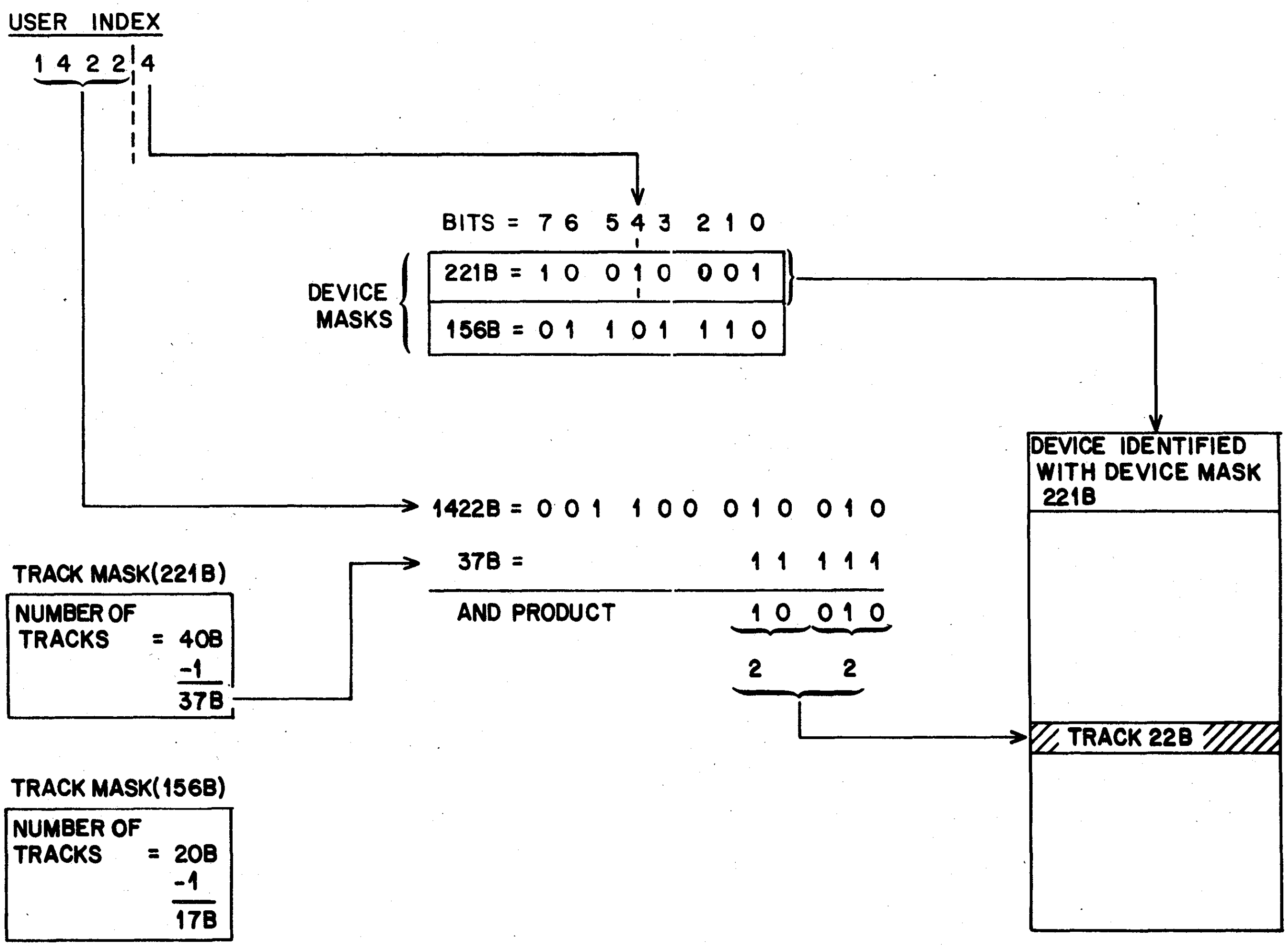


Figure IV-2-3. Example of Masking Operations with User Index

2.1.3 DEVICE MASKS

Two types of device masks exist for each device, the device mask and the secondary mask.

DEVICE MASK

An 8-bit quantity used to identify the group of users who have this particular device as their master device; that is, it is the device that contains their file catalogs, all their indirect access files, and may contain some or all of their direct access files. (The appropriate bit must be set in the secondary mask of the device, if a user's direct access files are to reside on that device. The direct access files are then put on the device with the most space available unless a device is specified by the user.)

If a bit is set in the mask, all user indices ending in that bit-position value have this device as their master device.

Example 1:

The device identified by the mask 261B will be the master device (have catalogs and permanent files) for users whose indices end in 0, 4, 5, and 7, since the bit setting in the mask is:

$$\begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{array}$$

$$\underbrace{\boxed{1}\,\boxed{0}}_{2}\;\underbrace{\boxed{1}\,\boxed{1}\,\boxed{0}}_{6}\;\underbrace{\boxed{0}\,\boxed{0}\,\boxed{1}}_{1}$$

Within a family, the sum of all the device masks must total exactly 377B and each bit must be accounted for only once.

Example 2:

For a family of three devices, appropriate device masks are:

```
1 0  0 1 0  0 0 1   =   221B
0 0  1 0 0  0 1 0   =   042B
0 1  0 0 1  1 0 0   =   114B
------------------      ----
1 1  1 1 1  1 1 1   =   377B
```

If the sum of the device masks is less than 377B, this means that one or more bits have not been accounted for and any user index ending in that ordinal will reference no device. When such a user tries to write a permanent file, he will get the message:

DEVICE UNAVAILABLE

Example 3:

For a family of four devices, the following masks were designated by the EQ entry at deadstart.

```
0 1  1 0 0  0 1 0   =   142B
0 0  0 1 0  0 0 0   =   020B
0 0  0 0 1  0 0 0   =   010B
1 0  0 0 0  1 0 0   =   204B
-----------------       ----
1 1  1 1 1  1 1 0   =   376B
```

Users whose index ends in zero have no master device.

If the sum of the masks for one family is greater than 377B, there is at least one duplication of master device for a single index.

Example 4:

For the following family of four devices:

```
1 0  0 0 1  0 1 0   =   212B
0 1  0 0 0  1 1 0   =   106B
0 0  1 0 0  0 0 0   =   040B
0 0  0 1 0  0 0 1   =   021B
-----------------       ----
1 1  1 1 1  1 1 1   =   401B
              1
```

User indices that end in 1 reference two master devices, one with the mask 212B and one with the mask 106B.

It is a necessary but not sufficient condition that all masks for one family total 377B. Consider the following example of three devices in one family.

Example 5:

```
1 0  1 1 0  0 0 1   =   261B
0 1  0 0 1  1 0 1   =   115B
0 0  0 0 0  0 0 1   =   001B
-----------------       ----
1 1  1 1 1  1 - -       377B
```

Indices that end in zero reference all three devices in the family. Indices that end in one reference no device.

When masks are assigned at deadstart, two rules must be observed.

1. The sum of the masks must be exactly 377B for each family.
2. Each bit position must be set exactly once for the devices in one family.

SECONDARY MASK

An 8-bit quantity used to identify the group of users who may place direct access files on this particular device. This quantity may range from 0 to 377B. Unlike the device mask, two or more devices in the same family may have the same bits set in the secondary mask. Also, no restriction is made regarding the sum of the secondary masks in a family.

If a bit is set in the secondary mask of a device, all user indices ending in that bit-position value may place direct access files on this device.

Example 1:

A device whose secondary mask is 072B can contain direct access files of users whose indices end in 1, 3, 4, and 5, since the bit setting in the secondary mask is:

7	6	5	4	3	2	1	0
0	0	1	1	1	0	1	0

$\underbrace{0\ 0}_{0}\ \underbrace{1\ 1\ 1}_{7}\ \underbrace{0\ 1\ 0}_{2}$

Example 2:

For a family of three devices, the following secondary masks exist:

7 6	5 4 3	2 1 0		
0 0	0 0 0	0 0 0	=	000B
0 1	0 1 1	0 0 1	=	131B
1 1	0 1 0	1 1 0	=	326B

Users whose indices end in 0, 3, 4, and 6 may place direct access files on the device whose secondary mask is 131B. Users whose indices end in 1, 2, 4, 6, and 7 may place direct access files on the device whose secondary mask is 326B. Users whose indices end in 4 or 6 can place direct access files on both of these devices. Users whose indices end in 5 are not allowed to use direct access files. No user is allowed to place direct access files on the device whose secondary mask is 000B.

2.1.4 MASS STORAGE TABLE

The configuration of mass storage devices currently available to the system is defined by the CMR mass storage table. Each logical device in this configuration has an entry in this table that contains the following information.

Device status

Number of current users

Number of catalog tracks

Device mask

Location of start of permission data

Location of start of catalog tracks

Location of start of indirect access files

Interlock status

Family (pack) name

Device number

User number for private auxiliary device

Available space on device

Logical description of device

These appear on the E, M display on the console.

2.1.5 CATALOG TRACK

Users are assigned by groups to catalog tracks on a permanent file device according to user index and device mask (refer to previous definitions). A catalog track contains the catalog entries (definition following) that locate and define each permanent file created by these users.

The device which contains a user's catalog track is his master device. This same master device contains all of the indirect access files created by this user; it may contain some or all of his direct access files. (Direct access files are put on a device in his family with the most space available unless the user specifies their destination. However, the appropriate bit must be set in the secondary mask of the device if the user's direct access files are to reside on that device.)

The end of a catalog track is indicated by an EOI on the device. All catalog tracks are linked in the track reservation table and appear as one logical chain. When catalog entries exceed the logical track, a continuation track is reserved at the end of the catalog track chain in the track reservation table. The disk linkage bytes for the overflowed track point to the continuation track.

2.1.6 CATALOG ENTRY

Files in the permanent file system are referenced by dynamically updated catalog entries on the catalog tracks (previous definition) of master devices. Whenever a user creates a direct or indirect access file, a catalog entry that specifies the characteristics of the file, access information about the file, and the location of the file is entered on the catalog track of his master device. As he modifies the file, the catalog entry is updated to reflect the modification.

The format of a catalog entry is as follows:

59	53	47	41	35	23	17	11 0
filename						user index	
file length					track		sector
random index				creation date and time			
access count				modification date and time			
ct	mode		dn	last access date and time			
file password							
program control word							

filename	Permanent file name
user index	User index of file creator
file length	Length of the file in PRUs. For direct access files, the length is determined by the TRT entries for the file. PFM (permanent file manager) gets the correct length from the TRT before returning a catalog entry to the caller. (Refer to PFM in the reference manual.)
track	Beginning track of the file
sector	Beginning sector of the file
random index	Random disk address of first permit sector
creation date and time	yymmddhhmmss in octal when this file was first entered on the permanent file system. The year (yy) is biased by 70.
access count	Total number of times this file has been accessed
modification date and time	yymmddhhmmss in octal when this file was last modified. The year (yy) is biased by 70. For direct access files this field is updated only when the file is attached in a modifiable mode.
ct	File category (private, semiprivate, or public)
mode	Mode of access for semiprivate and public files. 0 Write, read, execute, append, modify, and/or purge 1 Read and/or execute 2 Append 3 Execute 4 Negate previous permission 5 Modify 6 Read and/or execute, allow modify 7 Read and/or execute, allow append
dn	Device number (0 through 77_8). This is only present when a direct access file resides on a device other than the master device for the user. The device number identifies this other device which is in the same family of devices.
last access date and time	yymmddhhmmss in octal when this file was last accessed. The year is biased by 70.
file password	Optional password
program control word	User control information (FET+11)

2.1.7 TRACK RESERVATION TABLE

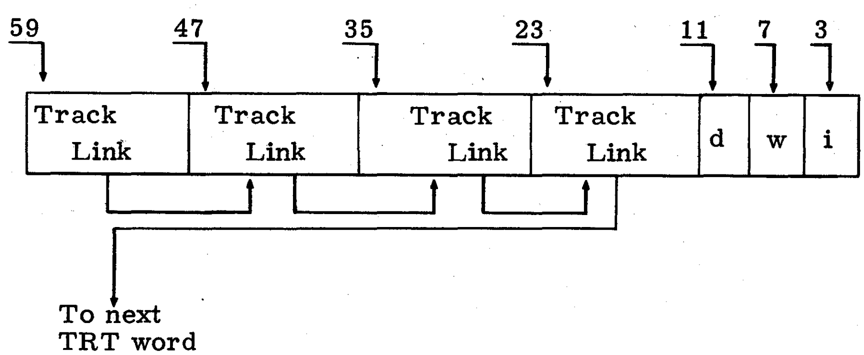
Every device in a permanent file system has a label track (usually track 0)† which contains a label sector describing the device (family name, device mask, secondary mask, location of permit information, catalog information, and indirect files) and a number of PRUs containing the track reservation table (TRT). The TRT describes the physical layout of data on the device and is the key to allocating information on the device.

The TRT contains single-word entries that define track linkage and bit-setting controls for those tracks (Figure IV-2-4). Each word has four 12-bit linkage bytes and three sets of 4-bit control settings that match the four bytes (Figure IV-2-5). Linkage format is given in Figure IV-2.6.

The numbering of the 12-bit linkage bytes or cells corresponds to the numbering of tracks on the device, with the first track starting at 4000. The entry in a cell references the next cell and its associated track. This next track either continues the information or starts a new sequence depending on the first-track bit setting.

Figures IV-2-4 and IV-2-5 show a sequential linkage of tracks; in Figure IV-2-5, from track 4000 to track 4001 to 4002 to 4003. This numerical sequence is purely illustrative. The linkage could just as well have been:

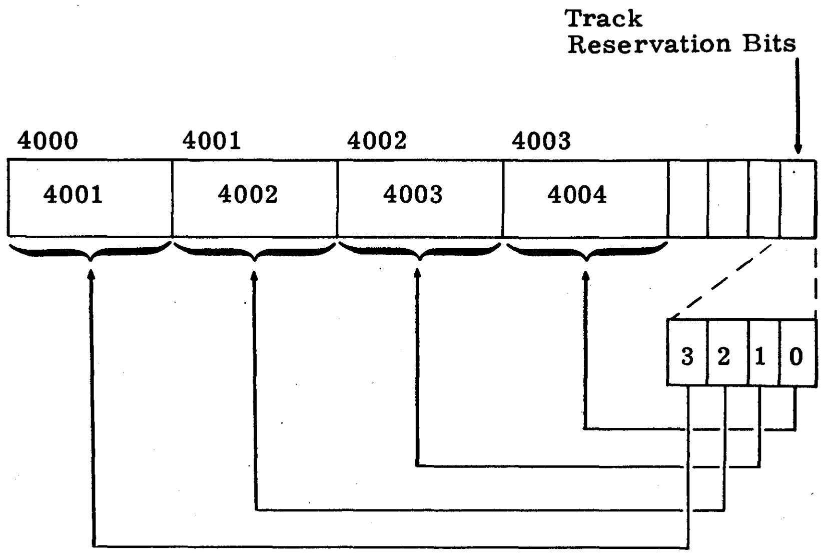
4000 ⟶ 4002 ⟶ 4015 ⟶ 4012



Track Link	Address of the next track that is a logical continuation of this file (The track links are shown as sequential within a word, but this is not a requirement.)
d	Bit settings for identifying the first track of a preserved file (permanent file chain or queued file)
w	Bit settings for establishing interlock of a track
i	Bit settings for track reservation

Figure IV-2-4. Track Reservation Table Word

†The label track is always track 0 for an 844 device.



Matching of four reserve track bit settings with corresponding track link bytes. The same correspondence holds for the interlock bits (4 through 7) and the first-track bits (8 through 11).

Figure IV-2-5. Bit Settings for Track Link Bytes

The first group of control settings (bits 8 through 11) is used to identify those tracks which begin a sequence of permanent file information. If any one of these bits is set, the associated track is the first track of a chain that may extend across a number of tracks. This chain can be a direct access file, an indirect access file data chain, a catalog chain, a permit chain, or a queued file.

The second group of control settings (bits 4 through 7) is used to interlock tracks. If any one of these bits is set, the associated track cannot be accessed so long as this bit remains set.

Interlocking capability is necessary since the permanent file manager may be processing several requests directed at the same file simultaneously. Without interlock, these requests could overlap.

Example:

Three indirect access file (FILE1, FILE2, and FILE3) are stored sequentially on a master device. User AA enters the command:

GET,FILE2

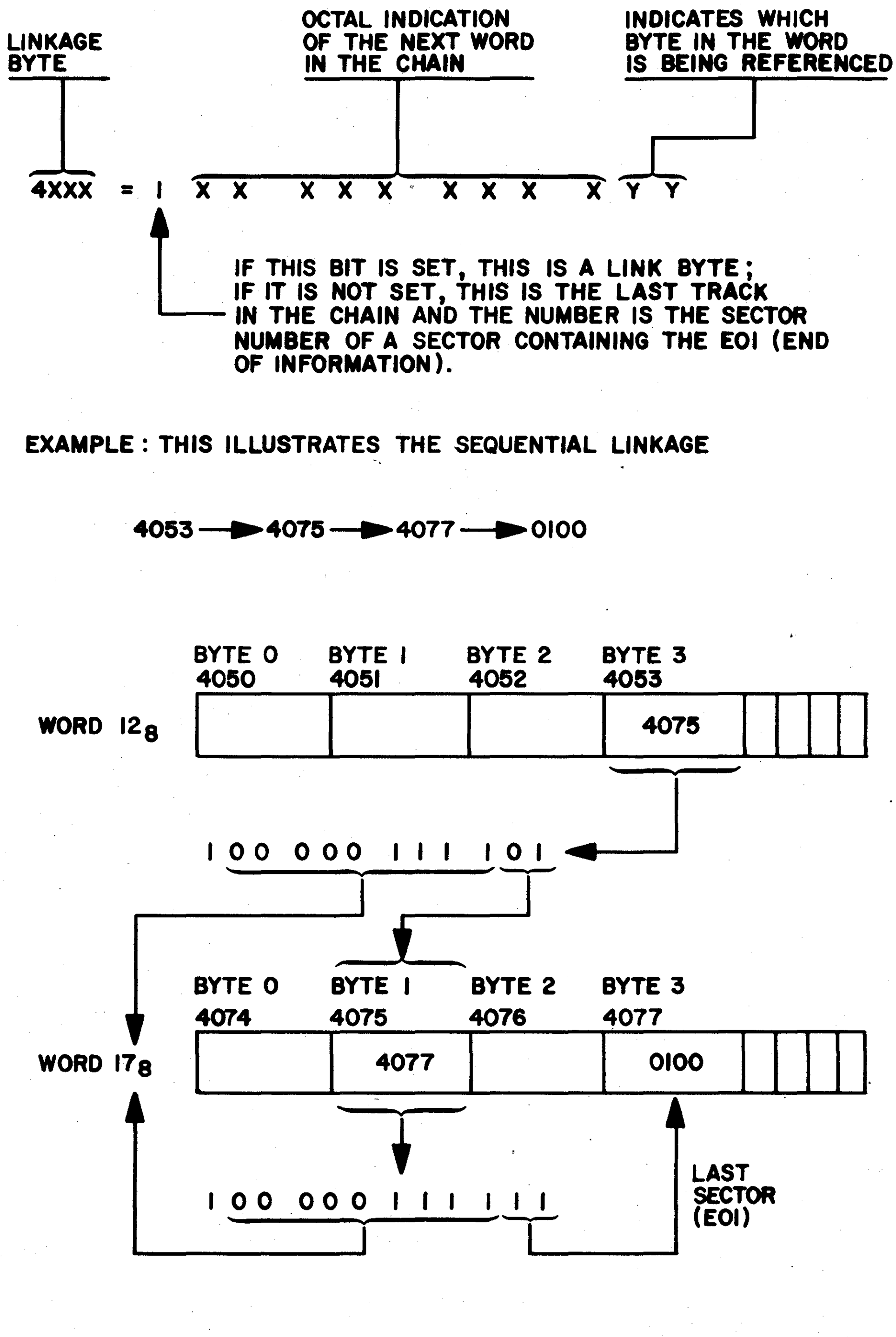


Figure IV-2-6. Linkage Format with Example

The system initiates retrieval to produce a working copy. Halfway through the retrieval, user BB enters the command:

PURGE, FILE2

This leaves a hole where FILE2 resided. User CC then enters the command:

SAVE, FILE4

FILE4 is put in the hole between FILE1 and FILE3, the area where the GET is being processed. Accordingly, user AA ends up with the first half of FILE2 followed by data from FILE4.

The previous problem is avoided by interlocking the catalog track that contains a file whenever it is accessed. Interlocking the track rather than the whole device allows users of another group (catalog track) to access their files. [It is possible to interlock the entire device, as in PFLOAD (section 2.4.2).]

The third group of control settings (bits 0 through 3) is used to identify reserved tracks. A track is reserved either because it has data written on it or it is a flawed track. A flawed track is removed from availability by reservation at deadstart or by subsequent initialization.

2.1.8 FAMILY

The family concept is a means of grouping users and their files by identifying the devices available to these users with a family name. Usually, the grouping is within the system on which these users normally run. By identifying devices with a family name, these devices can be connected to any system and still relate to the same users and files within that family.

An auxiliary device is a device that is not included in any family and can be accessed by validated users from any family. It provides the user or group of users with an alternative to his normal master device for storing and accessing permanent files. Use of auxiliary devices enables an installation to provide special sets of permanent files for selected users or for designated periods. As an example, an auxiliary device could be made available from 1200 to 1700 every day for any group of users who has proper validation.

A permanent file device is either part of a family or it is an auxiliary device. A family is a collection of 1 to 63 logical devices identified by a 1 through 7 character family name. An auxiliary device is a single device identified by a 1 through 7 character pack name. Files on a family device are accessed through user catalogs on a master device within the family; an auxiliary device holds all the catalogs that reference its files and as such is a self-contained entity. Families and auxiliary devices are defined at initialization time in the PF entry or by the MSI parameters, FN and DN.†

Normally, a system runs with one family available. Additional families can be defined or introduced (on removable devices) on the same system. If more than one family is available on a system, the user supplies the family name at login or on the USER statement with a batch job. The default family is used if the name is not supplied.††

The following example (Figure IV-2-7) considers a typical application of families.

One configuration with six permanent file devices (EST ordinals 0 through 5) is identified as SYSTEM A. Three of the devices are grouped into a family with the name FAMA. They have device numbers 40, 41, and 42. The remaining three ordinals have been defined as removable. This means that the system will allow family and auxiliary devices to be introduced on these equipments during system operation.

A second configuration with six permanent file devices is identified as SYSTEM B. Three of the devices have been grouped into a family with the name FAMB. They also have device numbers 40, 41, and 42. The remaining three pieces of equipment have been defined as removable.

Two auxiliary devices are available to both systems. These have the pack name AUX1 and AUX2.

Device masks for FAMA will total 377. Device masks for FAMB will total 377. (Refer to section 2.1.3.)

Users of FAMA would normally run jobs on SYSTEM A. Users of FAMB would normally run jobs on SYSTEM B.

---

† Refer to the INITIALIZE DSD command in the operator's guide.
†† The default family name is specified at deadstart time and cannot be changed dynamically.

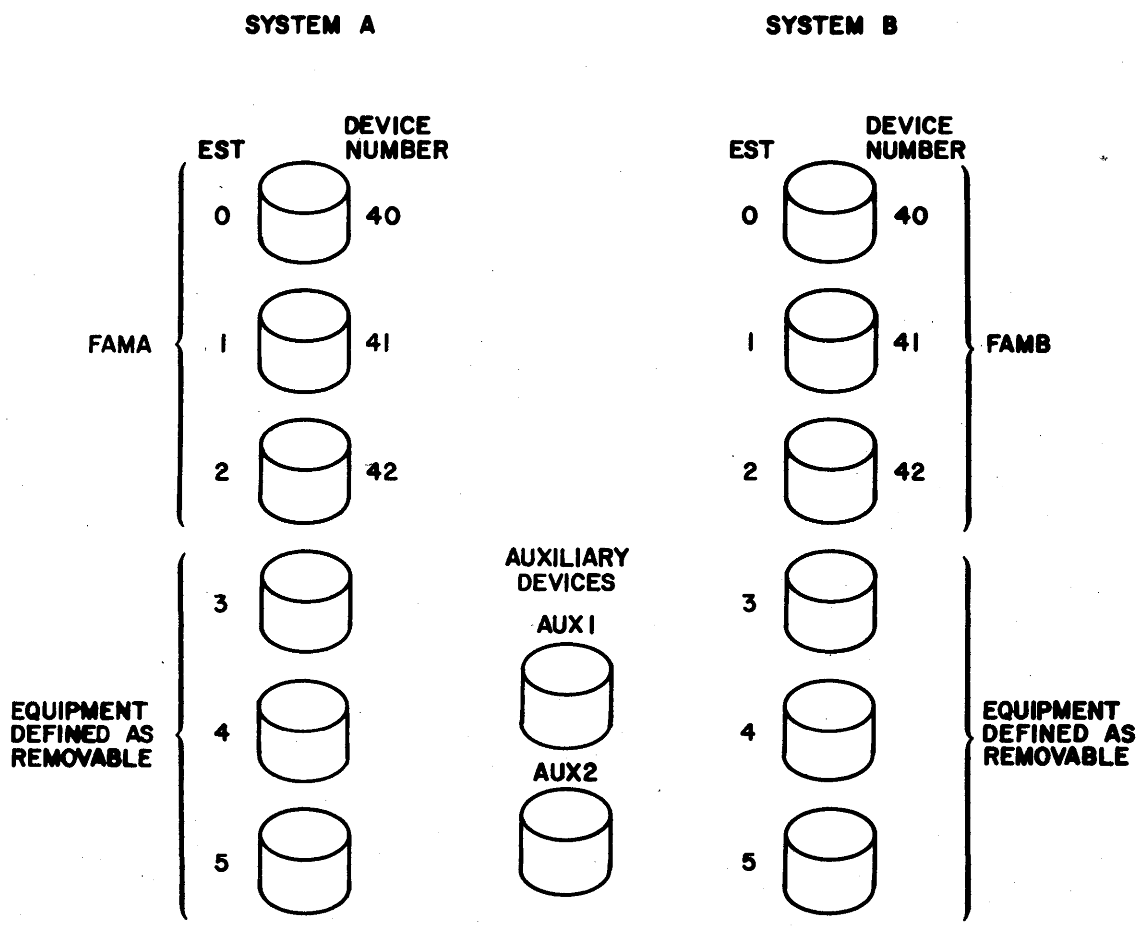


Figure IV-2-7. Example of Families in Two Systems

The removable equipment on SYSTEM A (EST 3, 4, and 5) could be used for two purposes:

1. To mount auxiliary devices AUX1 and/or AUX2 as required by users of FAMA. If users of FAMB need access to AUX1 and/or AUX2, these auxiliary devices will have to be mounted on some combination of EST 3, 4, and 5 on SYSTEM B.

2. To mount FAMB devices when they are no longer accessible through SYSTEM B. In this case, users of FAMB will have to transfer their access to SYSTEM A. This transfer could involve a physical transportation of disk packs from EST 0, 1, and 2 in SYSTEM B to EST 3, 4, and 5 in SYSTEM A, or SYSTEM A could already have alternate channel connections to EST 0, 1, and 2 in SYSTEM B but would define them as EST 3, 4, and 5 in its own system. In order for the users of FAMB to access their files through SYSTEM A, they would either submit their jobs to SYSTEM A or dial into SYSTEM A using a different telephone number. (They would dial the same number if switching of the communication gear is done.)

   Assuming that access to all of FAMB is transferred from SYSTEM B to SYSTEM A, the new array of SYSTEM A will be as illustrated in Figure IV-2-8. SYSTEM A now has two families: its original default family FAMA and the newly attached FAMB whose devices now have the EST ordinals 3, 4, and 5. Its device numbers (40, 41, and 42) remain the same. These happen to be the same as those used within FAMA but device numbers only provide uniqueness between devices within a family. Accordingly, total uniqueness of a device is provided by the combination of family name and device number.

   Users of FAMA will still be able to access their files without specifying the family name since FAMA is still the default for SYSTEM A. However, users of FAMB will have to specify to SYSTEM A that they belong to FAMB either at login or on the USER statement.

   If another equipment (EST 6) were available on SYSTEM A, either auxiliary pack AUX1 or AUX2 could be mounted on it and users from either family could access files on this pack.

A system origin job can use the FAMILY control statement to change the family name associated with the job. The control statement format is:

FAMILY(familyname)

familyname	1- to 7- character name of a family of permanent file devices

If an alternate family of permanent file devices is introduced into the system without a VALIDUs file, the job to create VALIDUs could include a FAMILY statement identifying the family. If the familyname parameter is omitted, the default family name is used. If the normal system has a default family name but the alternate system does not, the user can specify the alternate system by submitting the following statement.

FAMILY,0.

If the FAMILY statement is included in any nonsystem origin job, the job is aborted and the system issues the following message to the user's dayfile.

ILLEGAL CONTROL CARD.

If the default family is required but not known to the user, a

FAMILY.

statement sets the default family name.

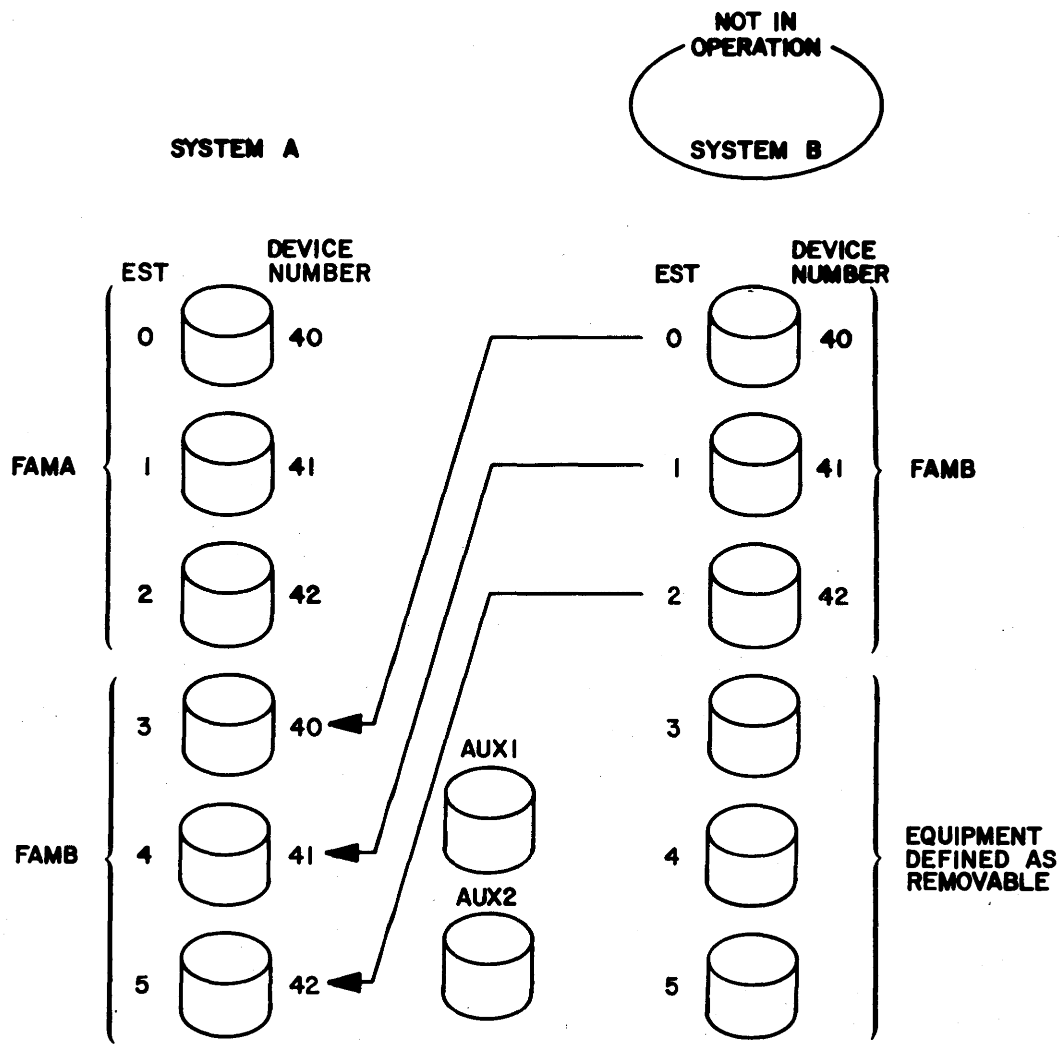


Figure IV-2-8. Example of Transfer of Family Access

2.2 CALLING THE UTILITIES

Any of the permanent file utilities can be called from console (K display), card input (batch), or terminal. For both batch and terminal use of the permanent file utilities, the user must have system origin privileges and DEBUG must be on.

2.2.1 INPUT FROM CONSOLE

Input from the console requires the following sequence of entries and responses.

1. The analyst types in:

   X. PFS. CR

   The B display indicates the control point to which PFS is assigned. To the right of the entry appears the flashing message:

   REQUEST *K* DISPLAY

2. The analyst types in:

   K, n. CR

   where n is the control point number noted on the B display. The permanent file utilities left K display (Figure IV-2-9) appears on the left screen.

3. The analyst types in:

   KK. CR

   This brings the permanent file utilities right K display (Figure IV-2-10) to the right screen. This will give a description of the permanent file utility options available.

4. The request for the desired utility is entered with:

   K. uo. CR

   where uo is one of the following.

uo.	Description
DU	Dump permanent files
LD	Load permanent files
CA	Catalog permanent files
AT	Catalog archive tape
CP	Copy archive file to control point

```
*** PERMANENT FILE UTILITY OPTIONS ***

                 CURRENT
OPTION           VALUE          DESCRIPTION

  UT                            UTILITY DESIRED









ENTER K.UO.    WHERE *UO* IS
   AT - TAPE CAT     CA - DEVICE CAT
   CP - TAPE COPY    DU - DUMP    LD - LOAD
```

Figure IV-2-9. Permanent File Utilities Left K Display (before UT is entered)

```
*** PERMANENT FILE UTILITY OPTION DESCRIPTION ***

OPTION                    DESCRIPTION

UT *UTILITY*              *LD* - PFLOAD
                          *DU* - PFDUMP
                          *CA* - PFCAT
                          *AT* - PFATC
                          *CP* - PFCOPY

LO *LIST OPTION*          T - TAPE CATALOGS
                          C - PERMANENT FILE
                              DEVICE CATALOGS
                          E - ERRORS
                          S - SUMMARY

OP *UTILITY OPTIONS*      C - CREATION*
                          A - LAST ACCESS*
                          M - LAST MODIFICATION*
                          I - INDIRECT ACCESS*
                          D - DIRECT ACCESS**
                          B - BEFORE
                          P - PURGE AFTER DUMP
                          R - REPLACE
                          N - NONINITIAL LOAD
                          Q - ADD CATALOG AND PERMIT RECOVERY
                          E - EXTRACT CIR
                          O - OMIT CIR

NOTE: OPTIONS ARE ENTERED AS FOLLOWS -

      K.LO = TCS.  or
      K.OP = CIBP

 *ONLY ONE TYPE OF DATE MAY BE SPECIFIED
**ONLY ONE ACCESS OPTION MAY BE SPECIFIED
```

Figure IV-2-10. Permanent File Utilities Right K Display

The parameters available under the chosen utility appear on the left screen. Figure IV-2-11 shows the left screen after the DU utility has been called (1). The TCE appearing after the LIST OPTION (2) indicates that

T Files processed

C Catalog files

E Errors

are the listings available under dump. Different combinations of list options are available with the other utilities.

The CAMIDBP (3) appearing after the UTILITY OPTION indicates that:

C Creation

A Last access

M Last modification

I Indirect access files

D Direct access files

B Before date and time

P Purge after dump

are the options available for the dump utility. Different combinations of utility options are available with the other utilities.

5. The desired parameters are entered with the command

   $K, p_1, p_2, \ldots, p_n$ (CR)

   where the p_i are selected from the parameter list on the left screen. The selected parameters replace the default values listed on the left screen. All parameters are described in detail in section 2.3.1.

6. To initiate execution, the analyst types:

   K.GO. (CR)

   The lower lines of the left display (4) disappears and is replaced by:

DEVICE	MASK
DATE	TIME

   Values appear with these identifiers as processing continues.

2.2.2 BATCH INPUT

The deck structure for statement input to call a permanent file utility is

JOB statement

USER(usernumber,password,familyname)

PFutil($p_1, p_2, \ldots, p_n$)

where util is one of the following utility options.

*** PERMANENT FILE UTILITY OPTIONS ***

OPTION		CURRENT VALUE	DESCRIPTION
UT		DU	UTILITY DESIRED ①
FM	=	UNDEFINED	FAMILY NAME
PN	=	0	PACK NAME
DN	=	0	DEVICE NUMBER
T	=	TAPE	ARCHIVE FILE NAME
LO	=	0	LIST OPTIONS VALID - TCE ②
L	=	OUTPUT	OUTPUT FILE NAME ③
OP	=	0	UTILITY OPTION VALID - CAMIDBP
NT	=	7-TRACK	NINE TRACK
DE	=	0	0-DFLT, 1-556, 2-200, 3-800, 4-1600
NR	=	0	NO REWIND
NU	=	0	NO UNLOAD
SF	=	0	SKIP FILES
N		--NA--	NUMBER OF FILES TO PROCESS
DT	=	0	DATE YYMMDD
TM	=	0	TIME HHMMSS
UI	=	0	USER INDEX
PF	=	0	PERMANENT FILE NAME
VF	=	PFVER	VERIFY FILE NAME
V	=	0	VERIFY FILE GENERATION
DI		--NA--	DESTINATION UI
DD		--NA--	DESTINATION DEVICE NUMBER
MF		--NA--	MASTER FILE NAME
UN	=	0	USER NUMBER

NOTE - N/A DENOTES INVALID PARAMETER

FILE NAME	USER INDEX

④

Figure IV-2-11. Permanent File Utilities Left K Display (after UT is entered) (The circled numbers are identified in the text.)

util	Description
DUMP	Dump permanent files
LOAD	Load permanent files
CAT	Catalog permanent files
ATC	Catalog archive tape
COPY	Copy archive file to control point

The pi are the parameters desired for this option (section 2.3).

When this control statement is read, the K display appears. However, since parameters were entered on the control statement, they cannot be entered via the K display.

Exception: If there is an error in parameters detected at initiation of a utility, control is returned to PFS and parameters may then be entered via the K display.

2.2.3 TERMINAL INPUT

The format of terminal input for calling the permanent files utilities is substantially the same as that for batch input. At login, the analyst enters the BATCH subsystem, calls the desired utility, and enters the appropriate parameters with the command

PFutil(p1, p2, . . . , pn)

where util and the pi are the same as for batch input.

NOTE

The system must be in debug mode for terminal input.

2.3 PARAMETERS FOR THE UTILITY CONTROL STATEMENTS

The parameters available for all the permanent file utility control statements are described in this section. The entry for each parameter includes the designation and the description of each parameter. Archive file preassignment is also discussed.

2.3.1 DESCRIPTIONS OF UTILITY PARAMETERS

FM = family name

1- to 7-character name of the family of permanent file devices to be dumped, loaded, or cataloged. This is not required if only one family of devices is active in the system. Default is normal system family name.

PN = packname

1- to 7-character name of the auxiliary device to be dumped, loaded, or cataloged. The device must be mounted and available. Default is no name.

DN = device number

1- or 2-octal digit number which identifies one specific device within the system (or family) that is to be dumped, loaded, or cataloged. This is used only when a part of the permanent file system is to be processed. If this parameter is omitted, all permanent file devices in the system (or family) may have their files processed. Default is zero. This entry will be assumed octal unless a nonoctal digit is encountered or a D radix is used.

If UI is specified, DN does not have to be specified as the utility will locate the proper device.

OP = utility options

This parameter, which can be from 1 to 7 characters, specifies the options which control the processing of files by the utility called. Character strings are permissible under defined conditions. Many of the options require additional parameters to complete the definition of the selection. Default is zero.

Only one of the following three options may be used at a time. Each requires TM and DT parameters to establish a dividing time after which all files that meet the criteria of the option are singled out for processing. If the B parameter is added, the time before is specified.

C	Make selection according to time of creation.
A	Make selection according to time of last access.
M	Make selection according to time of last modification.

The following option is used only with C, A, or M.

B Denotes that the time specified on the TM and DT parameters is a dividing time before which all files meeting the criteria of the C, A, or M option are processed

Only one of the following two options may be used at any time. These may be used in conjunction with the previous temporal options.

I Select indirect access files only.

D Select direct access files only.

The following option is used only with a dump.

P Purge after dump. All files included in the dump are purged after the dump is completed. This option may only be used when the dump is initiated from the console.

The following four options are used only with a load.

R Select replace option. Files being loaded from an archive tape replace those files in the permanent file system for which there is a one-to-one matching of file names. Normally, if a file already exists in the permanet file system, PFLOAD skips loading of the same-name file from the archive tape (refer to N option). This option is not allowed when the load is initiated from a batch or terminal origin job.

N Noninitial load. On a noninitial load, PFLOAD reads the catalogs of files on the permanent file system and matches them with the archive catalog. If a file already exists in the permanent file system, the archive counterpart is skipped. If the file does not exist in the permanent file system, the file is added to the catalog and the data stored.

  This process is the reverse of the R option which purges a file in the permanent file system and replaces it with an archive file whenever a name match is found.

Ordinarily the noninitial load would be the default if R were omitted. However, the N parameter is required to satisfy the check made by PFLOAD to be sure the analyst wants to load a master device that already contains permanent files.

E Extract catalog image record (CIR) only. The CIR is described in section 2.4.4. PFLOAD reads the CIR from the designated tape and generates a random file and directory. However, none of the files after the CIR on the archive file are processed. Instead, the utility requests the next archive file.

O Omit CIR read. PFLOAD does not read the CIR for the specified archive file but processes the records in this archive file. After processing, PFLOAD terminates normally and does not request another archive file.

The following parameter is valid only for PFCOPY.

Q Select leading records. When specified, the archive file that is copied to a control point includes two header records, one with the catalog entry for the file and the second with the permit information for the file.

T = archive file name — 1- to 7-character name of the file that a utility is using to store or read archive files. Although usually a physical tape, it can be a mass storage device. Default name is TAPE.

NT — Specifies a 9-track archive tape. Default value is 7-track(MT).

DE = density — Specifies the density of the recording on the archive tape. Available values for density are the following.

0	Default
1	556 bpi
2	200 bpi
3	800 bpi
4	1600 bpi

NR

The NR parameter cancels out all rewinds for the operation in which it is specified.

The following rewinds are default.

	Rewind before Processing	Rewind after Processing
PFDUMP	X	
PFLOAD	X	X
PFATC	X	X
PFCOPY	X	X

Default is rewind.

NOTE

Positioning of a file should take into account that the first step in processing the file is to read (or write) the file label.

NU

No unload option. When PFDUMP has finished creating an archive tape, it automatically unloads that tape; that is, the whole physical tape is returned to the takeup reel and vacuum released. At this point, the reel is either removed or physically reloaded for further use. The NU parameter inhibits this action.

The NR and NU parameters can be used to generate several dumps on one archive tape. As an example, the following control statement outlines could be used to generate three dumps on one archive tape.

PFDUMP(...NU)	PFDUMP automatically rewinds before but not after processing. It also automatically unloads after processing. The NU cancels this unload and leaves the tape positioned at the end of this first dump.
PFDUMP(...NR, NU)	The NR keeps PFDUMP from rewinding before the dump. This second dump is written after the first. NU inhibits unload and leaves the tape positioned after this second dump.

	PFDUMP(...NR)	NR keeps PFDUMP from rewinding before the dump. This third dump is added to the other two. Since NU is not specified, the tape is automatically unloaded.

SF = number of files to skip

1- or 2-digit number which specifies the number of archive files to skip before processing begins. Default is zero (no skip). This parameter is assumed decimal unless B radix is used (refer to example under N).

N = number of files to process

1- or 2-digit number which specifies the number of archive files on an archive tape to process. If set to zero, one file will be processed. Default is zero. This parameter is assumed decimal unless B radix is used.

Example of the use of SF and N:

Given an archive tape AT with six archive files, F1, F2, F3, F4, F5, and F6. It is desired to load F4 and F5. The control statement

```
PFLOAD(T=AT,SF=3,N=2)
```

causes the utility to skip F1, F2, and F3, and begin processing with F4 (SF=3). The utility processes F4 and F5 (N=2).

DT = date

Specifies the date to be used with C, A, M, or B option. Format is yymmdd. Value is in decimal. Default is zero.

TM = time

Specifies the time to be used with C, A, M, or B option. Format is hhmmss. Value is in decimal. Default is zero.

UI = user index

This parameter limits processing to files located under this user index. It can be further restricted by using it with other options. This 1- to 6-digit parameter is assumed octal unless D radix or nonoctal digit is used. Default is zero. If DN=0, utilities will locate proper device to process. This parameter is usually entered to make use of the R utility option.

If UI is specified, DN does not have to be specified as the utility will locate the proper device.

DI = destination user index

When specified, all files being processed by PFLOAD are loaded to this user index. The 6-digit parameter is assumed octal unless D radix or nonoctal digit is used. Default is zero.

PF = permanent file name	1- to 7-character name which specifies the permanent file name for which processing is desired. This parameter is associated with the UI option and is ignored if UI is not specified. Default is no name.
VF = verify file name	1- to 7-character name which indicates the name of a file on which PFDUMP stores a duplicate of the archive file it creates. This file may be on mass storage or on tape. Default name is PFVER.
V	Indicates to PFDUMP that it is to produce a verification file that is a duplicate of the archive file it creates. After PFDUMP has completed processing, this file is verified with the primary archive file. Default is no verify file written.

If NU parameter is specified, neither the archive file nor the verify file are unloaded. Thus, multifile verify files can be generated along with multifile archive files.

Example:

The following sequence of commands creates a two-file archive tape (AA) and a matching verify tape (BB). The VERIFY command compares the accuracy of the duplication. (Refer to volume 1 of the system reference manual for a complete description of VERIFY.)

PFDUMP(T=AA, VF=BB,V,NU)	The dump is written on archive file AA. A duplicate is written on the verify file BB. PFDUMP does not rewind after processing and an NR is not needed.
PFDUMP(T=AA, VF=BB,V,NU,NR)	This writes a second dump after the first on both the archive (AA) and the verify (BB). Each dump produces a separate file on the archive and verify files. The NR is necessary to inhibit the rewind before the dump.

VERIFY(AA, BB, N=O, A, R)	Performs a binary compare of AA and BB. If words do not match, this command lists: Record number Word number within the record Words from both files that do not match N=O specifies that the verify terminates on the first empty file. A specifies an abort if a mismatch is found. R rewinds both files before and after the verify.

L = name of output file

1- to 7-character name of the file on which reports are to be written. The default name is OUTPUT.

LO = list option

The string of characters listed in this option specify the type of output records desired from the permanent file utilities. Default is LO=E (list errors).

- T List all files processed
- C List all files in catalog for system
- E List errors
- S List cumulative statistics for catalog

These options are specified as a character string.

Example: LO = TES

DD = destination device number

This parameter is used only by PFLOAD and specifies the device number where files are to be loaded when their original device is no longer defined in the system. Default is zero. The 1- or 2-digit parameter is assumed octal unless D radix is affixed or a nonoctal digit appears.

Example:

Files have been dumped from device 3 to archive tape AB. It is desired to reload these files. The control statement

PFLOAD(T=AB, DD=5, DN=4)

is executed. The utility tries to load the files on archive tape AB on device 3, the source device. If device 3 is no longer defined in the system, it will load them on device number 5 which is the specified default device. All files on dump tape AB which resided on the master device are loaded to device number 4.

MF = master file name

This parameter is used only with PFCOPY and specifies to the utility that all the files extracted from the designated archive tape are to be copied to a control point under one file name, the master file name (1- to 7-characters). Default is no name. Normally, the archive files are copied as individual files retaining their permanent file names. With the MF capability, it is possible to extract a file of one name from an archive file and change the name as a local file.

UN = user number

This parameter specifies the 1- to 7-character user number which is associated with the packname parameter. This user number must match information in the mass storage table (MST) for the specified packname. Default is no name. If UN is specified and PN is not, the utility will convert UN to a user index. If, in addition, DN is not specified, the utility will locate the proper device in the family.

2.3.2 PREASSIGNING THE ARCHIVE FILE

In all permanent file utilities except PFCAT, consideration should be given to the assignment of the archive file. The type and density of the file (tape) can be specified with the utility parameters NT and DE (refer to section 2.3.1). These attributes can also be specified when the file is assigned to a control point. The relation between file preassignment and the specification of the DE and NT parameters is shown in the following cases.

1. If the file specified by the T option (or the default file TAPE) has been assigned to a control point prior to the utility call, that file is used as the archive file. In this case, the NT and DE tape attributes are ignored if specified.

   Example:

   Archive file assignment and the utility call are made with the following statements.

   ```
   LABEL(TAPE)              ASSIGN(MS,TAPE)
   PFDUMP(NT,DE=4)     or   PFDUMP(NT,DE=4)
   ```

   In this example, the file TAPE is already at a control point when the utility is called. The utility does not make a tape request, and NT and DE are ignored.

2. If a vsn for the archive file has been specified prior to the utility call, the assignment of the archive file is made to a tape having the specified vsn and the attributes specified by the NT and DE parameters.

   Example:

   Tape assignment and the utility call are made with the following statements.

   ```
   VSN(TAPE=A)
   PFDUMP(NT,DE=4)
   ```

   In this example, the vsn has been specified prior to the utility call. The system attempts to assign a tape with vsn A and type and density as specified by NT and DE (9-track, 1600 bpi).

3. If no archive file preassignment was done, the operator is requested to assign the unit to be used with attributes that are specified by the NT and DE parameters.

Example:

The following statement is issued.

```
PFDUMP(NT,DE=4)
```

The operator is requested to assign a unit with attributes as specified by the NT and DE parameters. If NT is not specified, any valid equipment, including mass storage, may be assigned.

2.4 PERMANENT FILE UTILITY ROUTINES

This section and its subsections describe the five permanent file utility routines. The parameters available for each of the routines are listed with the routine.

An option allows each permanent file utility to produce a cataloged directory of file information. This catalog information is either derived from the catalog tracks on the permanent file device or from the archive tape. Although the header information may differ from utility to utility, the format of the catalog information remains the same. The following is the general format of the information listed for each file on the directory.

```
filenam   access   type                  length   dn   cdate   ladate   lmdate
passwor      mc      userin   permission   subsystem   time    time     time
```

(1) filenam	Permanent file name
(2) access	Type of permanent file. This field may be either DIR (direct access) or IND (indirect access).
(3) type	Method of access or category. This field may be either PRIVATE, SEMI-PR, or PUBLIC. These methods of access are described in volume 1 of the system reference manual.
(4) length	Length of the file in PRUs
(5) dn	Device number of the mass storage device on which the file is stored. If the file resides on the master device, this field is replaced by an *.
(6) passwor	Password associated with the file
(7) mc	Modification count. This number specifies the number of times the file has been accessed in WRITE, MODIFY, or APPEND mode.

(8) userin	User index of the user on whose catalog this file resides
(9) permission	Permission mode. This may be WRITE, MODIFY, APPEND, READ, READMD, READAP, or EXECUTE. These modes are explained in detail in volume 1 of the system reference manual.
(10) subsystem	Specifies the subsystem under which the file was saved. Possible entries include FTNTS, BASIC, EXEC., or BATCH. If this field contains no entry, a subsystem is not associated with the file.
(11) cdate time	Specifies the time and date of file creation. The format is: yy/mm/dd. hh. mm. ss.
(12) ladate time	Specifies the time and date of the last access to the file
(13) lmdate time	Specifies the time and date of the last modification to the file

Figure IV-2-12 illustrates a typical page from a cataloged directory. Although this directory was produced using the command PFATC (LO=T), directories in similar format could be produced using the other utilities. Each of the fields previously described is shown on the figure.

```
P. F. ARCHIVE CATALOG        PFDUMP                           PAGE 101
                             EQ. MASK    377          76/01/14.    22.44.57.

FILE NAME ACCESS FILE-TYPE LENGTH ON CREATION LAST ACCESS LAST MOD
 PASSWORD MD/CNT INDEX PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME

1701 D1P   ①      IND.  PRIVATE            1      75/12/28.   75/12/28.   75/12/28.
                   0       106   WRITE             06.46.58.   11.17.57.   11.17.57.
1702 D1       ②   IND.  PRIVATE           16      75/12/27.   75/12/28.   75/12/28.
                   1       106   WRITE             07.54.08.   11.17.57.   11.17.57.
1703 PSILIB       IND.  PRIVATE ③        43  ④   75/12/27.   75/12/28.   75/12/28.
                   7       106   WRITE             08.47.09.   10.49.04.   10.48.56.
1704 PSIPL        DIR.  PRIVATE           81 ⑤ *75/12/37.   75/12/28.   75/12/28.
                   6       106   WRITE             09.39.51.   10.46.38.   10.38.46.
1705 REK          IND.  PUBLIC             1      74/03/20.   76/01/13.   76/01/13.
                   3       113    READ             15.01.56.   11.23.55.   11.11.19.
1706 D5CREAT      IND.  PRIVATE            1      75/12/19.   75/12/23.   75/12/21.
                  14      1111   WRITE             00.13.35.   13.56.54.   19.56.16.
1707 PFTB         IND.  PRIVATE            1      76/01/14.   76/01/14.   76/01/14.
     PASSWOR ⑥     2       113    EXEC     EXEC   09.45.25.   09.57.36.   09.57.36.
1708 D            IND.  PRIVATE            1      76/01/03.   76/01/03.   76/01/03.
                   0       115   WRITE             17.45.36.   17.45.36.   17.45.36.
1709 MORNING      IND.  SEMI-PR            1      76/01/14.   76/01/14.   76/01/14.
     AM            1 ⑦     113 READMD     FTNTS   09.59.30.   10.03.07.   09.59.30.
1710 PRIME        IND.  PRIVATE            1      73/10/30.   76/01/13.   73/10/30.
                   4 ⑧    113   WRITE ⑨ BASIC    13.42.30.   11.18.16.   13.42.30.
1711 AFTNOON      IND.  PUBLIC             1      76/01/14.   76/01/14.   76/01/14.
     PM            1       113 READAP     BASIC ⑩ 10.00.13.   10.03.31.   10.00.13.
1712 STRIP        IND.  PRIVATE            1      73/12/18.⑪ 76/01/14.⑫ 76/01/14.⑬
                   4       113 APPEND              01.53.29.   09.48.11.   09.48.11.
1713 REKSRC       IND.  PRIVATE          128      74/02/06.   76/01/13.   76/01/13.
                   7       113    NULL             13.30.16.   11.12.55.   11.12.55.
```

Figure IV-2-12. Sample Directory Produced by PFATC (LO=T)

2.4.1 CATALOG PERMANENT FILE ARCHIVE TAPE (PFATC)

PFATC is a permanent file utility program used to produce a cataloged directory of file information derived from an archive tape. An archive tape is used as backup storage and is created by the utility PFDUMP (refer to section 2.4.3). The format of the directory depends upon the parameters selected. The following parameters are available with PFATC.

PFATC(p_1, p_2, ..., p_n)

p_i parameters are as follows:

p_i	Description	
T =	Archive file name	
LO =	T	Files processed
	C	Catalog files
	E	Errors
L =	Output file name	
OP =	C	Creation
	A	Last access
	M	Last modification
	I	Indirect access files
	D	Direct access files
	B	Before date and time
NT	Nine track	
DE =	Density	
NR	No rewind	
SF =	Number of files to skip	
N =	Number of files to process	
DT =	Date	
TM =	Time	
UI =	User index	
PF =	Permanent file name	
UN =	User number	

Complete descriptions of these parameters are available in section 2.3.1.

The directory produced by this utility is similar to that shown in Figure IV-2-12 if LO=T is specified. If LO=C is specified and the archive tape to be cataloged was produced by a selective dump, a directory similar to Figure IV-2-17 is produced. Selective dumps and Figure IV-2-17 are described in section 2.4.4.

2.4.2 CATALOG PERMANENT FILE DEVICE (PFCAT)

PFCAT is a permanent file utility program used to produce a cataloged directory of file information derived from catalog tracks on a permanent file device. The following parameters are available with PFCAT.

PFCAT(p_1, p_2, ..., p_n)

p_i parameters are as follows:

p_i	Description	
FM =	Family name	
PN =	Pack name	
DN =	Device number	
LO =	T	Files processed
	E	Errors
	S	Summary
L =	Output file name	
OP =	C	Creation
	A	Last access
	M	Last modification
	I	Indirect access files
	D	Direct access files
	B	Before time and date
DT =	Date	
TM =	Time	
UI =	User index	
PF =	Permanent file name	
UN =	User number	

Complete descriptions of these parameters are available in section 2.3.1.

The directory produced by this utility when LO=T is specified is similar to Figure IV-2-12. However, the files in the directory produced by PFCAT are listed according to user index and totals are given after the files for each user index. Figure IV-2-13 illustrates a page from a PFCAT(LO=T).

```
        CATALOG FILE                                              76/01/14. 21.51.01.
DIRECTORY OF PERMANENT FILE DEVICE     40 SYS172                      PAGE      127
CATALOG OF USER INDEX       103
    FILE NAME ACCESS FILE-TYPE LENGTH ON CREATION LAST ACCESS LAST MOD
     PASSWORD MD/CNT INDEX PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME

    1 SSEHC      IND. PRIVATE          1      75/01/17.   75/12/03.   75/10/13.
                  140         WRITE   BATCH   13.00.51.   14.08.47.   13.36.13.
    2 S          IND. PRIVATE          2      75/03/18.   75/10/29.   75/08/22.
                   19         WRITE           15.01.37.   15.06.41.   13.41.22.
    3 BCFILE     IND. PRIVATE         18      75/01/31.   75/04/29.   75/01/31.
                    8         WRITE           10.28.13.   15.46.35.   11.26.31.
    4 BEER       IND. PRIVATE          1      75/06/09.   75/11/07.   75/11/07.
                   25         WRITE           13.15.28.   10.33.13.   10.33.13.
    5 TST        IND. PRIVATE          1      75/06/11.   75/06/11.   75/06/11.
                    0         WRITE           14.06.19.   14.06.19.   14.06.19.
    6 CH2        DIR. PRIVATE        248    * 75/05/02.   75/12/03.   75/05/02.
                   48         WRITE           11.58.07.   14.08.56.   11.58.07.
    7 MASTER     DIR. PRIVATE          2    * 75/05/21.   75/12/23.   75/12/23.
                   39         WRITE           12.15.01.   10.18.51    10.18.51.
    8 LIBRARY    IND. PRIVATE         20      75/05/02.   75/12/03.   75/12/03.
                   48         WRITE   BATCH   12.43.57.   14.17.01.   14.17.01.

    TOTALS          8  FILE(S),          293  SECTORS
```

Figure IV-2-13. Directory Produced by PFCAT(LO=T)

In addition to the directory shown in Figure IV-2-13, PFCAT(LO=T) also produces a mass storage table report. This report gives information about each mass storage device in the system. Figure IV-2-14 shows an example of a mass storage table report.

Messages issued with this report give information concerning the type and status of the device. The following status messages can be issued.

ACCOUNT INITIALIZE PENDING.
CATALOG TRACK OVERFLOW.
DAYFILE INITIALIZE PENDING.
ERRLOG INITIALIZE PENDING.
FORMAT PENDING.
I/O QUEUE INITIALIZE PENDING.
PF INITIALIZE PENDING.
TOTAL INITIALIZE PENDING.
UNAVAILABLE FOR PF ACCESS.

The following types of devices are listed.

ALTERNATE SYSTEM.
REMOVABLE DEVICE.
SYSTEM.
TEMPORARY.

MASS STORAGE TABLE REPORT 76/01/14. 21.51.01.

FAMILY/PACK NAME SYSTAA

DEVICE NUMBER	=	1	DEVICE TYPE	=	DI
DEVICE MASK	=	377	NUMBER UNITS	=	1
SECONDARY MASK	=	377	USER NUMBER	=----------	
D/A USER COUNT	=	0	EQUIPMENT STATUS	=	ACTIVE

SYSTEM
TEMPORARY.

Figure IV-2-14. Mass Storage Table Report

The LO=S option generates cumulative statistics for the device cataloged. This summary report lists file information for each user index and for the entire device. Figure IV-2-15 lists a page of the information concerning individual users. Figure IV-2-16 gives an example of the general device information. Figures IV-2-15 and IV-2-16 were obtained using PFCAT(LO=S, DN=40).

SUMMARY REPORT 76/01/14. 21.51.01.
DIRECTORY OF PERMANENT FILE DEVICE 40 SYS172 PAGE 204

(* = DAF RESIDENT ON OTHER THAN MASTER DEVICE.)

USER	INDEX	FILES	SECTORS	AVE. SEC.
	3000	1	315	315
	10	26	57	2
	210	3	665	221
	20	18	33	1
	30	12	117	9
	230	4	394	98
	2230	42	1543	36
	40	18	58	3
	60	4	15	3
	100	23	552	24
	150	3	50	16
	160	6	247	41
		160	4046	(GROUP TOTAL)
	1	29	202	6
	201	5	21	4
	11	26	649	24

•
•
•

Figure IV-2-15. Cumulative Statistics by User Index PFCAT(LO=S,DN=40)

SUMMARY REPORT 76/01/14. 21.51.01.
DIRECTORY OF PERMANENT FILE DEVICE 40 SYS172 PAGE 207

INDIRECT ACCESS FILE HOLES

CATALOG TRACK	NUMBER HOLES	NUMBER SECTORS
0	114	631
1	12	66
2	177	1520
3	27	154
4	46	1368
5	86	891
6	52	238
7	13	277
10	7	120
11	83	1107
12	50	229
13	48	324
14	2	13
17	20	139
TOTAL	737	7077

SUMMARY REPORT 76/01/14. 21.51.01.
DIRECTORY OF PERMANENT FILE DEVICE 40 SYS172 PAGE 208

PERMANENT FILE STATISTICS SUMMARY

MASTER DEVICE USAGE

GROUP TOTALS

GROUP	TOTAL FILES	TOTAL SECTORS	PERCENT OF TOTAL USAGE
0	160	4046	6
1	497	6193	10
2	475	7658	12
3	149	3554	5
4	412	6320	10
5	213	5427	8
6	376	21514	35
7	216	5693	9

	TOTAL	IAF	DAF
TOTAL SECTORS	60405	24618	35787
TOTAL FILES	2498	2174	324
TOTAL USERS	94		
AVE. FILES/USER	26	23	3
AVE. SEC/FILE	24	11	110
DEVICE TYPE - MD-2			
PERCENT DEVICE USAGE	59	24	35

DIRECT ACCESS DEVICE USAGE

DEVICE NUMBER	DEVICE TYPE	TOTAL FILES	TOTAL SECTORS	PERCENT USAGE
40	MD-2	324	35787	35

Figure IV-2-16. Cumulative Statistics for Entire Device PFCAT(LO=S,DN=40)

2.4.3 COPY ARCHIVE FILE UTILITY (PFCOPY)

PFCOPY is a permanent file utility that extracts files from an archive file and copies them to one or more files at a control point. The following parameters are available with PFCOPY.

PFCOPY($p_1, p_2, \ldots, p_n$)

p_i parameters are as follows:

p_i	Description	
T =	Archive file name	
LO =	T	Files processed
	E	Errors
L =	Output file name	
OP =	C	Creation
	A	Last access
	M	Last modification
	I	Indirect access files
	D	Direct access files
	B	Before time and date
	C	Catalog and permit records
NT	Nine Track	
DE =	Density	
NR =	No rewind	
SF =	Number of files to skip	
N =	Number of files to process	
DT =	Date	
TM =	Time	
UI =	User index	
PF =	Permanent file name	
MF =	Master file name	
UN =	User number	

Refer to section 2.3.1 for a complete description of these parameters.

2.4.4 PERMANENT FILE DUMP UTILITY (PFDUMP)

PFDUMP is a utility that copies permanent files to backup storage (an archive file). This backup can be reloaded by the PFLOAD utility (section 2.4.5) or its permanent file copies can be selectively accessed by other utilities (PFATC and PFCOPY) for cataloging and copying. The following parameters are available with PFDUMP.

PFDUMP($p_1, p_2, \ldots, p_n$)

p_i parameters are as follows:

p_i	Description
FM =	Family name
PN =	Pack name
DN =	Device number
T =	Archive file name
LO =	T Files processed
	C Catalog files
	E Errors
L =	Output file name
OP =	C Creation
	A Last access
	M Last modification
	I Indirect access files
	D Direct access files
	B Before time and date
	P Purge after dump
NT	Nine track
DE =	Density
NR	No rewind
NU	No unload
SF =	Number of files to skip
DT =	Date
TM =	Time
UI =	User index
PF =	Permanent file name
VF =	Verify file name
V	Verify
UN =	User number

Refer to section 2.3.1 for a complete description of these parameters.

Three types of dumps are available. These are determined by the options selected. The choice of options depends on the purpose of the dump. The following are descriptions of these three types and what they accomplish.

SELECTIVE PERMANENT FILE SYSTEM DUMP

If the purpose of the dump is to select and copy those permanent files with a specified update relevance (modified after some recent date), a selective dump will be used; that is, the option OP=M is specified along with a time origin (DT=date and TM=time). This produces an archive file with copies of all permanent files modified after this specified time. This selection can be further restricted by use of additional parameters available to the PFDUMP utility. For example, if the select indirect access files only option is included (OP=MI), only the indirect access files modified after the specified time origin (DT, TM) are copied by this dump. Normal procedure would follow up this dump with successive selective dumps (usually with periodic advancement of the time origin). This produces a series of archive files containing successive time-levels of updating for all the files in the defined category. These archive files can then be incrementally loaded to return the most recently modified versions to the permanent file system.

The incremental load proceeds back through the archive files in the reverse order in which they are created (that is, it starts with the most recent) and loads each file the first time it appears. Thereafter, a reappearance of that file is skipped. This produces a load of all the latest modifications within the specified category.

Each selective dump puts a record (or records) at the beginning of the archive file it creates that contains catalog images of all files active in the permanent file system when dump took place. This makes it possible to incrementally load these files on a system basis or a device basis.

The catalog image record is explained later in this section; the incremental load is treated in section 2.4.5. An example of selective dumping is also given in section 2.4.5.

PARTIAL PERMANENT FILE SYSTEM DUMP

If no specification of time and date of last modification is made (OP≠M), but some other option is selected, then the dump is classed as partial. Thus, a dump that copied all files created after a certain date (OP=C) or that copied files accessed after a specified date (OP=A) would be a partial dump. Likewise, the specification of direct access files only (OP=D) or indirect access files only (OP=I) would constitute a partial dump.

A partial dump may specify device or it may copy all files in the system that meet the criteria of the option.

FULL PERMANENT FILE SYSTEM DUMP

If no options are selected (OP=0), and either no device is specified (DN=0) or a master device is specified (DN=master device), then a full dump is initiated by the utility. A full dump copies all files in the system or on the specified master device.

Table IV-2-1 shows the defining characteristics of the three kinds of dump.

TABLE IV-2-1. PFDUMP TYPES

Selective Dump	Partial Dump	Full Dump
OP=M	OP=C A I D P O, DN=0 master device	OP=0, DN=0 master device

CATALOG IMAGE RECORD (CIR)

Each selective dump (OP=M) writes a catalog image record (CIR) at the beginning of the archive file on which the permanent files are dumped. There is an entry in the CIR for every permanent file in the system at the time of the dump (not just the files included in the dump).

This record of catalog images is used by a PFLOAD to reinstate files in the permanent file system with the parametric status and description they had when dumped.

Each entry in the CIR is two words. The following is the format of the entry.

<table>
<tr><td>59</td><td>41</td><td>35</td><td>17   0</td></tr>
<tr><td colspan="3">Permanent File Name</td><td>User Index</td></tr>
<tr><td>Access Count</td><td>Dev No</td><td colspan="2">Access Date and Time</td></tr>
</table>

The first word of the entry describes the file uniquely in the system. In the second word, the access count is the number of times this file was accessed as of the last access date and time. The device number is used only to identify direct access files.

When a file is loaded, this CIR information is placed in the permanent file catalog of the device being loaded. No other information can be changed over dumps and loads without updating modification time and date.

An example involving selective dumping is given in section 2.4.2.

Figure IV-2-17 contains a partial listing of the CIR. This listing can be obtained after a selective dump by using the LO=C option on the PFDUMP, PFLOAD, and PFATC statements. The listing obtained when LO=T is specified is similar to Figure IV-2-12. Although the header is different, the format of the information is the same.

PFATC CATALOG OF CATALOG IMAGES   PAGE 19
ON 76/01/14. AT 22.07.54.

	FILE NAME	USER INDEX	ACCESS COUNT	LAST ACCESS DATE	LAST ACCESS TIME	DEVICE NUMBER
451	Z	23	2	74/03/23.	11.15.23.	
452	3DT	26	10	74/06/04.	09.32.26.	
453	ADS	26	14	74/06/04.	09.30.04.	
454	DEMO	20	14	75/07/11.	16.39.20.	
455	FTN2	20	0	75/07/15.	11.13.20.	
456	TAPE2	20	2	75/07/11.	16.38.07.	
457	HRACER	25	14	75/08/22.	13.19.09.	
458	AA	20	40	75/07.22.	12.26.18.	
459	DSPX	27	12	75/12/18.	19.40.17.	
460	B1MSAJ	21	0	75/12/18.	12.51.34.	
461	BLDDIR	27	32	75/01/07.	16.10.29.	

Figure IV-2-17. Catalog of Catalog Image Record

2.4.5 PERMANENT FILE LOAD UTILITY (PFLOAD)

Archive files produced by the PFDUMP utility (section 2.4.4) can be loaded back onto the permanent file system with the PFLOAD utility. The load can reestablish the permanent file system exactly as it was at the time of the dump, or it can introduce additional restrictions and load only a desired subset of the files on an archive tape. The following parameters are available with PFLOAD.

PFLOAD(p_1, p_2, . . . , p_n)

p_i	Description	
FM =	Family name	
PN =	Pack name	
DN =	Device number. This parameter must be specified for PFLOAD.	
T =	Archive file name	
LO =	T	Files processed
	C	Catalog files
	E	Errors
L =	Output file name	
OP =	C	Creation
	A	Last access
	M	Last modification
	I	Indirect access files
	D	Direct access files
	B	Before date and time
	R	Replace
	N	Noninitial
	E	Extract CIR
	O	Omit CIR
NT	Nine track	
DE =	Density	
NR	No rewind	
SF =	Number of files to skip	
N =	Number of files to process	
DT =	Date	
TM =	Time	
UI =	User index	
PF =	Permanent file name	
DI =	Destination user index	
DD =	Destination device number	
UN =	User number	

Refer to section 2.3.1 for a complete description of these parameters.

Execution of the PFLOAD utility involves three checks.

1. A check of the archive tape to determine which files are eligible for load under the option specified

2. A check of the additional parameters in the load command (UI, PF, UN, etc.) to filter out the eligible files that meet these criteria

3. A check of the device or devices to be loaded to identify the files they may already contain and thereby determine which files from the archive tape are actually loaded

The following paragraphs detail these three steps

CHECK OF THE ARCHIVE TAPE

There are two possible ways to select load files from an archive tape; either an incremental load is used as a measure of selection or it is not. If an incremental load is used, the archive file must be checked against a CIR created by a selective dump. (The CIR gives identifying information on every file in the system at the time the selective dump was made.) Use of a CIR to select archive files for loading is known as an incremental load since it builds up (increments) an accumulation of the most recently modified versions of the files extracted from the archive tapes used in the load. The usual procedure in such a case is to read the CIR from the most recent selective PFDUMP tape (this tape is the first reel of the incremental load process) and place it on a random file. Then the archive tapes are read in the reverse order in which they were dumped. Each file on an archive tape is compared with the CIR. If a match is found on the CIR, the file becomes a candidate for load (goes to parameter check, stage 2), and this entry is cleared out of the CIR. If no match is found, the file is skipped because it has either been purged or previously loaded.

If no incremental load is used, then there is no CIR matching and the whole archive tape is passed on to the next check (stage 2).

CHECK OF ADDITIONAL PARAMETERS

All files selected from an archive tape as candidates for load are further checked against the additional restrictions specified in the parameter list of the PFLOAD command. Candidate files that meet all criteria are passed on to the final device check; all others are skipped.

DEVICE CHECK

The final check made by the PFLOAD utility is of the device(s) being loaded. Either the device has been initialized and is clear of all files, or it has not been initialized and contains permanent files at the time of load. The utility always begins this final check with the assumption that the device should be initialized (unless the OP=N option has been included), and if it finds that it is not, issues the diagnostic:

DEVICE NOT INITIALIZED

If the intention is to load an initialized device, the device must be initialized and the load reinitiated. If the intention is that the permanent files already on the file should remain there and only those files should be loaded which do not duplicate the ones already there (or should replace those there if the R options is specified), the PFLOAD command must include the noninitial parameter (OP=N). This additional check makes certain that it really is intended to load a device that already contains files.

The utility determines the catalog track where the selected archive files are to reside and builds an index file with a one-word entry for each file on the track. The following is the format of the entry.

59	17 0
file name	user index

The utility checks each candidate file against the index to see if it is already in the permanent file system. If the R parameter has not been specified, duplicates are skipped. If the R parameter has been specified, the archive duplicate replaces the one already in the permanent file system.

The listing produced when LO=T is similar to Figure IV-2-12. When LO=C, the listing is similar to Figure IV-2-17.

Example:

In this example, permanent files created and modified on three devices are dumped nine times. The action runs from March 1, 1975 (75.3.1) to March 10, 1975 (75.3.10). For simplification, the time of day (TM) is not shown.

Before the last dump, one of the devices goes down. The example then shows how this device can be loaded from the archive tapes.

The following are the three devices.

DEVICE 40 — A master device containing indirect access files only. Users with this master device have their direct access files written on 41 or 42.

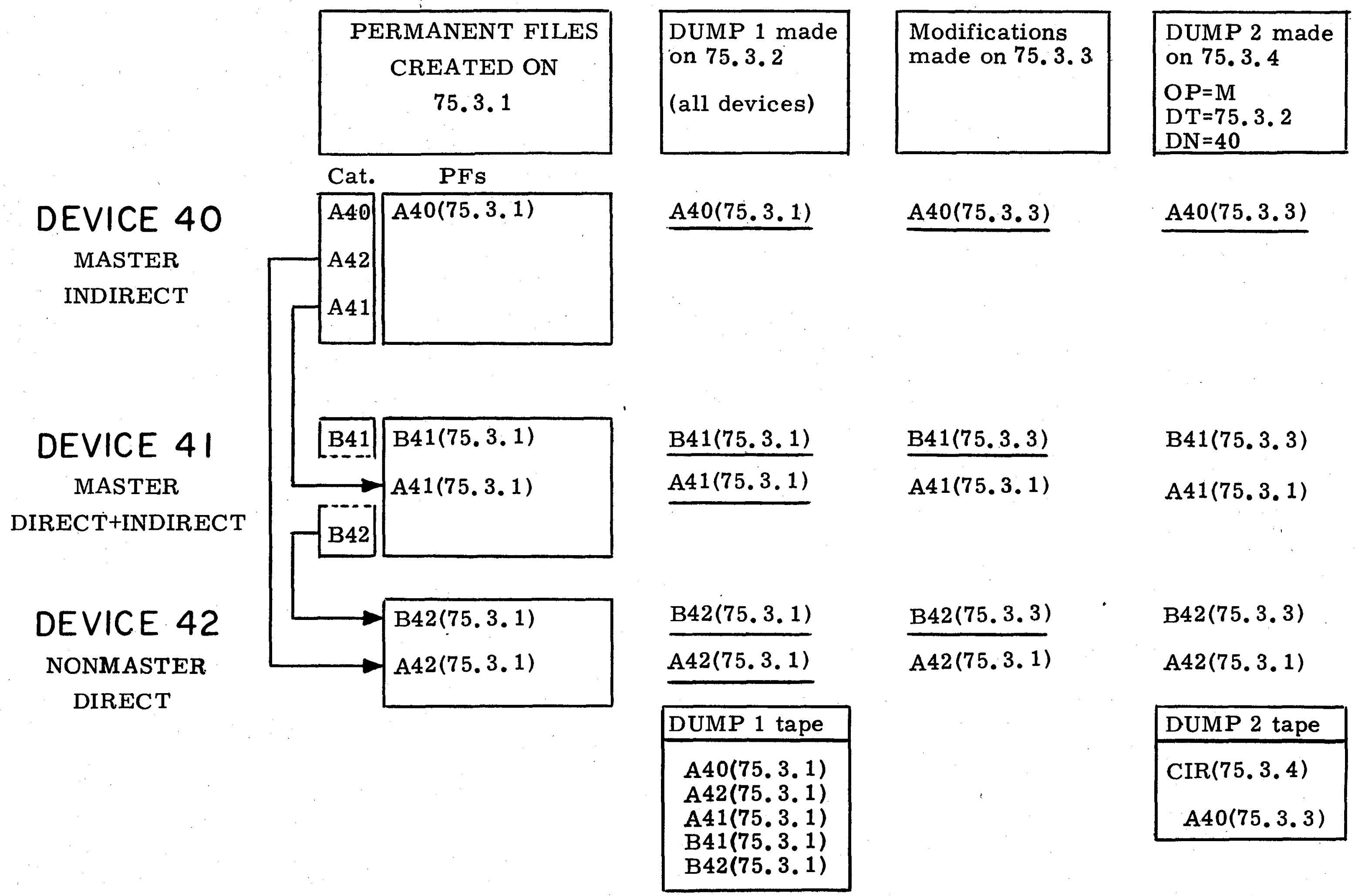
DEVICE 41 — A master device containing indirect and direct access files. Direct access files cataloged on this device may be written on 42 or this device.

DEVICE 42 — A nonmaster device containing direct access files only. All files written on this device will be cataloged on 40 or 41.

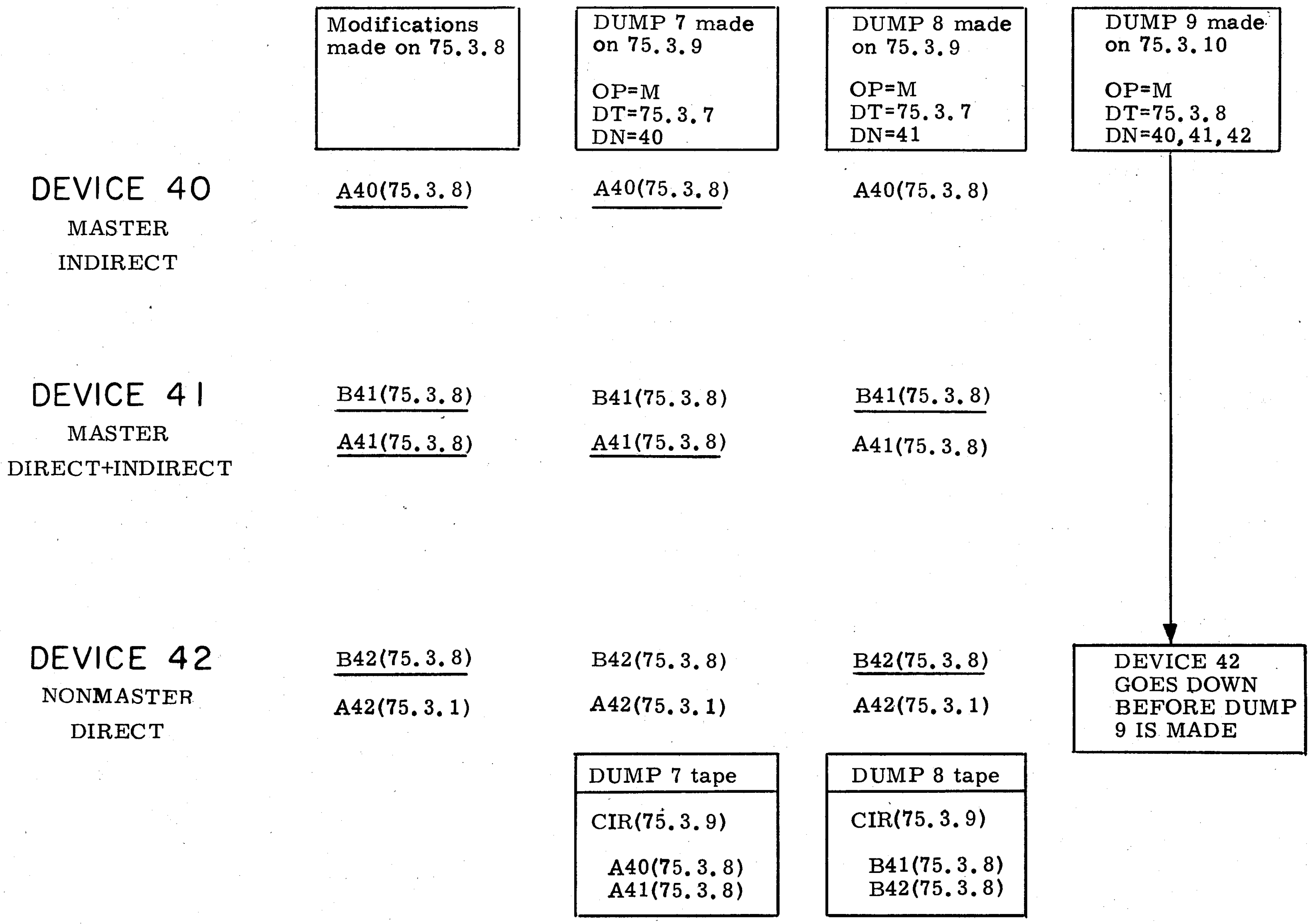
File names consist of a letter and a number. The letter identifies which device contains the catalog entry for this file (A=device 40 and B=device 41). The number is the same as the device on which the file is written.

Each file name is followed by a date in parentheses. This indicates either the day on which the file was created or when it was last modified.

The following illustration outlines the creation, modifications, and dumps that were made. There is one particular action per column. The top of the column identifies the action and the key parameters used by the utility. If an archive tape is produced, it is shown at the bottom of the column. All files in the system are listed in each column. Those files affected by the action are underscored.


PERMANENT FILES CREATED ON 75.3.1
DUMP 1 made on 75.3.2 (all devices)
Modifications made on 75.3.3
DUMP 2 made on 75.3.4
OP=M
DT=75.3.2
DN=40
Cat.
PFs
DEVICE 40
MASTER
INDIRECT
A40
A42
A41
A40(75.3.1)
A40(75.3.1)
A40(75.3.3)
A40(75.3.3)
DEVICE 41
MASTER
DIRECT+INDIRECT
B41
B42
B41(75.3.1)
A41(75.3.1)
B41(75.3.1)
A41(75.3.1)
B41(75.3.3)
A41(75.3.1)
B41(75.3.3)
A41(75.3.1)
DEVICE 42
NONMASTER
DIRECT
B42(75.3.1)
A42(75.3.1)
B42(75.3.1)
A42(75.3.1)
B42(75.3.3)
A42(75.3.1)
B42(75.3.3)
A42(75.3.1)
DUMP 1 tape
A40(75.3.1)
A42(75.3.1)
A41(75.3.1)
B41(75.3.1)
B42(75.3.1)
DUMP 2 tape
CIR(75.3.4)
A40(75.3.3)

	DUMP 3 made on 75.3.4 OP=M DT=75.3.2 DN=41	DUMP 4 made on 75.3.4 OP=M DT=75.3.2 DN=42	Modifications made on 75.3.5	DUMP 5 made on 75.3.6 OP=M DT=75.3.4 (all devices)	DUMP 6 made on 75.3.7 OP=M DT=75.3.4 DN=42
DEVICE 40 MASTER INDIRECT	A40(75.3.3)	A40(75.3.3)	A40(75.3.5)	A40(75.3.5)	A40(75.3.5)
DEVICE 41 MASTER DIRECT+INDIRECT	B41(75.3.3) A41(75.3.1)	B41(75.3.3) A41(75.3.1)	B41(75.3.5) A41(75.3.1)	B41(75.3.5) A41(75.3.1)	B41(75.3.5) A41(75.3.1)
DEVICE 42 NONMASTER DIRECT	B42(75.3.3) A42(75.3.1)	B42(75.3.3) A42(75.3.1)	B42(75.3.5) A42(75.3.1)	B42(75.3.5) A42(75.3.1)	B42(75.3.5) A42(75.3.1)
	DUMP 3 tape CIR(75.3.4) B41(75.3.3) B42(75.3.3)	DUMP 4 tape CIR(75.3.4) B42(75.3.3)		DUMP 5 tape CIR(75.3.6) A40(75.3.5) B41(75.3.5) B42(75.3.5)	DUMP 6 tape CIR(75.3.7) B42(75.3.5)


Modifications made on 75.3.8
DUMP 7 made on 75.3.9
OP=M
DT=75.3.7
DN=40
DUMP 8 made on 75.3.9
OP=M
DT=75.3.7
DN=41
DUMP 9 made on 75.3.10
OP=M
DT=75.3.8
DN=40,41,42
DEVICE 40
MASTER
INDIRECT
A40(75.3.8)
A40(75.3.8)
A40(75.3.8)
DEVICE 41
MASTER
DIRECT+INDIRECT
B41(75.3.8)
A41(75.3.8)
B41(75.3.8)
A41(75.3.8)
B41(75.3.8)
A41(75.3.8)
DEVICE 42
NONMASTER
DIRECT
B42(75.3.8)
A42(75.3.1)
B42(75.3.8)
A42(75.3.1)
B42(75.3.8)
A42(75.3.1)
DEVICE 42 GOES DOWN BEFORE DUMP 9 IS MADE
DUMP 7 tape
CIR(75.3.9)
A40(75.3.8)
A41(75.3.8)
DUMP 8 tape
CIR(75.3.9)
B41(75.3.8)
B42(75.3.8)

Actions during the 10 days in March in which the dumps and modifications were made run as follows:

Date	Action
75.3.1	Five permanent files are created on this date. File A40 (indirect) is written on device 40 and cataloged on device 40. File A42 (direct) is written on device 42 and cataloged on device 40. File A41 (direct) is written on device 41 and cataloged on device 40. File B41 (indirect) is written on device 41 and cataloged on device 41. File B42 (direct) is written on device 42 and cataloged on device 41.
75.3.2	DUMP 1. A full permanent file system dump. No CIR is written on the archive file.
75.3.3	Three files (A40, B41, and B42) are modified on this date. This is indicated by the modification date in parentheses following the file name.
75.3.4	DUMP 2. This is a selective dump of device 40 that specifies permanent files modified after 75.3.3. The files for this device are the ones listed on its catalog track. The utility scans this catalog which contains entries for files A40, A42, and A41. File A40 is on device 40 and the other two are on the devices indicated by their names. Only file A40, which was modified on 75.3.3, qualifies for this dump. It is written on an archive tape after a CIR for this date (75.3.4). The CIR gives a current description of all five files in the permanent file system.
75.3.4	DUMP 3. This is a selective dump of device 41 that specifies all files modified after 75.3.2. The catalog for this device contains entries for files B41 and B42. Both files were modified on 75.3.3 and qualify for this dump. They are written on an archive file after a CIR for this date.
75.3.4	DUMP 4. This is a selective dump of device 42 that specifies all files modified after 75.3.2. The PFDUMP utility scans the catalogs on devices 40 and 41 to locate the files on 42. One file, B42 (75.3.3), falls in this category.
75.3.5	Three files are modifed on this date.
75.3.6	DUMP 5. This is a selective dump of each of the three devices. All files modified after 75.3.4 are specified by the DT option. Since no DN is specified, all devices are dumped in turn, producing a dump tape with three archive files.
75.3.7	DUMP 6. This selctive dump of device 42 specifies all files modified after 75.3.4. Since 42 is nonmaster, the dump routine scans the catalogs of the other two devices.

Date	Action
75.3.8	Four files are modified on this date.
75.3.9	DUMP 7. A selective dump of device 40 is made with DT=75.3.7. Two files qualify.
75.3.9	DUMP 8. A selective dump of device 41 is made with DT=75.3.7. The catalog on device 41 contains entries for files B41 and B42. File B41 is an indirect access file located on device 41, and file B42 is a direct access file located on device 42. Both files were modified after 75.3.7 and are dumped.
75.3.10	DUMP 9. A selective dump of devices 40, 41, and 42 is specified, but device 42 goes down before the dump is made.

To reestablish permanent files on device 42 with the most recent modifications available, the device is first initialized and then incrementally loaded, beginning with the most recent selective dump tape. Archive dump 8 is the most recent selective dump tape and accordingly contains the most recent CIR. This reel is assigned first and then the CIR read onto a random file. The dump tapes are read in reverse order with each file on each tape being checked against the CIR. Dump tapes 2 and 3 are omitted since they do not involve device 42.

For each catalog track on device 42 that is to receive file entries, PFLOAD builds and maintains an index file. Before each file can be loaded, PFLOAD checks the index file to see if the file has already been loaded. If it has, the archive duplicate is skipped; if not, the archive file is loaded and another entry goes into the index file.

After initialization of device 42, the loading runs as follows:

Archive File	Action
DUMP8 tape CIR(75.3.9) B41(75.3.8) B42(75.3.8)	The analyst assigns DUMP8 tape and enters the command: PFLOAD(T=DUMP8, DN=42) The utility reads the CIR from this tape onto a random file. It then looks for files for device 42 on this tape. B42 qualifies, and its entry in the CIR is removed. The file B42 is loaded on device 42. The catalog of this file on device 41 is created. After loading this file, the utility makes an entry in the index file as follows: [B42 | user index] The utility then requests the next reel.
DUMP7 tape CIR(75.3.9) A40(75.3.8) A41(75.3.8)	The analyst assigns this tape, and the utility reads the archive file and checks it against the CIR loaded from DUMP8 above. No match is found. The next reel is requested.

Archive File	Action
DUMP6 tape CIR(75.3.7) B42(75.3.5)	This tape is assigned and then compared with the CIR. B42 is a file from device 42. However, its entry in the CIR was removed when it was read from DUMP8.
DUMP5 tape CIR(75.3.6) A40(75.3.5) B41(75.3.5) B42(75.3.5)	This tape is assigned and then compared with the CIR. B42 is from device 42 but it has already been removed from the CIR and the utility skips this version.
DUMP4 tape CIR(75.3.4) B42(75.3.3)	This tape is assigned and compared with the CIR originally read from DUMP8 tape. It has been removed from the CIR and this file is skipped.
DUMP1 tape A40(75.3.1) A42(75.3.1) A41(75.3.1) B41(75.3.1) B42(75.3.1)	This tape, produced by a full dump, contains all the files that were on the permanent file system when the dump was made (75.3.1). These files are checked against CIR (75.3.9). A match is found with A42. This file is loaded. Device 42 now has the same files with the same update status that they had on 75.3.9 when the last selective dump was made.

2.5 DIAGNOSTICS

2.5.1 PERMANENT FILE SUPERVISOR (PFS)

OPERATOR MESSAGES

ENTERED PARAMETER IS ILLEGAL.

ILLEGAL FUNCTION.

DAYFILE MESSAGES

BOTH FAMILY AND PACK NAME.

Family and pack name may not both be specified.

FAMILY/PACK NOT FOUND.

Family or pack specified is not defined in the permanent file system.

USER NUMBER INVALID.

User number cannot be converted to user index correctly

PF SPECIFIED BUT UI NOT.

Filename has been designated but no associated user index is entered.

2.5.2 PERMANENT FILE UTILITY PROCESSOR (PFU)

DAYFILE MESSAGES

ALTERNATE DEVICE NOT FOUND.

PFU is unable to locate an alternate device in the system.

DEVICE NOT INITIALIZED.

PFLOAD attempting to load to uninitialized master device

DUPLICATE FILE NAME.

File already used at control point

EQxx TRACK LIMIT.

No remaining tracks available on EQxx

FET POINTERS OUT OF BOUNDS.

Out pointer is greater than limit pointer.

NO DEVICE SPECIFIED.

Device number 0

PFU ABORTED.

PFU has detected an error flag at its control point.

PFU - BUFFER CONTROL WORD ERROR ON filenam AT xxxxxx.

Word count of sector to be read from central memory exceeds word count limit of a sector (100B). The file being processed is filnam with FET address xxxxxx.

PFU - CATALOG TRACK NOT FOUND.

No permanent file catalog track could be found for the user index being processed.

PFU - FILE NOT FOUND.

FNT entry for file requested not found in FNT

PFU - I/O SEQUENCE ERROR ON filnam AT xxxxxx.

An operation was requested on file filnam before the previous operation was completed. xxxxxx is the FET address of file filnam.

PFU - NOT SYSTEM JOB.

Calling program is not system origin and does not have system origin privileges with debug set on.

PFU - PARAMETER ERROR.

Data in PFU call in error

PFU - TRACK INTERLOCK ALREADY CLEAR ON filname AT xxxxxx.

When called to clear the track interlock on file filname, PFU found that the FST for filname indicated that the interlock was already clear. xxxxxx is the FET address.

PFU - TRACK INTERLOCK ALREADY SET ON filname AT xxxxxx.

When called to set the track interlock on file filname, PFU found that the FST for filname indicated that the interlock was already set. xxxxxx is the FET address.

2.5.3 PERMANENT FILE DUMP OVERLAY

INFORMATIVE OPERATOR MESSAGES

ACCESSED AFTER yy/mm/dd. hh.mm.ss.
ALL FILES FOR USER INDEX xxxxxx.
CHECK DAYFILE FOR ERRORS.
CLEARING PF ACTIVITY COUNT.
CREATED AFTER yy/mm/dd. hh.mm.ss.
DUMPING - DIRECT ACCESS FILES ONLY.
DUMPING - INDIRECT ACCESS FILES ONLY.
DUMPING (filename)(user index)
FILENAME xxxxxxx USER INDEX xxxxxx.
GENERATING CATALOG IMAGE.
MODIFIED AFTER yy/mm/dd. hh.mm.ss.
NO FILES PROCESSED.
P. F. DEVICE (devicnm) DUMPED.
PFDUMP DEVICE MASK xxx.
PFDUMP DEVICE (devicenm) FAMILY (familynam).
PFDUMP DEVICE (devicenm) PACK (packnam).
PFDUMP yy/mm/dd. hh.mm.ss.
SETTING PF ACTIVITY COUNT.
WAIT FOR CATALOG INTERLOCK.
WAIT FOR PF UTILITY ON xx.

FATAL SYSTEM ERROR MESSAGES

These errors result in PFDUMP aborting.

CATALOG INDEX OUT OF RANGE

Catalog buffer location not in buffer range

INITIALIZATION IN PROGRESS.

Device cannot be accessed because it is being initialized.

WPE UNRECOVERED - ABORT.

Operator has aborted PFDUMP when WPE UNRECOVERED was detected on archive file.

USER DOES NOT RESIDE ON MASTER DEVICE.

User index specified does not reside on device specified.

DEVICE SPECIFIED NOT FOUND.

Device to be processed not in system

NO PF DEVICE IN EST.

No PF device is defined in the system.

NONFATAL SYSTEM ERROR MESSAGES

These errors cause PFDUMP to skip the dumping of the file to the archive tape.

STATUS ERR (filename) (user index)

Bad device or status 0 for a direct access file. The file is skipped.

INDIRECT TOO LONG (filenam) (userin).

The file is truncated.

INDIRECT TOO SHORT (filenam) (userin).

The file is padded with EOFs.

DIRECT TOO LONG (filename) (userin).

The file is truncated.

DIRECT TOO SHORT (filename) (userin).

The file is padded with EOFs.

DAF BUSY (filenam) (userin).

DAF ZERO LENGTH (filenam) (userin).

BAD SYSTEM SECTOR (filenam) (userin).

PERMIT RI RANGE ERR (filenam) (userin).

UNRECOVERED PARITY ERROR -
ENTER K.GO TO CONTINUE.
K.END TO ABORT.

2.5.4 PERMANENT FILE LOAD OVERLAY

NONFATAL SYSTEM ERROR MESSAGES

CATALOG CONTROL WORD MISSING.

Encountered other than catalog control word on initial control word of archive record

PERMITS PRESENT THAT SHOULD NOT BE.

Permit block found on tape but no previous permit random index found in catalog.

PERMITS MISSING.

No permit block present but there was a previous permit random index in the catalog, or entire permit block not read up yet and next word is not a permit control word.

NO DATA BLOCK.

Encountered other than data control word when expecting data for current file

MISSING EOR.

Logical EOR is missing, invalid data.

NO EOI FOR FILE.

Next catalog found before physical EOR found for current file

PARITY ERR (lfn) (u. i.).

Parity error encountered while loading file lfn; file is skipped.

STATUS ERR (lfn) (u. i.).

Status error on DAF (lfn); file is skipped.

FILE TRUNCATED (lfn) (u. i.)

File on archive tape is too short either with error padded EOFs or tape error. File is truncated and the length in the catalog is updated to reflect the smaller size.

TAPE ERROR (lfn) (u. i.)

Issued when certain of preceding errors occur. File is noted and skipped, and loading is resumed.

TAPE PARITY ERROR.

Parity error encountered. File name unknown. Tape skipped to next EOR.

ALTERNATE DEVICE NOT FOUND.

Device residency specified in catalog not available in this system and destination device is not specified. This message is accompanied by (STATUS ERROR).

FATAL SYSTEM ERROR MESSAGES

These errors result in PFLOAD being aborted.

ANY LOADING TO THIS POINT IS
INCOMPLETE A REINITIALIZE
AND RELOAD IS REQUIRED.

DESTINATION DEVICE NOT FOUND.

DEVICE SPECIFIED NOT FOUND.

ERROR IN CATALOG IMAGE ON PFDUMP TAPE.

ILLEGAL CATALOG TRACK COUNT.

NO PF DEVICE IN EST.

NO USER INDEXES ON TAPE MATCH DEVICE MASK.

PARITY ERROR IN CATALOG IMAGE RECORD.

PFLOAD ABORTED.

USER INDEX NOT ON DEVICE.

INFORMATIVE OPERATOR MESSAGES

ACCESSED AFTER yy/mm/dd. hh.mm.ss.

ALL FILES FOR USER INDEX (userin).

CHECK DAYFILE FOR ERRORS.

CLEARING UTILITY INTERLOCK.

CREATED AFTER yy/mm/dd. hh.mm.ss.

FILENAME (filenam) USER INDEX (userin).

LOADING - DIRECT ACCESS FILES ONLY.

LOADING - INDIRECT ACCESS FILES ONLY.

LOADING (filenam) (userin).

LOADING FROM (armask) TO (dvmask).

MODIFIED AFTER yy/mm/dd. hh.mm.ss.

NO FILES PROCESSED.

P. F. DEVICE (devicnm) LOADED.

PACKNAME (packnam) LOADED.

PFLOAD DEVICE (nn) FAMILY (familynam)

PFLOAD DEVICE (nn) PACK (packnam)

READING (filenam) (userin).

SETTING UTILITY INTERLOCK.

The action to be taken for the following five messages is to make a check to determine whether a dump tape is being assigned and that the proper label information being requested is in the label.

ASSIGN TAPE.

LABEL BAD.

TAPE LABEL IO ERROR.

REEL SEQUENCE ERROR.

FORMAT ERROR.

The following messages require K display input for incremental load operations.

TAPE SEQUENCE ERROR.

REENTER NL OR IL PARAMETERS IF DESIRED.

PROPER SELECTIVE DUMP TAPE NOT ASSIGNED.

```
ENTER E   TO TERMINATE LOADING.
      L   TO LIST REMAINING FILES.
      GO  TO RESUME INCREMENTAL LOAD.
```

2.5.5 CATALOG PERMANENT FILE DEVICE OVERLAY

SYSTEM ERROR MESSAGES

PFCAT COMPLETE.

Catalog of permanent file device completed.

EQUIP. NOT IN PFCAT TABLE.

There is no entry in PFCATs tables for the equipment type being cataloged. Therefore, no percent device usage can be determined.

INDEX BUFFER LIMIT.

Index buffer length (INDBL) must be increased to accommodate all user indexes on a catalog track.

DEVICE ERROR.

Device number refers to a nonmaster device.

ILLEGAL DEVICE NUMBER.

Alternate device not defined in system

DEVICE NOT FOUND.

Device specified to catalog cannot be located.

USER INDEX NOT ON DEVICE.

User index specified does not belong on device specified.

NO FILES PROCESSED.

No files were cataloged.

CHECK DAYFILE FOR ERRORS.

Nonfatal error(s) are in dayfile.

OPERATOR MESSAGES

CATALOGING (filename) (user index).

Identification of user number being cataloged

WRITING SUMMARY.

Summary report being generated

CATALOGING COMPLETED.

PFCAT completed.

SETTING PF. ACTIVITY COUNT.

CLEARING PF ACTIVITY COUNT.

WAIT FOR PF UTILITY ON xx.

2.5.6 CATALOG PERMANENT FILE ARCHIVE TAPE OVERLAY

SYSTEM ERROR MESSAGES

CATALOG COMPLETE.

Completion of catalog

PFATC ABORTED.

If error flag gets set

OPERATOR MESSAGES

CATALOGING (file name) (user index).

Identification of current file being loaded

ASSIGN TAPE.

Assign file to be cataloged

PARITY ERR (lfn) (u. i.).

Parity error on file (lfn) (Refer to next message.)

TAPE PARITY ERROR.

Always follows above message and is present when file name is unknown.

PREMATURE END OF FILE.

End-of-file detected before end of dump control word.

NO FILES PROCESSED.

No files were cataloged.

CHECK DAYFILE FOR ERRORS.

One nonfatal error at least has been detected. Read dayfile.

2.5.7 PERMANENT FILE ARCHIVE FILE COPY OVERLAY

OPERATOR MESSAGES

END OF COPY.

Program complete

READING (filename) (user index).

COPYING (filename) (user index).

DAYFILE MESSAGES - SYSTEM ERRORS

TAPE PARITY ERROR.

Parity error encountered. File name unknown. Tape skipped to end-of-file (physical EOR).

PARITY ERR (lfn) (u. i.).

Parity error on file lfn. File is skipped and execution resumed.

CATALOG CONTROL WORD MISSING.

Catalog control word expected but not found.

ASSIGN FILE.

Assign archive file.

PFCOPY ABORTED.

PFCOPY has encountered CP error. Device in catalog is not correct.

DATA CONTROL WORD ERROR.

Data control word expected but not correct identification

NO FILES PROCESSED.

No files have been copied.

CHECK DAYFILE FOR ERRORS.

A nonfatal error has been detected. Read dayfile.

TERMINAL NETWORK 3

Under NOS, terminals are grouped into sets. Each set is connected to a multiplexer that maintains communication with central memory through a PPU. A collection of multiplexers and their associated terminals connected through one PPU constitutes a terminal network.

A terminal is either a time-sharing terminal or a transaction terminal. In either case, the multiplexer for the terminal is attached to the Time-Sharing Executive at control point 1. Under Time-Sharing Executive direction, communication can be made with other control points. The transaction terminal, under Time-Sharing Executive direction, extends communication to the Transaction Executive (TRANEX) at control point 2. Processing under TRANEX makes use of the Transaction Subsystem, an extension of NOS.

3.1 TERMINAL NETWORK DESCRIPTION

The configuration and operational parameters of a terminal network are specified in a terminal description file. This file is a sequence of directives, one for each multiplexer and one for each terminal. Each set of terminal directives immediately follows the directive for the multiplexer through which these terminals will communicate with the system.

The network description file is read and interpreted by the common deck COMCRTN. This produces a table for each multiplexer directive (TMDK table) and a table for each terminal directive (TTDK) table. These tables contain the keywords that constitute each description. The Time-Sharing Executive and TRANEX use these tables during initialization to put the network on the air and exercise its capabilities.

The network description file must be a direct access public file under user index 377777B on the default family permanent files. The default file name is NETWid† (referred to as NETWORK in error messages). If that is not found, the alternate file SIMFid† (referred to as SIMFILE in error messages) will be used. If neither file has been defined, the Time-Sharing Executive will default to all TTY time-sharing terminals, and TRANEX will not completely initialize. Entries on the selected network file that do not have a match in the equipment status table will be ignored by the Time-Sharing Executive. In such a case, a dayfile message will be issued at initialization time.

The alternate network description file SIMFid is implemented in such a way that a simulated network description can be retained and used without the necessity of changing the description of the actual physical network on the NETWid file. Accordingly, for the analyst to run a stimulation, he performs the following steps to obtain the simulated network description.

1. Stop Time-Sharing Executive and TRANEX.
2. Attach NETWid file in write mode from the stimulation job.
3. Initiate Time-Sharing Executive and TRANEX. (Since NETWid is attached in write mode, Time-Sharing Executive and TRANEX will attach the alternate file SIMFid.)
4. Begin stimulation.

†id is the 2-character machine id. This can be obtained by using the MACHID macro as described in volume 2 of the system reference manual.

3.2 DIRECTIVE FORMATS

A directive statement may be up to 90 columns but only columns 1 through 72 are interpreted. Any statement containing an asterisk in column 1 is a comment statement and is ignored.

3.2.1 MULTIPLEXER DEFINITION DIRECTIVE

The format of the multiplexer definition directive can be one of the following.

//muxtype, CH=n_1, EQ=n_2.

//DIAL.

//STIM, CH=n_1, EQ=n_2.

The muxtype parameter indicates the model designation of the multiplexer (either 6671 or 6676, depending on which multiplexer is used). The CH parameter gives the channel to which the multiplexer is attached, and the EQ parameter gives the equipment number of the multiplexer. The channel range (n_1) is 0 to 13B and 20B to 33B. The equipment number range (n_2) is 0 to 7.

The //muxtype form of this directive permanently associates a multiplexer with a set of terminals.

The //DIAL form of this directive does not refer to any particular multiplexer or port. It requests that all terminals specified under it be assigned to whatever multiplexer ports are available when such assignment is needed. In the latter equipment assignment, no multiplexer port is permanently restricted to a fixed set of equipment.

The //STIM form of this directive is used to specify a simulated set of terminals and is used only in connection with the TRANEX Stimulator. This feature is described in the TRANEX Reference Manual.

3.2.2 TERMINAL DEFINITION DIRECTIVE

The format of the terminal definition directive is:

/termnam,p_i=n_i, p_2=n_2, . . . , p_n=n_n

termnam   Mandatory, unique alphanumeric name (1 to 7 characters) that identifies the terminal

The pi=ni have the following possible values.

Keyword	Description
PN=n	Port number of the multiplexer with which the terminal is associated. It is required only if the terminal is associated with a specific multiplexer equipment; that is, if the last preceding multiplexer directive was a //6671, //6676, or //STIM. The port number n ranges from 1 to 16 for a 6671 multiplexer, 1 to 64 for a 6676 multiplexer, and 1 to 4096 if the multiplexer is stimulated.
PL=n	Port range limit. This is used in conjunction with PN to establish the range of ports that can be used with one multiplexer. This can only occur after the PN has been established. The range of n is PN+1 to 15 (6671 multiplexer), PN+1 to 63 (6676 multiplexer), or PN+1 to 4095 (stimulated multiplexer). PN is the previously declared port number. As an example, the terminal definition /TERMA,PN=2,PL=5 would connect terminal TERMA to port 2 and limit the remaining terminal definitions in this set to ports 3 through 7.
TT=aaa	Terminal type. aaa may be one of the following. AB Terminal will be identified by answerback. ID Terminal will be identified by terminal operator entry. NIX CYBERLOAN NIXDORF terminal
LS=nnnn	Line speed. Declares the terminal line speed (character rate). nnnn is in characters/second. The range of values is 1 through 2047.
PC=nnnnnnnn.	Polling code. Defines the terminal polling code for a terminal that is on the same line as other terminals. This terminal code enables the terminal to identify its own data from that of the other terminals on this line. The range is 0 to 16777215 (24 bits).

Under NOS, a TRANEX terminal is defined with the previous arguments as well as the following which are unique to TRANEX.

DB=aa	Data base name. Declares the 2-character data base name to be used by the terminal.
IS=aaa.	Initial status. Declares the terminal's initial on/off status. aaa may be ON or OFF. Default value is ON.
RS=n.	Data base read security. Declares the data base read security. n may range from 0 to 7. Default value is 0.

Keyword	Description
TT=*aaa	Declares the transaction terminal type. *aaa may be any of the following. *AB Terminal will be identified by answerback. *ID Terminal will be identified by terminal operator entry. *NIX CYBERLOAN NIXDORF terminal.
UA=nnnnnn.	User argument. Defines the contents of the user argument area. nnnnnn may range from 0 to 77777777B (24 bits). Default value is 0.
US=n.	Data base update security. Declares the data base update security. n may range from 0 to 7. Default value is 0.

3.3 VALNET VALIDATION PROGRAM

VALNET is a program used to validate the syntax and logic of a terminal network description file. Errors in the network description are diagnosed and error messages entered on a list file. If no errors are encountered, no list is produced. If the network description is error free, the analyst should then create a system file from the description file. This system file will be either SIMFid or NETWid, depending on its use. Typical coding would be:

```
X.DIS.

USER(usernumber,password,familyname)

SUI=377777.

DEFINE,NETWid/CT=L.

COPY(INPUT,NETWid)
```

The control statement used to call the VALNET program has the following format.

VALNET(p_1, p_2, p_3)

p_i

P   Terminal network description file name

P   COMPILE file is diagnosed.

P=fn   File fn is diagnosed, where fn is the name supplied by the user.

Default name is NETWid.

L   File to receive list of errors

L   LIST file receives error listing.

L=fn   File fn receives error listing, where fn is the name supplied by the user.

L=0   No list is produced.

Default name is OUTPUT.

NR   Do not rewind network description file before reading.

NR   Do not rewind.

Default causes a rewind before reading.

3.3.1 VALNET OUTPUT LISTINGS

For each error encountered in a network description file, VALNET produces two lines of listing. These lines have the following format.

Error Line	
Statement Number	Diagnostic Message

The error line is the faulty line, or in the case of an omission, a near subsequent line. As an example, the descriptive code

(line 3)   /TERMA.

(line 4)   /TERMB, PN=3.

is missing the mandatory port number in line 3. However, this is not diagnosed until line 4 is encountered. Line 4 is listed along with the appropriate diagnostic as follows:

```
TERMB, PN=3
CARD 04 PORT NUMBER UNDEFINED FOR PREVIOUS TERMINAL.
```

List lines are formatted for 72 columns except for input lines that exceed this length.

VALNET calls common deck COMCRTN to process the multiplexer and terminal description directives. Following is a list or error messages which are placed on the output file when errors are detected by COMCRTN.

ARGUMENT VALUE MISSING.

A p_i = was encountered but not equivalenced.

CHANNEL NUMBER OUT OF RANGE.

The channel specified was not between 0 and 13B or 20B and 33B.

DATA BASE NAME FIRST CHARACTER BAD.

The first character of the TRANEX data base name is not alphanumeric.

DATA BASE NAME NOT 2 CHARACTERS.

The TRANEX data base name was not two characters in length.

EQUIPMENT NUMBER OUT OF RANGE.

The equipment number specified was not between 0 and 7.

KEYWORD NOT EQUIVALENCED.

A keyword was entered without a value associated with it.

LINE SPEED OUT OF RANGE.

A line speed value was entered which was not in the range from 1 to 2047.

MULTIPLEXER CHANNEL UNDEFINED ON PREVIOUS MUX DESCRIPTION.

No channel parameter was specified on the previous multiplexer definition directive. This is mandatory for //STIM, //6671, or //6676 directives.

MULTIPLEXER DESCRIPTION EXPECTED.

No multiplexer definition directive was specified. This directive should precede a set of terminal definition directives.

MULTIPLEXER EQUIPMENT UNDEFINED ON PREVIOUS MUX DESCRIPTION.

No equipment parameter was specified on the previous multiplexer definition directive. This is mandatory for //STIM, //6671, and //6676 directives.

NUMERIC CONSTANT ERROR.

Indicates any of the following.

1. Nonnumeric characters were present in a numeric constant.
2. An 8 or 9 was present in a constant value with a postradix B.
3. A character followed the postradix.

PN DESIGNATION MUST PRECEDE *PL*.

The PN parameter was specified after the PL parameter.

POLLING CODE OUT OF RANGE.

The polling code specified was not within the range of 0 to 77777777B.

PORT LIMIT OUT OF RANGE.

The port limit specified was not between PN+1 and 15 (for a 6671 multiplexer), PN+1 and 63 (for a 6676 multiplexer), or PN+1 and 1022 (for a simulated multiplexer).

PORT NUMBER UNDEFINED FOR PREVIOUS TERMINAL.

The port number was not specified on the previous terminal definition directive. This is required for //STIM, //6671, and //6676 directives.

PREVIOUS KTS TERMINAL MAY NOT SPECIFY PORT RANGE.

A port range was specified on a TRANEX terminal definition directive. PL is an illegal argument for a TRANEX terminal.

PSUEDO MULTIPLEXER DESCRIPTION ERROR.

Channel/equipment information has been specified on a //DIAL multiplexer definition directive.

READ SECURITY OUT OF RANGE.

The data base read security was not in the range of 0 to 7.

TERMINAL NAME DUPLICATES THAT OF KTS TERMINAL.

A terminal name was specified which was the same as an already specified TRANEX terminal.

TERMINAL NAME TABLE OVERFLOW.

Not enough storage is available to contain all the terminal description table entries. The job should be rerun with a greater field length specified.

TERMINAL NAME TOO LONG.

The terminal name was more than seven characters.

UNKNOWN INITIAL STATUS.

The initial status specified was neither ON nor OFF.

UNKNOWN KEYWORD.

An unknown keyword has been entered, or the keyword entered was not followed by =.

UNKNOWN MULTIPLEXER TYPE.

An entry other than 6671, 6676, DIAL, or STIM has been entered in the multiplexer definition directive.

UNKNOWN TERMINAL TYPE.

A terminal type other than AB, ID, NIX, *AB, *ID, or *NIX has been specified.

UPDATE SECURITY OUT OF RANGE.

The data base update security specified was not in the range of 0 to 7.

USER ARGUMENT OUT OF RANGE.

The user argument specified was not in the range of 0 to 77777777B.

PORT NUMBER OUT OF RANGE.

The port number specified was not in the range of 1 to 16 (6671 multiplexer), 1 to 64 (6676 multiplexer), 0 to 4096 (DIAL multiplexer), or 1 to 4096 (stimulated multiplexer).

3.3.2 DAYFILE MESSAGES ISSUED

NO DESCRIPTIONS FOUND.

No multiplexer and/or terminal descriptions were found on the description record.

FILE NAME CONFLICT.

The P file name (terminal network description) is the same as the L file name (error listing).

TERMINAL DESCRIPTION ERRORS.

Errors were found in the terminal description. Examine the listing for diagnostic messages.

VALNET ARGUMENT ERROR.

The VALNET control statement contains unknown or incorrectly used arguments.

VALNET COMPLETE.

All descriptions have been validated as correct.

TABLE OVERFLOW.

Not enough storage was available to hold all terminal description table entries.

EXPRESS DEADSTART DUMP INTERPRETER 4

The deadstart dump interpreter (DSDI) is a utility program, called by batch control statement, which converts selected portions of the binary information on an express deadstart dump file into reports to be listed. The express deadstart dump file is generated, on magnetic tape, by the express deadstart dump (EDD) utility which may be run at deadstart time after a system malfunction has occurred. Information on the EDD dump file is sequenced according to the illustration in Figure IV-4-1. Refer to part II, section 3.4 for complete information concerning the use of EDD.

Selection of data to be listed by DSDI is provided through input directives, either on an input file or on the DSDI control statement itself. Normal octal dumps of CM, ECS, and PPU memory can be produced by these directives as well as specially formatted dumps of specific system tables and buffers.

Features provided through the use of EDD and DSDI include the following.

- All binary information in the entire system is placed on the dump file created by EDD. This includes all of central memory, ECS, all PPU memory, the executing exchange package(s), and CDC CYBER 170 S/C registers, where applicable. This permits analysis of a system malfunction to be performed entirely off-line after the system has been recovered.

- Because DSDI copies the EDD dump file to a word-addressable random file on mass storage, dump data can be accessed in any order.

- EDD requires only a small amount of time during deadstart because the data is transferred in binary form to magnetic tape.

- The tape file created by EDD can be retained on magnetic tape or mass storage until it is no longer needed. Thus, a selective listing can be generated at any time.

- General information from the EDD dump file appears in the title and subtitle line of each page of listed output. The first 50 columns of each input directive are included in the title line of the output list it produces. An input directive is provided which enables insertion of comments into the subtitle line.

- Use of DSDI is possible from an interactive terminal as well as from the batch environment. The output produced by several directives is formatted for terminal output (72 columns). From a batch environment, output is formatted for a 136-column printer.

4.1 CALLING THE EXPRESS DEADSTART DUMP INTERPRETER

Processing of the EDD dump file is initiated with the DSDI control statement. The format of the statement is:

DSDI($p_1, p_2, \ldots, p_n$)

Status/Control Registers (CYBER 170 only)	*SCR* I	I - Bit 0 set if I (ignore error) deadstart option selected
	EXPRESS nn	nn = dump identifier
	hh.mm.ss.	Time from CM
	yy.mm.dd.	Date from CM
	-EOR-	
	SC16	One word header for channel 16 Status/Control register
	-EOR-	
	SC36	One word header for channel 36 Status/Control register, if it exists
	-EOR-	
PPUs	*PP*	
	EXPRESS nn	nn = dump identifier
	hh.mm.ss.	Time from CM
	yy.mm.dd.	Date from CM
	-EOR-	
	PP00	One word header for first PPU
	PP00 MEMORY	
	-EOR-	
	PP01	One word header for second PPU
	PP01 MEMORY	
	-EOR-	
	. . .	One record for each PPU
	PPnn	nn = last PPU number
	PPnn MEMORY	
	-EOR-	
CM	*CM* size	size = CM size/1000
	EXPRESS nn	nn = dump identifier
	hh.mm.ss.	Time from CM
	yy.mm.dd.	Date from CM
	-EOR-	
	CM CONTENTS	
	-EOR-	
CPU hardware registers from deadstart time	*CPR*	
	EXPRESS nn	nn = dump identifier
	hh.mm.ss.	Time from CM
	yy.mm.dd.	Date from CM
	-EOR-	
	CPU0 F	F - Bit 0 set in CPU0 header word if CPU0 turned off 20B word exchange package executing at deadstart
	-EOR-	
	CPU1 F	F - Bit 0 set in CPU1 header word if CPU1 turned off. This record exists only if two CPUs exist or CPU0 is turned off
	-EOR-	
ECS (if dumped)	*ECS* FR size	FR - ECS flag register, size = ECS size dumped/1000
	EXPRESS nn	nn = dump identifier
	hh.mm.ss.	Time from CM
	yy.mm.dd.	Date from CM
	-EOR-	
	ECS CONTENTS	
	-EOF-	

Figure IV-4-1. EDD Dump Tape Format

Each p_i is either a keyword or a keyword equated to a value. All keywords are optional and may be in any order within the parentheses. If a keyword is omitted, a default value is assumed. The value name represents a 1- to 7-character file name.

The following is a list of the valid keywords and values available.

Keyword	Description
D- Create random dump file	
omitted	Do not create random dump file.
D	Create random dump file. This file can then be used as the dump file on subsequent executions of DSDI, eliminating the need for reading the entire dump tape on each call.
F- Express dump file	
omitted	Read express dump from file DUMP.
F=name	Read express dump from named file.
I- Directive input	
omitted	Read directives from file INPUT.
I=name	Read directives from named file.
L- List output	
omitted	List output is written on file OUTPUT; the file is automatically printed.
L=name	List output is written to named file. It is the responsibility of the user to ensure that the file is saved or is printed.
NR- No rewind of express dump file	
omitted	EDD dump file is rewound before processing.
NR	EDD dump file is not rewound.
PD- Print density	
omitted	Print density is 6 lines per inch.
PD	Print density is 8 lines per inch.
PD=n	Print density is n lines per inch, where n may be 3, 4, 6, or 8.
P- Low core pointers option	
omitted	Use low core pointers from EDD dump file.
P	Use low core pointers from running system. Selecting the P option causes the low core pointers from the running system to be used to locate tables and buffer areas on the EDD dump file. This option is typically used when it is known that the low core pointers on the EDD dump file were destroyed by the system malfunction (for example, a CPUMTR error exit leaves an exchange package in memory locations 0 through 20 octal). Directives used to dump low core will dump the low core pointers contained on the EDD dump file, not those from the running system. This option should only be used when the configuration of the running system is the same as the system in use when the EDD dump file was created.

Keyword	Description
Z- Control statement input directives option omitted	Control statement does not contain input directives.
Z	The DSDI control statement contains the input directives following the closing parenthesis. The I keyword is ignored. This eliminates the need to use a separate input file for the directives when only a few directives are required.

When input directives appear on the DSDI control statement, a separator must follow the closing parenthesis of the statement, and must separate each directive. The separator character may be any character that will not be used in a directive. Each directive enclosed by separators must be terminated by a period. Likewise, the final directive must be terminated by a period although a final separator is not necessary. The directives may extend to column 72 on the statement. Continuation cards are not permitted.

For example (slant bar used for separator):

```
DSDI(Z)/SC./XP./P./D,0,20000./EC./D,0,10000.
```

If the directives are included in the input file, the following equivalent job would appear.

```
  :
DSDI.
7/8/9   (EOR)
SC.              Dump CDC CYBER 170 S/C register
XP.              Dump executing exchange pack-
                 ages
P.               Dump all PPUs
D,0,20000.       Dump the first 20000 octal
                 locations of CM
EC.              Set memory type to ECS
D,0,10000.       Dump the first 10000 octal
                 locations of ECS
6/7/8/9 (EOI)
```

A request for the EDD dump tape must precede the DSDI control statement. Since EDD writes express dump information on an unlabeled, 7- or 9-track tape, at a density of 800 bpi, the request should appear as follows:

$$\text{REQUEST(DUMP, D=800,} \begin{Bmatrix} \text{NT} \\ \text{MT} \end{Bmatrix}\text{, F=X, LB=KU, VSN=DUMP)}$$

It is recommended that the vsn parameter be specified in the request. If this is done, the request is presented in the resource mounting preview display and the job is rolled-out until the tape is mounted and ASSIGNed. Although the default express dump file name (DUMP) is used in this example, a different file name may be specified. This is provided that the same file name is also specified on the DSDI control statement (F keyword).

4.2 INPUT DIRECTIVES

DSDI input directives provide the capability to selectively dump only those portions of the EDD dump file that are of interest. The input directives are grouped into the following categories of function.

- List Control Directives

  Allows user to control line printer page eject and print density, and to specify comments in subtitle line of the output listing.

- File Manipulation and Control Directives

  Used to specify alternate files for DSDI input directives and listing output.

- CM/ECS Dump Directives

  Provides octal dump of specified portion of central memory or ECS. Absolute or relative addresses may be dumped (three dump formats are available).

- PPU Dump Directives

  Provides octal memory dump of all or selected PPUs (two dump formats are available). PPU analysis data can be included in the dump.

- CMR Dump Directives

  Provides specially formatted dumps of selected areas in central memory resident. These areas are specified by name rather than by address.

- Subsystem Dump/Analysis Directives

  Provides specially formatted dumps of subsystem control points and associated tables and buffers.

- Hardware Register Dump Directives

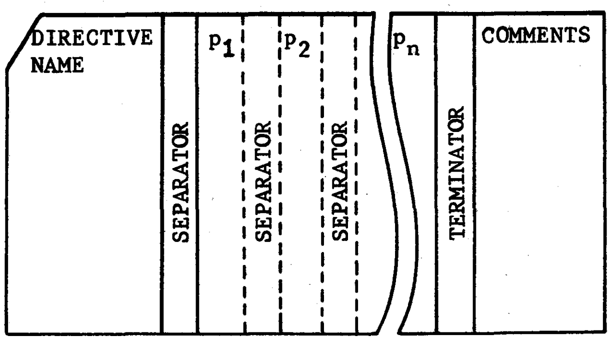
  Allows user to dump specified hardware registers.

Several of the DSDI input directives have interactive capabilities. The output produced by these directives is specially formatted for listing at an interactive terminal. Refer to section 4.3 for information concerning directive entry and use of DSDI from an interactive terminal.

A condensed summary of all DSDI input directives is contained in section 4.4. Section 4.5 contains examples of printer output listing produced by several of the DSDI input directives.

4.2.1 DIRECTIVE FORMAT

A directive has the following format. †



Directive Name

The directive name starts in column 1. It is terminated by a separator or terminator character.

Separator

Any character, including a space, other than the following.

A-Z 0-9 + - *

A period is the directive terminator; therefore, it cannot be used as a separator.

p_i

Parameter. Depending on the requirements of the directive, the directive may have no parameters or a number of parameters. Directives which do not process parameters will ignore everything beyond the first separator. Directives which require a fixed number of parameters will ignore everything beyond the separator for the last legal parameter.

Terminator

The explicit directive terminator is a period. Anything beyond the period is ignored.

Any characters following the directive terminator are considered comments and are ignored by DSDI. However, the comments are included with the directive in the title line of each page of the output listing (combined total of 50 characters appear in the listing).

† Although the directive format shown illustrates a directive as it would appear in a batch job deck, the same format is used when directives are stored on a file or entered from an interactive terminal. Refer to section 4.3 for additional information concerning directive entry from an interactive terminal.

4.2.2 LIST CONTROL DIRECTIVES

The list control directives provide control over DSDI printer output listings. This control includes density and page eject specification, as well as, a capability to add comments in the subtitle line of each page listed.

EJON - TURN ON AUTO PAGE EJECT

The EJON directive enables auto page eject (default condition). DSDI automatically issues a page eject function before listing the output produced by each new directive processed. This directive has no effect unless auto page eject has been disabled by the EJOFF directive.

Format:

```
EJON.
```

EJOFF - TURN OFF AUTO PAGE EJECT

The EJOFF directive disables auto page eject. Until this directive is processed, DSDI automatically issues a page eject function before listing the output produced by each new directive.

Format:

```
EJOFF.
```

EJ - FORCE PAGE EJECT

This directive is used to force DSDI to issue a page eject function before listing the output produced by the next directive processed. The EJ directive can also be used to force a page eject upon reaching a specified point on the page being printed. The page eject function is performed automatically unless disabled by the EJOFF directive.

Format:

```
EJ,nn.
```

nn	Force page eject only if less than nn decimal lines remain on the current page. If nn is omitted, page eject is forced before listing the output from the next directive processed.

PD - RESET PRINT LINE DENSITY

This directive is used to reset the print line density to a value other than that specified on the DSDI control statement.

Format:

```
PD,n.
```

n	Specifies new print line density in number of lines per inch (3, 4, 6, or 8). If n is omitted or an illegal value is specified, the message ILLEGAL PRINT DENSITY SELECTION is issued.

*. - COMMENT IN SUBTITLE LINE

This directive is used to specify a comment that appears in the subtitle line of each page listed.

Format:

```
*.ccc...ccc
```

ccc...ccc	Comment (up to 36 characters are printed)

4.2.3 FILE MANIPULATION AND CONTROL DIRECTIVES

File manipulation and control directives provide the capability to specify alternate files for DSDI input directives and listing output.

READ - READ ALTERNATE DIRECTIVES FILE

The READ directive causes DSDI to temporarily stop reading the current directives file and begin reading directives from the specified record on the named alternate file, or from current position if the record name is omitted. DSDI reads from the specified alternate directives file until an end of record is encountered (end of file or empty record if * is specified) and then resumes with the next directive on the original input directives file.

Format:

```
READ,lfn.
READ,lfn,name.
READ,lfn,*.
```

lfn — Name of alternate directives file (local file)

name — Optional record name; file lfn is searched for the specified record from current position to end-of-file or an empty record. If DSDI is unable to find the named record, it issues the message:

RECORD NOT FOUND.

If the record name is omitted, DSDI reads directives from the current position to end-of-record. Records must be in text format where the first word of the record is the record name.

* — Optional; if specified, DSDI reads directives from all records until end-of-file or an empty record.

OUTPUT - ASSIGN OUTPUT TO ALTERNATE LIST FILE

This directive is used to temporarily assign DSDI listing output to a file other than that specified on the DSDI control statement. When the alternate file is disposed to the print queue (refer to DISPOSE directive), output resumes on the original output file. If the alternate file is not disposed, both the original and the alternate output files remain at the job's control point as local files. Refer to section 4.3 for additional information concerning use of this directive from an interactive terminal.

Format:

```
OUTPUT,lfn.
```

lfn — Name of alternate list file (1 to 7 characters). Only one alternate output file may be active at a time. The file name (lfn) cannot be the same as the normal output file.

DISPOSE - DISPOSE ALTERNATE LIST FILE TO PRINT QUEUE

The DISPOSE directive causes the alternate list file specified by the OUTPUT directive to be disposed to the print queue. DSDI listing output then resumes on the original output file. This directive has no effect unless output has previously been assigned to an alternate list file (refer to OUTPUT directive). Refer to section 4.3 for additional information concerning use of this directive from an interactive terminal.

Format:

```
DISPOSE,usernum.
```

usernum — Route listing to remote batch terminal logged-in under user number usernum. If omitted, the listing is printed at a central site line printer.

REWIND - REWIND FILE

The REWIND directive repositions the specified file to beginning-of-information.

Format:

```
REWIND,lfn.
```

lfn — Name of file to be rewound

4.2.4 CM/ECS DUMP DIRECTIVES

The CM/ECS dump directives provide the capability to dump any portion of central memory or extended core storage in instruction parcel, byte, or word format. Display code character equivalents are included with each format. Either absolute or relative memory locations may be dumped. Refer to section 4.2.6 for directives used to dump specific portions of NOS Central Memory Resident (CMR).

DUMP CONTROL DIRECTIVES (CM, EC, RA, RAC)

Control directives are provided to select the type of memory to be dumped (CM or ECS) and the addressing mode to be used (absolute or relative).

CM - SET MEMORY TYPE TO CM

This directive specifies that subsequent C, D, and E directives dump central memory locations. If memory type is not specified, CM is assumed by default.

Format:

```
CM.
```

EC - SET MEMORY TYPE TO ECS

The EC directive specifies that subsequent C, D, and E directives dump extended core storage (ECS) locations. Unless this directive is specified, central memory locations are dumped by default.

Format:

```
EC.
```

RA - RESET REFERENCE ADDRESS

This directive specifies that subsequent C, D, and E directives dump memory locations relative to a specified reference address. Unless the RA or RAC directive is entered, absolute memory locations are dumped by default.

Format:

```
RA, nnnnnnn.
```

nnnnnnn	Reference address; addresses specified on subsequent C, D, and E directives are relative to this address.

Clearing the reference address specified on the most recent RA or RAC directive reenables absolute addressing. This is done by entering the RA directive in the following format.

```
RA, 0.
```

RAC - RESET REFERENCE ADDRESS TO RA OF CONTROL POINT

The RAC directive specifies that subsequent C, D, and E directives dump memory locations relative to the reference address of a specified control point. Unless the RA or RAC directive is entered, absolute memory locations are dumped by default.

Format:

```
RAC,nn.
```

nn — Control point number; addresses specified on subsequent C, D, and E directives are relative to the reference address of this control point.

Refer to the description of the RA directive to reenable absolute addressing.

MEMORY DUMP DIRECTIVES (C, D, E)

The following directives specify the area of memory to be dumped and determine the format of the output listing (refer to section 4.5 for sample output listing). The CM and EC directives determine the type of memory to be dumped (default is central memory). Absolute memory locations are dumped unless relative addressing has been enabled (refer to RA and RAC directives).

C - DUMP MEMORY IN INSTRUCTION PARCEL FORMAT

The C directive causes the specified locations of central memory or ECS to be dumped in four groups of five octal digits (three words per line) with display code character equivalents. Repetitive data is suppressed.

Format:

```
C,fwa,lwa.
```

fwa — Specifies first-word address to be dumped (mandatory)

lwa — Specifies last-word address, plus one location, to be dumped. If lwa is omitted, fwa+1 is assumed by default.

The output listing is read from top to bottom by column rather than across the page. Refer to section 4.3 for additional information concerning use of this directive from an interactive terminal.

D - DUMP MEMORY IN BYTE FORMAT

The D directive is used to dump specified locations of central memory or ECS in five groups of four octal digits (three words per line) with display code character equivalents. Repetitive data is suppressed.

Format:

```
D,fwa,lwa.
```

fwa — Specifies first-word address to be dumped (mandatory)

lwa — Specifies last-word address, plus one location, to be dumped. If lwa is omitted, fwa+1 is assumed by default.

The output listing is read from top to bottom by column rather than across the page. Refer to section 4.3 for additional information concerning use of this directive from an interactive terminal.

E - DUMP MEMORY IN WORD FORMAT

The E directive provides the capability to dump specified locations in central memory or ECS in word format (four words per line) with display code character equivalents.

Format:

```
E,fwa,lwa.
```

fwa	Specifies first-word address to be dumped (mandatory)
lwa	Specifies last-word address, plus one location, to be dumped. If lwa is omitted, fwa+1 is assumed by default.

4.2.5 PPU DUMP DIRECTIVES

PPU dump directives provide the capability to obtain a memory dump of all or selected PPUs. Two dump formats are available.

P - DUMP PPU MEMORY IN BLOCK FORMAT

The P directive causes PPU memory to be dumped in block format where each block represents 100 octal words of memory. The blocks are read by column (top to bottom), where each column contains 10 octal 12-bit words numbered 0 through 7. There are 10 octal columns in each block, numbered 0 through 7. Repetitive data is not suppressed and zero words are represented by hyphens (----).

Format:

$P, n_1, n_2, \ldots, n_n.$

n_i	Number of PPU to be dumped. If omitted, all PPUs are dumped.

Q - DUMP PPU MEMORY IN LINE FORMAT

This directive is used to dump PPU memory in line format with display code character equivalents. Each line contains 20 octal bytes (PPU words) printed in two sets of 10 octal bytes. Each set consists of an address, 10 octal bytes, and display code character equivalents. Repetitive lines are suppressed and zero bytes are represented by hyphens (----). Refer to section 4.3 for additional information concerning use of this directive from an interactive terminal.

Format:

$Q, n_1, n_2, \ldots, n_n.$

n_i	Number of PPU to be dumped. If omitted, all PPUs are dumped.

AP - DUMP ANALYSIS OF PPU AND PPU MEMORY IN LINE FORMAT

The AP directive causes PPU memory to be dumped in line format with display code character equivalents (same format as Q directive). Repetitive lines are suppressed and zero bytes are represented by hyphens. An analysis of the PPU is printed before the memory dump. Analysis data includes the associated PPU communications area, resident entry point call addresses, and read-only variables in direct cells. Certain direct cell variables are verified and those in error are indicated.

NOTE
Correct operation of this directive requires that the PPU communication area on the EDD dump file be intact.

Format:

AP, n_1, n_2, . . . , n_n.

n_i Number of PPU to be dumped or a program name. If a program name is specified, all PPUs executing that program are dumped. A warning message is issued if an illegal number is specified or the program name is not found in any PPU. If n is omitted, all active PPUs are dumped.

Section 4.5 contains a sample of the printer output listing produced by this directive.

4.2.6 CMR DUMP DIRECTIVES

The CMR dump directives provide the capability to selectively dump specified areas of central memory resident. Figure IV-4-2 illustrates the layout of central memory and lists the appropriate directives used to dump each of the indicated areas. Refer to the NOS Instant Manual for detailed illustrations of each area shown.

Address	CMR Layout	Dump Directive
000 : 077	system pointers and control words	LC.
100 : 111	channel status table	LC.
112 : 122	status/control registers	LC.
123 : 177	reserved	LC.
200	control point areas	CP.
(n+1)*200	system control point	CP.
(n+2)*200	PP communication area	PP.
	dayfile buffer pointers	DP.
	equipment status table (EST)	EST.
	file name/file status table	FNT.
	mass storage tables (MST)	MST.
	job control area	JC.
	dayfile buffers	ACCOUNT. or ERRLOG.
	dayfile dump buffer	DDB.
	ECS/PP buffer	EPB.
	CPUMTR	MTR.
	resident peripheral library (RPL)	RPL.
	resident central library (RCL)	RCL.
	peripheral library directory (PLD)	PLD.
	central library directory (CLD)	CLD.

Figure IV-4-2. Central Memory Layout

Successful use of the CMR dump directives is dependent upon the integrity of central memory at the time EDD was performed. Most important is the integrity of the low core pointers on the EDD dump file. If these pointers are not intact, the dump produced by DSDI may prove meaningless. Thus, if it is suspected that the low core pointers are not intact, specifying the P keyword on the DSDI control statement allows DSDI to use the low core pointers from the running system. This option should only be used when the configuration of the running system is the same as the system in use at the time the EDD dump file was created. If the low core pointers on the EDD dump file are not intact, the integrity of the other areas of central memory is also questionable. In this case, the output produced by the CMR dump directives may be unpredictable.

LC - DUMP LOW CORE

This directive instructs DSDI to dump the contents of low core (that is, central memory locations 0 through 177 octal). Each word is divided into the appropriate parameter fields. Each field is listed on a separate line with a description of the parameter. The absolute address and display code character equivalents are also listed for each word.

Format:

```
LC.
```

CP - DUMP ACTIVE CONTROL POINT AREAS

The CP directive causes all active control point areas, or a selected subset, to be dumped. List options provide the ability to dump only desired portions of the control point area and other control point related data. Refer to section 4.3 for additional information concerning use of this directive from an interactive terminal.

Format:

```
CP.
```

$CP, n_1/ops, n_2/ops, \ldots, n_n/ops.$

n_i — Control point number, or job name

/ops — List options; a continuous string of up to 10 characters indicating the portion of the control point area, or control point related data, to be dumped. If list options are specified, they apply only to the control point number or job name (n) with which they are associated. Valid options are as follows:

- X — Dumps exchange package and parameter summary.
- T — Provides detailed dump of control point area with English description of each parameter field and SYSTEXT symbol for each word.
- A — Dumps job dayfile pointers and buffer in word format with display code character equivalents. This format is the same format as E Memory Dump directive.
- F — Provides dump of FNT/FST, EST, and mass storage track chain, if one exists, for all files attached to the specified control point.
- C — Dumps field length of specified control point in instruction parcel format with display code character equivalents. This format is the same format as C Memory Dump directive. Repetitive data is suppressed.

D	Dumps field length of specified control point in byte format with display code character equivalents. This format is the same format as D Memory Dump directive. Repetitive data is suppressed.
E	Dumps field length of specified control point in word format with display code character equivalents. This format is the same format as E Memory Dump directive.
G	Dumps control point area in instruction parcel format with display code character equivalents. This format is the same format as C Memory Dump directive.
H	Dumps control point area in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.
I	Dumps control point area in word format with display code character equivalents. This format is the same format as E Memory Dump directive.
P	Provides dump and analysis of all active PPU's associated with control point n.
default	If n is specified with no corresponding list options, options XTAF are selected automatically. Refer to section 4.5 for a sample of the printer output listing produced. The default options selected can be changed with the CPO directive.

All of the list options specified for a particular control point area (up to 10) are processed. For example, if the C and D options are both specified, the control point field length is dumped twice, once in instruction parcel (C) format and again in byte (D) format.

CPO - RESET DEFAULT LIST OPTIONS

The CPO directive provides the capability to select a new string of default list options for the CP directive.

Format:

```
CPO,ops.
```

ops	New default list options for CP directive, a continuous string of up to 10 characters. Refer to description of CP directive for list of valid option characters.

PP - DUMP PPU COMMUNICATION AREAS

The PP directive is used to dump the contents of all PP communication areas in byte format with display code character equivalents. This format is the same format as D Memory Dump directive. The control point assignment, channel assignment, and monitor function are listed with each communication area. Refer to section 4.5 for a sample of the printer output listing produced by this directive. Section 4.3 contains additional information concerning use of the PP directive from an interactive terminal.

Format:

```
PP.
```

DP - DUMP DAYFILE BUFFER POINTERS

The DP directive causes DSDI to dump the dayfile buffer pointers in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.

Format:

```
DP.
```

EST - DUMP EQUIPMENT STATUS TABLE

This directive causes the equipment status table to be dumped in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.

Format:

```
EST.
```

FNT - DUMP FILE NAME/FILE STATUS TABLE

The FNT directive is used to dump the file name/file status table (FNT/FST) in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.

Format:

```
FNT.
```

MST - DUMP MASS STORAGE/TRACK RESERVATION TABLES

This directive instructs DSDI to dump all mass storage and track reservation tables unless equipment numbers are specified, in which case only the specified equipment MTSs are dumped. The dump format for the mass storage tables reflects the appropriate parameter fields and SYSTEXT symbol of each word. The portion of the dump describing the track reservation tables is presented in byte format with display code character equivalents. The track link byte ordinal and status bits (three groups of four bits) are indicated for each word. Refer to section 4.5 for a sample of the printer output listing produced by this directive.

Format:

```
MST.
```

```
MST,eq1,eq2,...,eqn.
```

eq_i — Equipment number of equipment whose mass storage table is to be dumped.

JC - DUMP JOB CONTROL AREA FOR EACH ORIGIN TYPE

The JC directive causes the job control area for each job origin type to be dumped. The dump is formatted to reflect the appropriate parameter fields and SYSTEXT symbol for each word.

Format:

```
JC.
```

ACCOUNT - DUMP ACCOUNT DAYFILE BUFFER

This directive causes the account dayfile pointers and buffer to be dumped in word format (four words per line) with display code character equivalents. This format is the same as that produced by the E Memory Dump directive.

Format:

```
ACCOUNT.
```

ERRLOG - DUMP ERROR LOG DAYFILE BUFFER

The ERRLOG directive is used to dump the error log dayfile pointers and buffer in word format (four words per line) with display code character equivalents. This format is the same as that produced by the E Memory Dump directive.

Format:

```
ERRLOG.
```

DDB - DUMP DAYFILE DUMP BUFFER

The DDB directive causes the dayfile dump buffer to be dumped in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.

Format:

```
DDB.
```

EPB - DUMP ECS/PPU BUFFER

The EPB directive is used to dump the ECS/PPU buffer in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.

Format:

```
EPB.
```

MTR - DUMP CPU MONITOR

This directive instructs DSDI to dump CPU monitor. Exchange packages are dumped in exchange package format while the program area is dumped, using relative addressing, in instruction parcel format with display code character equivalents. This format is the same format as C Memory Dump directive.

Format:

```
MTR.
```

RPL - DUMP RESIDENT PERIPHERAL LIBRARY

This directive causes the resident PPU library to be dumped in byte format with display code character equivalents. This format is the same format as D Memory Dump directive. A header line for each PPU program dumped indicates the name of the program and its length in bytes. Each succeeding line contains 10 bytes (two central memory words) of the PPU program. The PPU address of the first byte in each line, relative to address zero of the PPU, is also listed.

Format:

```
RPL.
```

RCL - DUMP RESIDENT CENTRAL LIBRARY

The RCL directive is used to dump the resident central library in instruction parcel format with display code character equivalents. This format is the same format as C Memory Dump directive.

Format:

```
RCL.
```

PLD - DUMP PERIPHERAL LIBRARY DIRECTORY

This directive is used to dump the PPU library directory in byte format with display code character equivalents. This format is the same format as D Memory Dump directory.

Format:

```
PLD.
```

CLD - DUMP CENTRAL LIBRARY DIRECTORY

The CLD directive causes the central library directory to be dumped in byte format with display code character equivalents. This format is the same format as D Memory Dump directive.

Format:

```
CLD.
```

4.2.7 SUBSYSTEM DUMP/ANALYSIS DIRECTIVES

The directives which follow are used to selectively dump portions of central and/or PPU memory associated with a specific subsystem (BATCHIO, EI200, MAGNET, and TELEX). Although many other directives previously described in this section can be used to dump the same areas of memory, these directives dump those areas most frequently analyzed when subsystem related malfunctions occur. In addition, many of the dumps are specially formatted to provide a detailed description of the area being dumped. If the specified subsystem was not active at the time the EDD dump file was created, the following message is issued.

SUBSYSTEM NOT FOUND.

BATCHIO - DUMP ASSOCIATED MEMORY FOR ANALYSIS

This directive is used to selectively dump areas of central and/or PPU memory most frequently analyzed when BATCHIO malfunctions are indicated. The areas and type of memory dumped is determined by the list options specified.

Format:

BATCHIO, ops.

ops — List options; a string of characters indicating the areas of memory to be dumped. If no options are specified, both options (PB) are selected by default and are processed in order as listed.

P — Provides analysis and full memory dump of PPUs having resident copies of 1CD, 1IO, or 1BA. The output listing generated is the same (in format and content) as that produced by entering the AP directive in the following format.

AP, 1CD, 1IO, 1BA.

Refer to the description of the AP directive in section 4.2.5 for additional information.

B — Provides specially formatted dumps of each active BATCHIO buffer point. Included with the dump of each buffer point is the associated equipment type and FET as well as EST and FNT/FST entries.

EI200 - DUMP ASSOCIATED MEMORY FOR ANALYSIS

The EI200 directive causes DSDI to selectively dump areas of central and/or PPU memory most frequently analyzed when Export/Import malfunctions are indicated. The areas and type of memory dumped is determined by the list options specified. The following three words are always listed at the beginning of the listing, in byte format, regardless of which list options are selected.

1. Driver recall word for 1ED (DRCL)

2. Executive call word for 1LS (EXCL)

3. Input register for 1LS (IREX)

Format:

```
EI200,ops.
```

ops — List options; a string of characters indicating the areas of memory to be dumped. If no options are specified, all four options (LTPO) are selected by default, and are processed in order as shown.

- L — Dumps EI200 low core pointer words. The dump is formatted to reflect the appropriate parameter fields and SYSTEXT symbol for each word.
- T — Provides specially formatted dump of EI200 terminal tables with associated FET, EST, and FNT/FST. Each word of the terminal tables is formatted to reflect appropriate parameter fields. In addition, each word is preceded by a description of the parameter fields and its SYSTEXT symbol.
- P — Provides analysis and full memory dump of all PPUs containing copies of 1ED, 1LS, and 1SP. The output listing generated is the same (in format and content) as that produced by entering the AP directive in the following format.

  ```
  AP,1ED,1LS,1SP.
  ```

  Refer to the description of the AP directive in section 4.2.5 for additional information.
- O — Dumps PPU overlays that reside within the EI200 field length in byte format with display code character equivalents. A header line for each overlay dumped indicates its length in bytes. Each succeeding line contains 10 bytes (two central memory words) of the program. The PPU address of the first byte, relative to address zero of the PPU is also listed.

MAGNET - DUMP ASSOCIATED MEMORY FOR ANALYSIS

The MAGNET directive is used to selectively dump areas of central and/or PPU memory most frequently analyzed when a malfunction within MAGNET is indicated. The areas and type of memory dumped are determined by the list options specified.

Format:

```
MAGNET,ops.
```

ops List options; a string of characters indicating the areas of memory to be dumped. If no options are specified, all three options (UQP) are selected by default, and are processed in order as listed.

U Provides specially formatted dump of the MAGNET unit descriptor tables (UDTs) with associated FET, EST, and FNT/FST. Each word of a UDT is formatted to reflect appropriate parameter fields. In addition, each word is preceded by a description of the parameter fields and its SYSTEXT symbol. If extended labels are present, they appear with the FET in the output listing. The FET also indicates the address and control point number of the user.

Q Provides dump of the MAGNET queue table in byte format (two words per line) with display code character equivalents. The first word in each line is preceded by its ordinal within the table.

P Provides analysis and full memory dump of all PPUs having resident copies of 1MT. The output listing generated is the same (in format and content) as that produced by entering the AP directive in the following format.

```
AP,1MT.
```

Refer to the description of the AP directive in section 4.2.5 for additional information.

TELEX - DUMP ASSOCIATED MEMORY FOR ANALYSIS

This directive causes DSDI to selectively dump areas of central and/or PPU memory most frequently analyzed when TELEX malfunctions are indicated. The areas and type of memory are determined by the list options specified. The TELEX current entry word (SSPA) is always printed at the beginning of the listing, in byte format, regardless of which list options are specified.

Format:

```
TELEX,ops.
```

ops List options; a string of up to four characters indicating the areas of memory to be dumped. If no options are specified, all four options (CTEP) are selected by default, and are processed in order as listed.

C Provides dump of the TELEX command table.

T Provides specially formatted dump of the TELEX terminal table in which each word reflects the appropriate parameter fields. In addition, each word is preceded by a description of the parameter fields and its SYSTEXT symbol.

E Dumps TELEX reentry table in byte format (two words per line) with display code character equivalents. The first word in each line is preceded by its ordinal within the table.

P Provides analysis and full memory dump of all PPUs having resident copies of TLX, 1TA, 1TB, 1TC, 1TD, and 1TO. This option also provides an analysis and dump of all PPUs having resident copies of 1RO and 1RI that are associated with control points of TELEX origin. The output listing generated is the same (in format and content) as that produced by entering the AP directive in the following format.

```
AP,TLX,1TA,1TB,1TC,1TD,1TO,1RO,1RI.
```

The exception is that the AP directive also dumps all PPUs having copies of 1RO and 1RI rather than only those associated with control points of TELEX origin. Refer to the description of the AP directive in section 4.2.5 for additional information.

4.2.8 HARDWARE REGISTER DUMP DIRECTIVES

The following directives provide a capability to dump specified hardware registers.

XP - DUMP DEADSTART EXCHANGE PACKAGE

This directive instructs DSDI to dump the CPU exchange package executing at the time of deadstart. If there are two CPUs in the system, both exchange packages in execution at the time of deadstart are dumped.

Format:

```
XP.
```

SC - DUMP S/C REGISTER

This directive is valid only on a CYBER 170 series computer system and is used to dump the S/C registers.

Format:

```
SC.
```

4.3 INTERACTIVE USE OF DSDI

Incorporated within DSDI is an interactive facility which allows several of the directives described earlier in this section to be entered interactively from a time-sharing origin job. This interactive facility is designed to provide the analyst with the following additional capabilities.

- Allows preliminary examination of the EDD dump file to determine which areas should be listed in detail at a line printer.
- Allows examination of certain areas of the EDD dump file not listed during normal operational procedures following a system malfunction. Typically, predefined portions of the EDD dump file are listed following a system malfunction.
- Allows on-line examination of the EDD dump file from a remote location.

An example showing interactive use of DSDI is contained in section 4.3.2.

When the DSDI control statement is entered from an interactive terminal (BATCH subsystem only), a delay will be experienced before input directives can be entered. During this time (10 to 60 seconds), DSDI is copying the EDD dump file to a random mass storage file. The length of the delay is dependent upon device speed and current system activity. When DSDI is able to accept input directives, it will issue the following prompt to the terminal.

```
ENTER DIRECTIVES --
?
```

Directives are entered following the ?. Only one directive can be entered at a time, and is restricted to one line. The format is the same as described for batch input (refer to section 4.2.1).

Generally, any of the DSDI input directives can be entered at an interactive terminal. However, the output produced by many of the directives is formatted for listing only at a line printer (136 columns) and cannot be listed at the terminal (72 columns). The L keyword on the DSDI control statement initially determines the disposition of the list output. If a file name is not specified, list output is assigned to file OUTPUT by default (that is, the terminal). In this case, entry of directives which produce output that cannot be listed at the terminal results in the message:

```
DIRECTIVE RESTRICTED TO PRINTER OUTPUT.
```

If a list output file name is specified on the DSDI control statement, all input directives can be entered at the terminal. All list output (including error messages) is written to the specified file.

Two input directives are provided to further control the disposition of list output.

OUTPUT,lfn.	This directive is used to assign output to alternate file lfn (file name OUTPUT is illegal; that is, alternate list output cannot be assigned to the terminal). While this directive is active, all input directives can be entered at the terminal. All list output (except error messages) is written to file lfn and is formatted for transmission to a line printer. Error messages are written directly to the terminal.

DISPOSE.	This directive disposes the alternate list file (specified in OUTPUT directive) to the print queue. Output will be printed at the central site line printer. All subsequent list output resumes on the original output file specified on the DSDI control statement.

Refer to section 4.2.3 for additional information concerning use of these directives.

4.3.1 TERMINAL OUTPUT DIRECTIVES

The following directives produce output formatted for listing at an interactive terminal.

C - DUMP MEMORY IN INSTRUCTION PARCEL FORMAT

This directive causes the specified locations of central memory or ECS to be dumped in four groups of five octal digits (one word per line) with display code character equivalents. No pagination is processed for terminal output. The CM and EC directives (section 4.2.4) determine the type of memory to be dumped; default is central memory. The RA or RAC directive (section 4.2.4) must be entered to dump relative addresses; default is absolute addressing.

Format:

C, fwa, lwa. CR

fwa	Specifies first-word address to be dumped (mandatory)
lwa	Specifies last-word address, plus one location, to be dumped. If omitted, fwa+1 is assumed by default.

Example of terminal output:

```
? C,6230,6240.
0006230 34240 10100 00012 50036 1TAA   AU 3
0006231 00764 70000 00000 10113    *    AAK
0006232 04154 70000 00000 10113 DM*     AAK
0006233 00004 67446 74000 10005   - -  A E
0006234 05153 05700 00000 00000 EMX.
0006235 00000 00000 00000 00000
0006236 00000 00000 00000 00000
0006237 00000 00005 05111 14422       EEII9R
```

D - DUMP MEMORY IN BYTE FORMAT

The D directive is used to dump specified locations of central memory or ECS in five groups of four octal digits (one word per line) with display code character equivalents. No pagination is processed for terminal output. The CM and EC directives (section 4.2.4) determine the type of memory to be dumped; default is central memory. The RA or RAC directive (section 4.2.4) must be entered to dump relative addresses; default is absolute addressing.

Format:

D, fwa, lwa. (CR)

fwa	Specifies first-word address to be dumped (mandatory)
lwa	Specifies last-word address, plus one location, to be dumped. If omitted, fwa+1 is assumed by default.

Example of terminal output:

```
? D,6230,6240.
0006230 3424 0101 0000 0125 0036 1TA4  AU 3
0006231 0076 4700 0000 0001 0113   *    AAK
0006232 0415 4700 0000 0001 0113 DM*    AAK
0006233 0000 4674 4674 0001 0005   - -  A E
0006234 0515 3057 0000 0000 0000 EMX.
0006235 0000 0000 0000 0000 0000
0006236 0000 0000 0000 0000 0000
0006237 0000 0000 0505 1111 4422      EEII9R
```

Q - DUMP PPU MEMORY IN LINE FORMAT

This directive causes the specified locations of PPU memory to be dumped in line format. Each line contains 10 octal bytes (PPU words) with display code character equivalents. Repetitive lines are suppressed and zero bytes are represented by hyphens.

Format:

Q, n, fwa, lwa. (CR)

n	Number of PPU to be dumped
fwa	Specifies first-word address to be dumped
lwa	Specifies last-word address, plus one location, to be dumped

NOTE

fwa and lwa are automatically adjusted so that the dump limits fall within a multiple of 10 octal words.

This format is valid only for terminal output. If attempted from a batch origin job or while an alternate list file is active, the fwa and lwa parameters will be interpreted as PPU numbers.

Example of terminal output:

```
? Q,5,Ø,1ØØ.
ØØØØ  ØØØ3 2Ø2Ø 334Ø ---- Ø614 ---- 4334 Ø117    CPPØ5  FL  81AO
ØØ1Ø  ØØ64 ØØØ1 7772 Ø1ØØ ØØØ6 1Ø73 14Ø1 6Ø72      A  A  FH LA
ØØ2Ø  225Ø 3225 ---- ØØ27 ØØ12 44Ø2 5747 5751    R/ZU   W J93.*.(
ØØ3Ø  2Ø11 7646 ---- ØØØ1 Ø141 Ø6ØØ ---- ----     I -   AA6F
ØØ4Ø  15Ø1 1116 2Ø14 ØØ74 Ø2Ø3 ---- ---- ----    MAINPL  BC
ØØ5Ø  34Ø4 233Ø ØØ35 6213 17Ø7 ---- 4ØØØ 6675    1DSX 2 KOG  5
ØØ6Ø  ---- 4521 ---- 61Ø1 ØØØ1 Ø153 ØØØ1 Ø532     +Q   A AA$ AEZ
ØØ7Ø  ØØØ1 Ø1ØØ 1ØØØ ØØØ3 6ØØØ 625Ø 6251 6252    AA 4  C   / ( )
```

PP - DUMP PPU COMMUNICATION AREAS

The PP directive causes the PPU number, executing program name, control point assignment, and input register address for each PPU communication area to be dumped.

Format:

PP. (CR)

Example of terminal output:

```
                                     PPU Number
? PP.                                  Control Point Assignment

PPØØ   PPØ1   PPØ2   PPØ3   PPØ4   PPØ5   PPØ6   PPØ7   PP1Ø   PP11
MTR-Ø1 DSD-3Ø 1MT-25 1TD-Ø1 1LS-27 1SP-3Ø 1ED-27 1LS-27 1TA-Ø1 CIO-Ø5
62ØØ   621Ø   622Ø   623Ø   624Ø   625Ø   626Ø   627Ø   63ØØ   631Ø
                                     Input Register Address
                                Program in Execution
```

CP - DUMP ACTIVE CONTROL POINT AREAS

This directive is used to dump the job name and control point area address for each control point.

Format:

CP. (CR)

Example of terminal output:

```
                               Control Point     Control Point Number
? CP.                          Area Address        Job Name at Control Point

CP Ø1    CP Ø2    CP Ø3    CP Ø4    CP Ø5    CP Ø6    CP Ø7    CP 1Ø
TELEX    RANØAAX                    AJNQBSR
Ø2ØØ     Ø4ØØ     Ø6ØØ     1ØØØ     12ØØ     14ØØ     16ØØ     2ØØØ

CP 11    CP 12    CP 13    CP 14    CP 15    CP 16    CP 17    CP 2Ø
                           AJØIBTI
22ØØ     24ØØ     26ØØ     3ØØØ     32ØØ     34ØØ     36ØØ     4ØØØ

CP 21    CP 22    CP 23    CP 24    CP 25    CP 26    CP 27    CP 3Ø
                                    MAGNET            EXPORTL  SYSTEM
42ØØ     44ØØ     46ØØ     5ØØØ     52ØØ     54ØØ     56ØØ     6ØØØ
```

4.3.2 EXAMPLE OF DSDI TERMINAL USAGE

This example illustrates how DSDI might be used, following a system malfunction, to analyze portions of the EDD dump file from an interactive terminal. It is assumed that an EDD dump file was created during normal system recovery procedures. Vertical spacing has been expanded to permit commentary.

yy/mm/dd. hh.mm.ss. CDC MULTI-MODE OPERATING SYSTEM NOS 1 FAMILY: sys172 USER NUMBER: jba2973, b5y TERMINAL: 37, TTY RECOVER/CHARGE: charge, 6375, 47asqw	Complete log-in sequence
READY.	
batch $RFL, 20000.	Enter BATCH subsystem.
/request, dump, vsn=dump, lb=ku, f=x, mt, d=800	The REQUEST control statement is entered to assign the EDD dump tape to this job. Use of the VSN parameter allows the job to be rolled out while the tape is mounted and ASSIGNed.
/get, altdir	Retrieve alternate directives file ALTDIR (refer to example in Figure IV-4-3).
/dsdi.	Calls DSDI which copies EDD dump tape to a random mass storage file.
ENTER DIRECTIVES-- ? ? ? ?	Enter terminal output directives (section 4.3.1) to list any portion of the EDD dump file at the terminal. DSDI issued the prompt (?) when it is ready to accept a new directive.
? output, altout. ? ? ?	List output produced by subsequent directives is written to local file ALTOUT. This allows entry of directives which produce line printer formatted output.
? read, altdir. . . .	All input directives in alternate directives file ALTDIR are read and processed. List output is written to local file ALTOUT. DSDI does not request terminal input until last directive on ALTDIR is processed.
? dispose. ? ? CR	Dispose local file ALTOUT to the print queue for listing at the central site line printer. Output produced by subsequent directives is listed at the terminal
EXPRESS DUMP COMPLETE (FL USED xxxxxxB) /	DSDI is terminated by pressing carriage return in response to the ? prompt.

In summary, the following operations were performed by DSDI. First, small areas of the dump file were listed at the terminal for preliminary examination. This was done both to analyze the cause and effect of the system failure, as well as determine the extent of line printer listings required. An appropriate comment may be placed in the list file subtitle at this time via the *.ccc...ccc directive.

Next, directives were entered to generate the necessary line printer listings. These listings are generally extensive, or contain specially formatted output that cannot be listed at the terminal. Thus, output was written to an alternate list file named ALTOUT.

After all necessary directives had been entered from the terminal, an alternate directives file (ALTDIR) was read. ALTDIR is a permanent file containing input directives necessary to obtain a printer listing of specific areas in the dump file that are frequently examined following a system failure (for example, CMR tables and buffers, PPU memory, etc.). The alternate directives file eliminates the need to enter the directives individually. Figure IV-4-3 illustrates a typical alternate directives file. Comments describing areas of the dump file to be listed appear, with the directive, in the title line of the output listing.

When DSDI finished processing the last directive in file ALTDIR, it again issued the ? prompt to the terminal requesting further directive input from the keyboard. At this time, the DISPOSE directive was entered causing file ALTOUT to be printed at the central site line printer. Refer to section 4.5 for examples of printer output listings.

```
DUMP†
LC.                   LOW CORE POINTERS
PP.                   PPU COMMUNICATION AREA
CPO,H.
EJOFF.
EJ.
CP.                   CONTROL POINT AREAS
CPO,XTAF.
DP.                   DAYFILE BUFFER POINTERS
EJ.
EST.                  EQUIPMENT STATUS TABLE
FNT.                  FILE NAME TABLE
EJ.
MST.                  MASS STORAGE TABLES
JC.                   JOB CONTROL PARAMETERS
CP,SYSTEM/A.          SYSTEM DAYFILE BUFFER
ACCOUNT.              ACCOUNT FILE BUFFER
ERRLOG.               ERROR LOG BUFFER
EJON.
AP.                   ACTIVE PPUS
```

Figure IV-4-3. Typical Alternate Directives File

† All records in an alternate directives file must be in text format; the first word in each record contains only the record name.

4.4 DSDI INPUT DIRECTIVE SUMMARY

LIST CONTROL DIRECTIVES

Page		
IV-4-7	EJON.	Turn on auto page eject
IV-4-7	EJOFF.	Turn off auto page eject
IV-4-7	EJ, nn.	Force page eject before next directive, or if less than nn lines remain on page
IV-4-7	PD, n.	Reset print line density (n=3, 4, 6, or 8)
IV-4-8	*, ccc... ccc	Enter comment in subtitle line of output listing

FILE MANIPULATION AND CONTROL DIRECTIVES

IV-4-8	READ, lfn, name.	Read alternate directives file
IV-4-9	OUTPUT, lfn.	Assign output to alternate list file
IV-4-9	DISPOSE, usernum.	Dispose alternate list file to print queue, or remote batch terminal
IV-4-9	REWIND, lfn.	Rewind file

CM/ECS DUMP DIRECTIVES

IV-4-10	CM.	Set memory type to CM
IV-4-10	EC.	Set memory type to ECS
IV-4-10	RA, nnnnnnn.	Reset reference address to nnnnnnn.
IV-4-11	RAC, nn.	Reset reference address to RA of control point nn
IV-4-11	RA, 0.	Clear reference address specified by RA or RAC
IV-4-11, IV-4-25	C, fwa, lwa.	Dump memory in instruction parcel format (4 groups of 5)†
IV-4-11, IV-4-26	D, fwa, lwa.	Dump memory in byte format (5 groups of 4)†
IV-4-12	E, fwa, lwa.	Dump memory in word format

PPU DUMP DIRECTIVES

IV-4-12	P, $n_1, \ldots, n_n$.	Dump PPU in block format
IV-4-12	Q, $n_1, \ldots, n_n$.	Dump PPU in line format
IV-4-26	Q, n, fwa, lwa.	Dump PPU in line format from terminal†
IV-4-13	AP, $n_1, \ldots, n_n$.	Dump PPU in line format with analysis.

CMR DUMP DIRECTIVES

IV-4-15	LC.	Dump low core pointers
IV-4-15, IV-4-27	CP, n_1/ops, ..., n_n/ops.	Dump active control point areas† X - Exchange package and summary†† T - Detailed dump†† A - Job dayfile buffer†† F - Attached files†† C - FL in instruction format D - FL in byte format E - FL in word format G - CP in instruction format H - CP in byte format I - CP in word format P - Attached PPUs

CMR DUMP DIRECTIVES (Cont'd)

Page		
IV-4-20	CP0, ops.	Reset default list options
IV-4-21, IV-4-27	PP.	Dump PPU communication area†
IV-4-17	DP.	Dump dayfile buffer pointers
IV-4-17	EST.	Dump equipment status table
IV-4-17	FNT.	Dump file name table
IV-4-17	MST.	Dump mass storage table
IV-4-17	MST, $eq_1, \ldots, eq_n$.	Dump specified equipment mass storage tables.
IV-4-18	JC.	Dump job control parameters
IV-4-18	ACCOUNT.	Dump account dayfile buffer
IV-4-18	ERRLOG.	Dump error log dayfile buffer
IV-4-18	DDB.	Dump dayfile dump buffer
IV-4-18	EPB.	Dump ECS/PPU buffer
IV-4-19	MTR.	Dump CPU monitor (CPUMTR)
IV-4-19	RPL.	Dump resident peripheral library
IV-4-19	RCL.	Dump resident central library
IV-4-19	PLD.	Dump peripheral library directory
IV-4-19	CLD.	Dump central library directory

SUBSYSTEM DUMP DIRECTIVES

IV-4-20	BATCHIO, ops.	Dump associated memory B - Buffer points †† P - 1CD, 1IO, and 1BA††
IV-4-21	EI200, ops.	Dump associated memory L - Low core pointers†† T - Terminal tables†† P - 1ED, 1LS, and 1SP†† O - PPU overlays††
IV-4-22	MAGNETIC, ops.	Dump associated memory U - UDTs†† Q - Queue table†† P - 1MT††
IV-4-22	TELEX, ops.	Dump associated memory C - Command table†† T - Terminal table†† E - Reentry table†† P - TELEX related PPUs††

HARDWARE REGISTER DUMP DIRECTIVES

IV-4-23	XP.	Dump executing exchange package(s)

†Directive generates 72-column output formatted for listing at an interactive terminal or 136-column output formatted for listing at a line printer.
††Indicates default list options.

4.5 PRINTER OUTPUT LISTING EXAMPLES

The listings illustrated in this section are samples of the line printer output listings produced by several of the input directives described in section 4.2.

Each page of output listing begins with two header lines: a title line and a subtitle line. The title lines are formatted as follows:

```
RA =     current input directive                              EXPRESS nn    DSDI - V2.0    yy/mm/dd. hh.mm.ss.    PAGE    n
0000000  dump type  hh.mm.ss. yy/mm/dd. CDC MULTI-MODE OPERATING SYSTEM    NOS 1                   comments
```

RA= 0000000	Indicates absolute addressing is in effect. If relative addresses were being dumped (RA or RAC directive entered), a nonzero reference address would appear in this field.
current input directive	Input directive currently being processed, including comments (50-character field)
EXPRESS nn	Reflects the EDD dump tape number currently being used (nn is a two-digit dump identifier assigned during EDD).
yy/mm/dd. hh.mm.ss.	This field reflects the date and time when DSDI was executed.
dump type	Indicates type of memory currently being dumped (CM, ECS, or a PPU number). If the AP directive is being processed, the PPU number is followed by the name of the program currently being dumped.
hh.mm.ss. yy/mm/dd. CDC MULTI-MODE OPERATING SYSTEM NOS 1	Time of day, date, system title line, and system version name taken from low core locations TIML and DTEL (80-character field)
comments	Up to 36 characters of comments specified on a *.ccc...ccc input directive

Example 1:

The following sample illustrates the printer output listing produced by the PP directive (dump PPU communication areas).

```
RA =     PP.        PPU COMMUNICATION AREAS.                  EXPRESS 02    DSDI - V2.0     75/05/05. 14.19.33.     PAGE      1
0000000  CM          16.06.39. 75/04/22. CDC MULTI-MODE OPERATING SYSTEM        NOS 1                1TD HUNG.

     PP00   MTR    1524 2201 0000 0000 0000   MTRA                 PP01   DSD    0423 0430 0000 0000 0000   DSDX
                   0000 0000 0000 0000 0000                                      0000 0000 0000 0000 0000
            CP01   3411 1766 0000 0004 0000   1IO     D                   CP30   3404 1430 0010 0044 0134   1DLX H 9A1
                   0000 0000 0000 0000 0000                               CH10   0000 0000 0000 0000 0000
                   0000 0000 0000 0000 0000                                      0125 2417 5755 0000 0000   AUTO.
                   0000 0000 0000 0000 0000                                      0000 0000 0000 0000 0000
                   0000 0000 0000 0000 0000                                      0000 0000 0000 0000 0000
                   0000 0000 0000 0000 0000                                      0000 0000 0000 0000 0000

     PP02   1MT    3415 2425 1313 0000 0001   1MTUKK    A          PP03   1TD    3424 0441 0000 0000 0000   1TD6
            RCHM   0012 0013 0000 0125 1220    J K  AUJP                         0000 0000 0000 0000 4000            5
            CP25   0000 0000 0101 0011 0003       AA I C                  CP01   0000 6531 0000 0003 0002      Y    C B
                   0000 0000 0000 4407 0001          9G A                 CH03   0317 1420 3523 3020 3421   COLP2SXP1Q
                   4747 4747 4747 4747 4747   **********                  CH11   3031 2100 0000 6010 3013   XYQ    HXK
                   4747 0000 0000 0000 0000   **                                 1203 3301 0554 0100 1211   JCOAE=A JI
                   0000 0000 0000 0000 0000                                      0100 5726 3031 2100 0000   A .VXYQ
                   0000 0000 0000 0000 0000                                      6010 3014 3407 0200 5010    HXL1GB /H

     PP04   1LS    3414 2327 5600 0000 0000   1LSW,                PP05   1SP    3423 2030 0000 0000 0000   1SPX
            DPPM   0044 0540 0000 0126 0375    9E5  AVC                   EATM   0032 0002 7700 0001 0000    Z B    A
            CP27   0020 0052 5024 0000 0010    P )/T   H                  CP30   0000 0000 0000 0000 0000
                   0010 0076 0022 0000 0000    H   R                             0000 0001 0001 4000 0000        A A5
                   0102 2350 1124 0515 1132   ABS/ITEMIZ                         3432 3004 0200 0446 0200   1ZXDB D-8
                   0555 0000 0000 0000 0000   E                                  1655 2411 1505 5555 0000   N TIME
                   5555 5555 5533 5733 3433         0.010                        0000 0000 0000 0000 0000
                   0000 0000 0000 0000 0000                                      0000 0000 0000 0000 0000

     PP06   1ED    3405 0427 0000 0000 0000   1EDW                 PP07   1LS    3414 2327 5600 0000 0000   1LSW,
                   0000 0000 0000 0000 0000                               DPPM   0044 0542 0000 0126 4100    9E7  AV6
            CP27   0035 3731 0001 3311 0000    24Y AOI                    CP27   0020 0052 5024 0000 0010    P )/T   H
            CH02   0000 0001 0001 4404 0074      A A9D                           0000 0000 0000 5374 0044           $  9
                   5555 5555 5533 5734 3543         0.128                        5700 0000 0000 0000 0000   .
                   5513 1416 2357 5555 0000    KLNS.                             0000 0000 0000 0000 0000
                   0000 0000 0000 0000 0000                                      5555 5555 5533 5733 3634         0.031
                   0000 0000 0000 0000 0000                                      0000 0000 0000 0000 0000

     PP10   1TA    3424 0101 0000 0125 0042   1TAA  AU 7           PP11   CIO    0311 1745 0000 0000 1640   CIO+    N5
                   0076 1101 3337 4203 0111     IA047CAI                  DCHM   0004 0001 0000 0000 0111    D A     AI
            CP01   0112 1101 3337 4203 0111   AJIA047CAI                  CP05   0000 0000 0201 0002 0025        BA B U
                   0000 6706 6706 0001 4005       F F A5E                 CH01   0000 0001 0001 5006 0042      A A/F 7
                   3442 5741 3341 2516 2423   17.606UNTS                         2355 0530 0503 2524 1117   S EXECUTIO
                   5755 0000 0000 0000 0000   .                                  1655 2411 1505 5555 0000   N TIME
                   5555 5555 5533 5733 3741         0.046                        0000 0000 0000 0000 0000
                   0000 0000 0504 2620 0323        EDVPCS                        0000 0000 0000 0000 0000
```

Channel Assignment
Control Point Assignment
Monitor Function
Program in Execution

Example 2:

The following sample illustrates the printer output listing produced by the AP directive (dump analysis of PPU and PPU memory in line format).

```
RA =     AP,1TD.   DUMP 1TD.                                     EXPRESS 02      DSDI - V2.0      75/05/05. 14.20.06.       PAGE      2
0000000  PP03  1TD  16.06.39. 75/04/22. CDC MULTI-MODE OPERATING SYSTEM               NOS 1                  1TD HUNG.

         ANALYSIS OF PP03

      PP03    1TD    3424 0441 0000 0000 0000  1TD6                        LAST MAIN PROGRAM LOADED - PM
                     0000 0000 0000 0000 4000          5                   LAST OVERLAY LOADED      -
             CP01    0000 6531 0000 0003 0002     Y   C B                  LAST MASS STORAGE DRIVER - 6J¬
                     0317 1420 3523 3020 3421  COLP2SXP1Q
                     3031 2100 0000 6010 3013  XYQ    HXK
                     1203 3301 0554 0100 1211  JCOAE=A JI
                     0100 5726 3031 2100 0000  A .VXYQ
                     6010 3014 3407 0200 5010   HXL1GB /H

        RESIDENT ENTRY POINTS           LOW CORE CONSTANTS
           NAME  LOC CALLER          NAME LOC   ACTUAL EXPECTED

            FTN  0364  0000            RA  55    7522  0454  *** WARNING ***
            PRL  0424  0000            FL  56    0167  0121  *** WARNING ***
            RCH  0437  0000            ON  70    0001  0001
            DCH  0446  0000            HN  71    0100  0100
            DFM  0501  0000            TH  72    1000  1000
            EXR  0533  0000            TR  73    0003  0003
            SMS  0547  0000            CP  74    0200  0200
            POS  0606  3014            IA  75    6230  6230
            WDS  0612  0000            OA  76    6231  6231
            RDS  0616  3563            MA  77    6232  6232

           PPU MEMORY

0000  0003 2020 3336 ---- ---- ---- ---- 4101    CPP03         6A   0010  ---- 0547 ---- 0127 6542 4175 5404 7274      E*  AW 76 =D
0020  ---- ---- ---- ---- 1563 ---- ---- ----              M        0030  ---- ---- 1100 0400 ---- 0555 ---- ----        I D   E
0040  ---- ---- ---- ---- ---- ---- 6335 ----                 2     0050  2015 ---- ---- ---- 0073 7522 0167 6741     PM          RA  6
0060  ---- 6531 ---- 0003 0001 1176 0001 1526      Y    C AI   AMV  0070  0001 0100 1000 0003 0200 6230 6231 6232      AA H  CB   X Y Z
0100  ---- ---- ---- ---- ---- ---- ---- ----                       0110  ---- ---- ---- ---- ---- ---- ---- ----
0120  ---- ---- ---- ---- ---- ---- ---- ----                       0130  ---- 4101 ---- ---- ---- ---- ---- ----      6A
0140  ---- ---- ---- ---- ---- ---- ---- ----                       0150  ---- ---- ---- ---- ---- 0112 ---- ----               AJ
0160  ---- ---- ---- ---- ---- ---- ---- ----                       0170  ---- ---- ---- ---- ---- ---- ---- ----
--------------------------------------------------------------------------------------------------------------------------------------
0600  1276 C100 6775 2000 0106 6010 3014 3261   J A    P AF HXLZ    0610  0603 2101 ---- 3464 3263 0702 3563 3074   FCQA  1 Z GB2 X
0620  1620 6000 3001 0555 2104 5414 6000 3004   NP   XAE QD=L  XD   0630  0404 1701 0576 0363 3437 3003 3404 2004   DDOAE C 14XC1DPD
0640  5414 6200 3014 3261 0603 2101 ---- 3201   =L   XLZ FCQA  ZA   0650  0611 2000 0175 ---- 0576 2000 0106 6010   FIP A    E P AF H
0660  0361 3014 3461 1066 1017 3313 1003 2200   C XL1 H HOOKHCR     0670  7777 5200 0600 0404 5500 0600 1401 3462     ) F DD   F LA1
0700  1402 5400 1566 3077 6260 2000 0100 5400   LB= M X    P A =    0710  0777 1403 3454 2000 7462 3455 5137 7727   G LC1=P    1 (4 W
0720  5400 1251 5037 7720 2100 6500 6010 2000   = J(/4 PQ     HP    0730  1146 3401 0303 4502 3601 4001 3402 3011   I-1ACC+B3A5A1BXI
0740  1703 0571 3014 1237 0403 0100 1061 3447   OCE XLJ4DCA H 1*    0750  5037 7726 5500 0777 1701 3456 3014 2200   /4 V   G OA1,XLR
0760  7000 3403 1101 7603 6503 0773 6503 0773     1CIA C CG  CG     0770  0323 0100 6763 5037 7727 7403 7303 0130   CSA   /4 W C CAX
1000  0570 6603 1001 7543 3003 1103 7603 6503   E  CHA 8XCIC C C    1010  1016 6503 1016 7543 0100 6751 5037 7727   HN CHN 8A  (/4 W
1020  7403 7103 7462 0545 7543 0200 1254 2005    C C  E+ 8B J=PE    1030  3666 6010 3045 3213 1014 3146 3214 0603   3  HX+ZKHLY-ZLFC
1040  2100 C100 2177 7737 0705 1446 0200 7567   Q A Q  4GEL-B       1050  3660 3637 5037 7720 0403 0100 0713 0100   3 34/4 PDCA GKA
1060  C603 5037 7727 3447 6503 1127 3647 2000   FC/4 W1* CIW3*P     1070  1112 3456 2000 1145 3455 2000 0100 5400   IJ1,P I+1 P A =
1100  1247 2000 1107 5400 1250 1500 0305 6503   J*P IG= J/M CE C    1110  1127 2000 ---- 7203 6603 1114 3747 0410   IWP    C CIL4*DH
1120  1400 4456 7003 5400 1145 0100 1257 3047   L 9, C= I+A J.X*    1130  3554 1400 3447 2000 3655 5400 1247 2000   2=L 1*P 3 = J*P
1140  2300 5400 1250 0100 1027 ---- 0763 0764   S = J/A HW  6 G     1150  0766 0775 0776 1001 1003 1006 1007 1011   G G G HAHCHFHGHI
1160  1013 1020 1021 1024 1064 1107 1113 1114   HKHPHQHTH IGIKIL    1170  1122 0740 1700 5000 1172 3424 3054 1003   IRG5O / I 1TX=HC
1200  2105 5317 6220 0100 1247 4316 5000 1205   QESO PA J*8N/ JE    1210  3424 3054 1003 2105 5317 6220 1601 6225   1TX=HCQESO PNA U
1220  0327 ---- 5000 1221 3424 3054 1003 2105   CW  / JQ1TX=HCQE    1230  5317 6373 0020 0314 2030 5000 1234 3424   $O   PCLPX/ J11T
1240  3054 1003 2105 5317 6220 140C 4456 3655   X=HCQESO PL 9,3     1250  2300 7522 0503 0100 1027 3656 4055 1010   S  RECA HW3,5 HH
1260  0720 3654 1003 2105 5317 602C 3024 2177   GP3=HCQESO PXTQ     1270  6214 0455 0704 2177 7636 0714 0100 1667    LD GDQ  3GLA N
```

Example 3:

The following sample illustrates the printer output listing produced by the CP directive (dump active control point areas). The default list options (XTAF) are used to dump the TELEX control point area. This example continues for two pages. Also, observe that the columns cross page boundaries; that is, the left column is read continuously, from the top of the first page to the bottom of the second page. The sequence then continues at the top of the right column on the first page.

```
RA =     CP,TELEX. DUMP TELEX CONTROL POINT AND DAYFILE.   EXPRESS 02      DSDI - V2.0      75/05/05. 14.21.17.       PAGE      7
0000000  CM          16.06.39. 75/04/22. CDC MULTI-MODE OPERATING SYSTEM          NOS 1                  1TD HUNG.

0200 - CONTROL POINT 01

         CP01 EXCHANGE PACKAGE                        PARAMETER SUMMARY

 P       5422  A0  10263  B0      0                   JOB NAME             TELEX    CPU STATUS                      I
 RA     45400  A1      1  B1      1                   USER NUMBER                   SENSE SWITCHES                 65
 FL     12100  A2     61  B2      0                   PRIMARY FILE                  KCL R1                     000000
 EM     70CC0  A3   6266  B3      0                   ORIGIN TYPE           SYOT    KCL R2                     000000
 RAX        0  A4   6243  B4  11073                   TIME USED          0000004    KCL R3                     000000
 FLX        0  A5     56  B5     20                   TIME REMAINING     7777773    KCL EF                         00
 MA       200  A6      1  B6     40                   QUEUE PRIORITY        7775    CPU PRIORITY                  077
            0  A7    501  B7     21
                                                      MESSAGE 1                - ONSW 3.   97/401            J%
 X0  0000 C000 0000 0000 0000                         MESSAGE 2                -
 X1  0000 C000 0000 0000 0000
 X2  0000 0C00 0000 00C0 0023              S          CURRENT CONTROL STATEMENT - *TELEX.
 X3  0000 C000 0000 0000 6335              2
 X4  0C00 CC00 0000 0000 0000                         LAST DAYFILE MESSAGE     -  16.02.45.ONSW 3.
 X5  000C CC00 0000 0000 0000
 X6  2203 1400 0000 0000 0000 RCL                     SPECIAL ENTRY POINTS     -
 X7  000C C000 0000 0000 0000
-------------------------------------------------------------
       CCNTROL POINT AREA
                                                            |
CPU STATUS FLAGS       STSW 000220 2002              PB   D=AQ | JOB SEQUENCE NUMBER     RFCW 000274 00000000
ERROR FLAGS                             0000                   |                                            000000
ACTIVITY CCUNT                              0000               | DEM FILE RANDOM INDEX                                000000
RA/100                                          0454           | RESERVED                ALMW 000275 7777
FL/100                                              0121       | MAX MAGNETIC TAPES                         7
JOB NAME               JNMW 000221 24051405305555    TELEX     | MAX REMOVABLE PACKS                         7
JOB ORIGIN CODE                                   00           | MAX DEFER BATCH JOBS                         7
OPER ASSIGNED EQUIP    OAEW                         0000       | MAX LOCAL WORK FILES                          7
CPU PRIORITY           JCIW 000222 0077                        | MAX TIME LIMIT                                 77
QUEUE PRIORITY                          7775                   | MAX SRU LIMIT                                    77
CPU STATUS FOR ROLLOUT                      0                  | MAX FIELD LENGTH                                   77
                                             0                 | MAX ECS FIELD LENGTH                                 77
ROLLCUT IN PROCESS                            0                | MAX LINES PRINTED                                      77
ROLLOUT REQUESTED                              0               | MAX CARDS PUNCHED                                        77
                                                00             | RESERVED                ACLW 000276 7777
CPUS ALLOWABLE                                    000000       | DAYFILE MESSAGES COUNT                7777
CM RESID TIME LIMIT    TSCW 000223 00000245          B+A  DA=  | CONTROL STATEMENTS CNT                    7777
CPU TIME SLICE LIMIT                    010000040154           | DISPOSED OUTPUT COUNT                         77
RESERVED                    000224 00000000000000000000        | MS PRUS COUNT                                 777777
PP INPUT REGISTER      RLPW 000225 00000000000000000000        | ACCOUNT ACCESS WORD     AACW 000277 77777777777777777777
PUNCH/CHARACTER SET    SNSW 000226 0                           | LENGTH BUFFER 0         ICAW 000300 0110                 AHAIB
                                    0000000                    | ADDRESS BUFFER 0                      011102
RESERVED-INSTALLATION                   0000                   | LENGTH BUFFER 1                            0000
PP PAUSE FLAGS                              0000               | ADDRESS BUFFER 1                               000000
SENSE SWITCHES                                  65             | SP ENTRY POINT FLAGS    SEPW 000301 0000
                                                  00           | RESERVED                              0000
RESERVED                    000227 00000000000000000000        | DMP= PARAMETER VALUE                      000000
MESSAGE 1 AREA         MS1W 000230 17162327553657000000 ONSW 3.| SSJ= PARAMETER ADDRESS                          000000
                            000231 44425037333455550000 97/401 | ENTRY POINT NAME        SPCW 000302 000000
                            000232 00000000000000006263        | SYSTEM PROCESSOR FLAGS                 00
                            000233 00000C00000000000000        | PARAMETER                                 000000
                            000234 00000000000000000000        | PARAMETER LIST ADDRESS                          000000
```

X List Option

Beginning of T List Option

Page Boundary

```
RA =     CP,TELEX. DUMP TELEX CONTROL POINT AND DAYFILE.   EXPRESS 02    DSDI - V2.0     75/05/05. 14.21.17.      PAGE     8
0000000  CM         16.06.39. 75/04/22. CDC MULTI-MODE OPERATING SYSTEM       NOS 1             1TD HUNG.

MESSAGE 2 AREA         MS2W 000235 00000000000000000000          | KCL - EF              JCRW 000303 00
                            000236 00000000000000000000          | KCL - R3                             000000
                            000237 00000C00000000000000          | KCL - R2                                  000000
INSTALLATICN WORD 0    IN0W 000240 00000000000000000000          | KCL - R1                                       000000
INSTALLATICN WORD 1    IN1W 000241 00C00000000000000000          |                       DBAW 000304 00
INSTALLATICN WORD 2    IN2W 000242 00000000000000000000          | INPUT BUFFER ADDRESS                 000000
INSTALLATION WORD 3    IN3W 000243 00000000000000000000          | RIGHT SCREEN BFR ADD                      000000
INSTALLATION WORD 4    IN4W 000244 00000000000000000000          | LEFT SCREEN BFR ADD                            000000
INSTALLATION WORD 5    IN5W 000245 00000000000000000000          | MAP OPTIONS           LB1W 000305 0000
INSTALLATICN WORD 6    IN6W 000246 00000000000000000000          |                                        00000000
INSTALLATION WORD 7    IN7W 000247 00000000000000000000          | GLOBAL LIB INDICATORS                          00000000
SRU LIMIT EXCEED FLAG  SRUW 000250 0000                          | SECOND LIB/GLOBAL IND  LB2W 000306 00000000000000000000
SRU ACCUMULATOR                        0000000000000000          | FIRST LIB/GLOBAL IND   LB3W 000307 00000000000000000000
CP LIMIT EXCEED FLAG   CPTW 000251 0000                     ABB  |                             000310 00000000000000000000
RESERVED                           0000                          |                             000311 00000000000000000000
CP ACCUM (MILLIUNITS)                     000000010202           |                             000312 00000000000000000000
IO ACCUMULATORS        IOAW 000252 00004220000000000010   7P   H |                             000313 00000000000000000000
M13 = M1 * M3          MP1W 000253 000000                        |                             000314 00000000000000000000
M14 = M1 * M4                            000000                  |                             000315 00000000000000000000
ADDER ACCUMULATOR                              00000000          |                             000316 00000000000000000000
M1 * 1000              MP2W 000254 000000                        |                             000317 0000000000000000000U
M12 = M1 * M2                            000000                  |                             000320 00000000000000000000
SRU LIMIT (UNITS/10)                           0000              |                             000321 00000000000000000000
CP LIMIT (UNITS/10)                                0000          |                             000322 00C00000000000000000
CPM                    MP3W 000255 00C0000000                    |                             000323 00000000000000000000
IOM                                          0000000000          |                             000324 00000000000000000000
SRU LIMIT (UNITS)      SRJW 000256 0000                          |                             000325 00000000000000000000
SRU ACCUMULATOR                        0000000000000000          |                             000326 00000000000000000000
CP LIMIT (UNITS)       CPJW 000257 00000000                      |                             000327 00000000000000000000
CP ACCUM (MILLIUNITS)                     000000000000           | CONTROL STATEMENT BFR  CSBW 000330 24051405305700000000 TELEX.
SRU LIMIT (UNITS)      SRLW 000260 0000                          |                             000331 24051405303557550000 TELEX2.
SRU LIMIT                              0000000000000000          |                             000332 05301124575500000000 EXIT.
CP LIMIT (UNITS)       CTLW 000261 77777777                      |                             000333 24051405303557550000 TELEX2.
CP LIMIT                                  770000000000           |                             000334 00000000000000000000
JOB CARD FL            FLCW 000262 0500                E E       |                             000335 00000000000000000000
LAST CARD FL                           0500                      |                             000336 00000000000000000000
FL FCR DMP= CALL                          0000                   |                             000337 00000000000000000000
ROLLIN FL                                     0000               |                             000340 00000000000000000000
FL INCREASE REQUEST                               0000           |                             000341 00000000000000000000
JOB CARD ECS FL        ELCW 000263 0000                          |                             000342 00000000000000000000
LAST CARD ECS FL                       0000                      |                             000343 00000000000000000000
                                          0000                   |                             000344 00000000000000000000
ROLLIN ECS FL                                 0000               |                             000345 00000000000000000000
ECS FL INCREASE REQ                               0000           |                             000346 00000000000000000000
ECS FIRST TRACK        ECSW 000264 0000                          |                             000347 00000000000000000000
ECS TRACKS/2000                        0000                      |                             000350 00000000000000000000
                                          0000                   |                             000351 00000000000000000000
ECS/CPUMTR CONTROL                            00000000           |                             000352 00000000000000000000
TXOT SUBSYSTEM         TXSW 000265 00                            |                             000353 00000000000000000000
RESERVED                             000000                      |                             000354 00000000000000000000
TERM INTERRUPT ADDRESS TIAW               000000                 |                             000355 00000000000000000000
OUTPUT POINTER         TIOW                     000000           |                             000356 00000000000000000000
AUXILIARY PACKNAME     PFCW 000266 0000000000000          D      |                             000357 00000000000000000000
EST OF FAMILY DEVICE          |                 04               |                             000360 00000000000000000000
LIMIT FOR SIZE OF DAF         |                   0              |                             000361 00000000000000000000
LIMIT FOR NUMBER OF PF        |                    0             |                             000362 00000000000000000000
LIMIT CUMM SIZE IAPF          v                     0            |                             000363 00000000000000000000
```

Page Boundary

```
RA =     CP,TELEX. DUMP TELEX CONTROL POINT AND DAYFILE.   EXPRESS 02    DSDI - V2.0     75/05/05. 14.21.17.      PAGE     9
0000C00  CM         16.06.39. 75/04/22. CDC MULTI-MODE OPERATING SYSTEM       NOS 1             1TD HUNG.

LIMIT FOR SIZE OF IAPF                              0            |                             000364 00000000000000000000
USER NUMBER            UIDW 000267 0000000000000                 |                             000365 00000000000000000000
USER INDEX                                      000000           |                             000366 00000000000000000000
NO EXIT FLAG           EECW 000270 0                             |                             000367 00000000000000000000
                                     000                         |                             000370 00000000000000000000
REPRIEVE ERROR OPTIONS                 0000                      |                             000371 00000000000000000000
TERMINAL INPUT POINTER TINW               000000                 |                             000372 00000000000000000000
ERROR RETURN ADDRESS                            000000           |                             000373 00000000000000000000
INPUT FILE FST         TFSW 000271 6605                E         |                             000374 00000000000000000000
PRIMARY FILE FST                       0000                      |                             000375 00000000000000000000
RESERVED                                  00                     |                             000376 00000000000000000000
EVENT DESCRIPTOR       TERW                 000000               |                             000377 00000000000000000000
ROLLOUT TIME PERIOD                               0000           |
RESERVED               CSPW 000272 0000                  5  AAYA1 |
EOR FLAG/CS COUNT                      40000001                  |
NEXT STATEMENT INDEX                          0131               |
LIMIT INDEX                                       0134           |
INPUT/SKIP FLAGS       CSSW 000273 0                             |
                                    0                            |
EQUIPMENT NUMBER                     00                          |
FIRST TRACK                            0000                      |
CURRENT TRACK                             0000                   |
CURRENT SECTOR                                0000               |
FIRST/SECOND HALF FLAG                            0000           |
```

T option resumes at top of right column on first page in sequence.

```
        DAYFILE POINTERS AND BUFFER <—— A List Option

0006322 0001 7210 0016 0100 0000  A H NA
0006323 000C 0000 0000 0000 0000

0017210 55344157333457403557 24051405305700000000 55344157333457403657 55160524271722135503  16.01.52.TELEX.     16.01.53. NETWORK C
0017214 01220423551107161722 05045700000000000000 55344157333557344057 17162327554057000000 ARDS IGNORED.       16.02.15.ONSW 5.
0017220 55344157333557364157 17162327553457000000 55344157333557373557 17162327554157000000  16.02.36.ONSW 1.   16.02.42.ONSW 6.
0017224 55344157333557374057 17162327553657000000 55344057374057353457 01242401031051202217  16.02.45.ONSW 3.   15.45.21.ATTACH(PRO
0017230 04250324502016541101 05200103135616015255 00000000000000000000 55344157333457344257 DUCT/PN=IAEPACK,NA)            16.01.17.
0017234 07242251202217042503 24560624165217261450 06241646062416353755 00000000000000000000 GTR(PRODUCT,FTN)OVL/FTN-FTN24
0017240 55344157333457353457 55050411241116075503 17152014052405570000 55344157333457353557  16.01.21. EDITING COMPLETE.   16.01.22.
0017244 07242251202217042503 24560617222422011656 04522514110250061722 24220116000000000000 GTR(PRODUCT,FORTRAN,D)ULIB/FORTRAN
0017250 55344157333457364457 55050411241116075503 17152014052405570000 55344157333457364457  16.01.39. EDITING COMPLETE.   16.01.39.
0017254 07242251202217042503 24561317021714521726 14500317021714460317 20310314000000000000 GTR(PRODUCT,KOBOL)OVL/COBOL-COPYCL
0017260 55344157333457374457 55050411241116075503 17152014052405570000 55344157333457374457  16.01.49. EDITING COMPLETE.   16.01.49.
0017264 07242251202217042503 24560317021714560452 25141102500317021714 00000000000000000000 GTR(PRODUCT,COBOL,D)ULIB/COBOL
0017270 000C0000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
--------------------------------------------------------------------------------------------------------------------------------------

        ATTACHED FILES <—— F List Option

   0002  SALVARE  CM     FNT - 6604  2301 1426 0122 0500 0701 SALVARE GA   EST - 6500  6000 0000 0000 0411 0760       DIG
                         FST - 6605  0000 4412 4412 0003 0315   9J9J CCM
                         TRACK CHAIN -
                         4412 0003
```

Example 4:

The following sample illustrates the printer output listing produced by the MST directive (dump mass storage/track reservation table). The MST is listed in two columns. The left column is read from top to bottom, perhaps across page boundaries, and continues at the top of the right column. The track reservation table is listed in single column following the MST.

```
RA =     MST.      MASS STORAGE TABLES.                   EXPRESS 02     DSDI - V2.0      75/05/05. 14.24.09.       PAGE    10
0000000  CM         16.06.39. 75/04/22. CDC MULTI-MODE OPERATING SYSTEM          NOS 1                 1TD HUNG.

EQUIPMENT 00 - MASS STORAGE TABLE

NUMBER AVAILABLE PRUS  TDGL 007600 00043416                    D1NFX5 8)
LENGTH OF TRT                              0630
FIRST AVAIL TRACK PTR                          4073
NUM AVAILABLE TRACKS                               0252
ECS ADDRESS MST/TRT    ACGL 007601 00000000
DIRECT ACCESS FILE CNT                     0000
CURRENT USER COUNT DAF                         0000
MMF INTERLOCKS                                     0000
FIRST TRACK IAPF       ALGL 007602 4351                        8(5 8) P
LABEL TRACK                            4000
FIRST TRACK PERMISSION                     4352
NUMBER CATALOG TRACKS                          0020
FIRST TRACK IQFI                                   0000
FAMILY OR PACK NAME    PFGL 007603 23312324051500              SYSTEM A
DEVICE NUMBER                                    01
RESERVED                                           00
REL UNIT MULTIUNIT DEV                               0
NUM UNIT MULTIUNIT DEV                                0
USER NUM PRIVATE PACK  PUGL 007604 00000000000000                     0
DEVICE MASKS                                     177777
DEVICE FLAGS           MDGL 007605 0                                AS AS AS
DEVICE STAT TABLE PTR                  000
REL RESER IF CHAN REL                    0
SINGLE UNIT SEC LIMIT                     153
MAXIMUM SECTOR LIMIT                       000153
MINIMUM SECTOR LIMIT                             000153
RESERVED               R1GL 007606 00000000000000000000
RESERVED               R2GL 007607 00000000000000000000

GLOBAL INSTAL AREA       ISGL 007610 00000000000000000000
                         I2GL 007611 00000000000000000000
808 POS/844 CHANNEL      DSLL 007612 0000
844 EQUIPMENT                            0000
844 TRACK                                    0000
844 SECTOR                                       0000
844 DEVICE STATUS                                    0000
FIRST CHAN RESERVES      DILL 007613 000
RESERVED                                0
SECOND CHAN RESERVES                     000
RESERVED                                    0000000000000
DAYFILE TRACK            DULL 007614 6726                           V U L65
ACCOUNT FILE TRACK                       6725
ERRLOG TRACK                                 6714
SYSTEM TABLE TRACK                               4140
MTR TRACK                                            0000
DEVICE STATUS FLAGS      STLL 007615 2002                         PB  AA C A
MACHINE ID                               0
RESERVED                                  0
ERROR STATUS                               00
RESERVED                                     0101
CURRENT USER COUNT DAF                           0003
NEXT EQUIPMENT                                       00
ORIGINAL NUMBER UNITS                                  0
FLAGS                                                   1
RESERVED                 R3LL 007616 00000000000000000000
LOCAL INSTAL AREA        ISLL 007617 00000000000000000000

          TRACK RESERVATION TABLE

007620 +0000  4144 4440 4502 5016 4017  1--- ---- 1111  6995+B/N50
007621 +00C4  4005 4006 4007 4010 0017  ---- ---- 1111  5E5F5G5H 0
007622 +001C  4011 4012 4013 4014 0017  ---- ---- 1111  5I5J5K5L 0
007623 +0014  4015 4016 4017 4020 0017  ---- ---- 1111  5M5N5O5P 0
007624 +0020  4021 4022 4023 4024 0017  ---- ---- 1111  5Q5R5S5T 0
007625 +0024  4025 4026 4027 4030 0017  ---- ---- 1111  5U5V5W5X 0
0C7626 +0030  4031 4032 4033 4034 0017  ---- ---- 1111  5Y5Z5051 0
0C7627 +0034  4035 4036 4037 4040 C017  ---- ---- 1111  52535455 0
007630 +0040  4041 4042 4043 4044 0017  ---- ---- 1111  56575859 0
007631 +0044  4045 4046 4047 4050 0017  ---- ---- 1111  5+5-5*5/ 0
007632 +0050  4051 4052 4053 4054 C017  ---- ---- 1111  5(5)5$5= 0
007633 +0054  4055 4056 4057 4060 0017  ---- ---- 1111  5 5,5.5  0
007634 +0060  4061 4062 4063 4064 0017  ---- ---- 1111  5 5 5 5  0
007635 +0064  4065 4066 4067 4070 0017  ---- ---- 1111  5 5 5 5  0
007636 +0070  4071 4072 4073 4074 0017  ---- ---- 1111  5 5 5 5  0
007637 +0074  4075 4076 4077 4100 0017  ---- ---- 1111  5 5 5 6  0
007640 +01CC  4101 4102 4103 4104 0017  ---- ---- 1111  6A6B6C6D 0
007641 +01C4  4105 4106 4107 4110 0017  ---- ---- 1111  6E6F6G6H 0
007642 +0110  4111 4112 4113 4114 0017  ---- ---- 1111  6I6J6K6L 0
007643 +0114  4115 4116 4117 4120 0017  ---- ---- 1111  6M6N6O6P 0
007644 +0120  4121 4122 4123 4124 0017  ---- ---- 1111  6Q6R6S6T 0
007645 +0124  4125 4126 4127 4130 0017  ---- ---- 1111  6U6V6W6X 0
007646 +0130  4131 4132 4133 4134 0017  ---- ---- 1111  6Y6Z6061 0
```

Track Link Byte Ordinal

Status Bits

Example 5:

The following sample illustrates the printer output listing produced by the C, D, and E memory dump directives (instruction parcel, byte, and word format, respectively). The same portions of central memory are dumped in each format. Auto page eject has been disabled via the EJOFF directive to allow listing the output from all three memory dump directives on one page.

```
RA =     C,6200,6240.    C - FORMAT DUMP.                    EXPRESS 00      OSDI - V2.0      75/05/15. 16.24.54.       PAGE      1
0000000  CM          09.44.58. 75/05/13. CDC MULTI-MODE OPERATING SYSTEM          NOS 1                   IMS HUNG - MST INTERLOCKED.

         CM          C,6200,6240.    C - FORMAT DUMP.

0006200 15242 22400 00000 00000 MTRT          ----------------------------------------  0006230 34050 42400 00000 00000 1EDT
0006201 00000 00000 00000 00000               0006220 20061 54410 11000 00154 PFM9HI  A=  0006231 00000 00000 00000 00000
0006202 34111 76600 00000 10000 1IO     A     0006221 00560 00000 00000 00110  ,       AH  0006232 20343 55600 64325 50000 P12,  Z
0006203 00000 00000 00000 00000               0006222 20000 00000 00003 30001 P      0 A  0006233 00000 00000 00433 40117       81AO
--------------------------------------        0006223 00000 00000 00447 20011        9 I  0006234 03171 52014 05240 55700 COMPLETE.
0006210 04230 43000 00000 00000 DSDX          0006224 34323 00402 00044 60200 1ZXDB D-B   0006235 00000 00000 00000 00000
0006211 00000 00000 00000 00000               0006225 04170 31725 24520 00000 DOCOUT)     ----------------------------------------
0006212 34041 43000 10004 40101 1DLX H 9AA    0006226 00000 00000 00000 00000
0006213 00000 00000 00000 00000               ----------------------------------------

         CM          D,6200,6240.    D - FORMAT DUMP.

0006200 1524 2224 0000 0000 0000 MTRT          ----------------------------------------  0006230 3405 0424 0000 0000 0000 1EDT
0006201 0000 0000 0000 0000 0000               0006220 2006 1544 1011 0000 0154 PFM9HI  A=  0006231 0000 0000 0000 0000 0000
0006202 3411 1766 0000 0001 0000 1IO     A     0006221 0056 0000 0000 0000 0110  ,       AH  0006232 2034 3556 0064 3255 0000 P12,  Z
0006203 0000 0000 0000 0000 0000               0006222 2000 0000 0000 0033 0001 P      0 A  0006233 0000 0000 0000 4334 0117       81AO
----------------------------------------       0006223 0000 0000 0000 4472 0011        9 I  0006234 0317 1520 1405 2405 5700 COMPLETE.
0006210 0423 0430 0000 0000 0000 DSDX          0006224 3432 3004 0200 0446 0200 1ZXDB D-B   0006235 0000 0000 0000 0000 0000
0006211 0000 0000 0000 0000 0000               0006225 0417 0317 2524 5200 0000 DOCOUT)     ----------------------------------------
0006212 3404 1430 0010 0044 0101 1DLX H 9AA    0006226 0000 0000 0000 0000 0000
0006213 0000 0000 0000 0000 0000               ----------------------------------------

         CM          E,6200,6240.    E - FORMAT DUMP.

0006200 15242224000000000000 00000000000000000000 34111766000000010000 00000000000000000000 MTRT                1IO    A
0006204 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
0006210 04230430000000000000 00000000000000000000 34041430001000440101 00000000000000000000 DSDX                1DLX H 9AA
0006214 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
0006220 20061544101100000154 00560000000000000110 20000000000000330001 00000000000044720011 PFM9HI  A= ,    AHP      0 A      9  I
0006224 34323004020004460200 04170317252452000000 00000000000000000000 00000000000000000000 1ZXDB D-B DOCOUT)
0006230 34050424000000000000 00000000000000000000 20343556006432550000 00000000000043340117 1EDT               P12,  Z          81AO
0006234 03171520140524055700 00000000000000000000 00000000000000000000 00000000000000000000 COMPLETE.
```

4.6 DIAGNOSTICS

4.6.1 DAYFILE MESSAGES

CM RECORD NOT FOUND.

The central memory record was not found in the EDD dump file.

DIRECTORY TABLE BAD.

An EOR or EOF was encountered while the random file directory which was created by the D option was being read.

DSDI ARGUMENT ERROR.

An unknown keyword was encountered on the DSDI control statement.

DSDI ERROR LIMIT EXCEEDED.

More than 50 errors were detected.

EXPRESS DUMP COMPLETE (FL USED xxxxxxB)

The dump was completed normally. The amount of field length used was xxxxxx octal words.

4.6.2 ERROR MESSAGES

ALTERNATE FILE ACTIVE

The alternate file was already being processed when entry of the OUTPUT directive was attempted.

ALTERNATE OUTPUT TO TERMINAL ILLEGAL

File name OUTPUT was specified on OUTPUT directive entered from a terminal. Alternate list output cannot be assigned to the terminal.

BAD SYSTEM POINTER

A bad system pointer was detected in the EDD dump file during processing of an input directive.

DIRECTIVE PARAMETER ERROR

An error was detected in a directive parameter.

DIRECTIVE RESTRICTED TO PRINTER OUTPUT

Directive entered produces output which cannot be listed at a terminal. Refer to description of OUTPUT directive in section 4.3.

ECS RECORD NOT FOUND

The ECS record was not found in the EDD dump file.

HARDWARE REGISTERS NOT FOUND

The hardware register's record was not found in the EDD dump file.

ILLEGAL CONTROL POINT NUMBER

Either the control point number specified was greater than the number in the EDD dump file, or the job name was not found.

ILLEGAL FILE NAME

An illegal file name was specified.

ILLEGAL PRINT DENSITY SELECTION

A print density other than 3, 4, 6, or 8 lines per inch was specified.

ILLEGAL PRINT OPTION SELECTION

An illegal list option was specified in a directive.

PPU NOT FOUND

The requested PPU record was not found in the EDD dump file.

RECORD NOT FOUND

The record name specified in a READ directive was not found in the specified file.

REQUESTED MEMORY NOT FOUND

The EPB directive was entered and no ECS/PP buffer was found in the EDD dump file.

STATUS/CONTROL REGISTERS NOT FOUND

The S/C register's record was not found in the EDD dump file.

SUBSYSTEM NOT FOUND

Requested subsystem was not found in the EDD dump file.

UNRECOGNIZABLE DIRECTIVE

Directive entered was not a valid DSDI input directive.

WORD ADDRESS NOT FOUND

A word address requested was not found in the specified record in the EDD dump file.

TIME-SHARING STIMULATOR 5

The purpose of the time-sharing stimulator software package is to enter a hypothetical load of time-sharing jobs into the system in order to analyze the effects of such a load on response time and system reliability.

The procedure to use the time-sharing stimulator package includes:

1. Creating the hypothetical load of time-sharing jobs (called a session file); refer to section 5.1.

2. Artificially processing (stimulating) these jobs using stimulator commands at the system console; refer to section 5.2.

Multiple stimulator sessions are described in section 5.3; errors that can occur while using the stimulator software are described in section 5.6.

5.1 SESSION FILE CONSTRUCTION

A session file is an indirect access permanent file that contains the hypothetical time-sharing job load that is to be analyzed by the stimulator. A session file consists of a group of records; each record is a session and contains a hypothetical time-sharing job that is composed of various entries. Note the following requirements.

1. Begin each session with the entry ANSWERBAC; this is a terminal type identifier to the Time-Sharing Executive.

2. Enter each session entry on a separate line or card.

3. After ANSWERBAC, enter a user number entry and a password entry.

4. Specify the time-sharing terminal commands and data.

5. End each session with the command BYE so that the login of the next session is performed correctly.

Figure IV-5-1 contains an example of a session file.

5.2 SESSION FILE PROCESSING

Use the stimulator software to process the time-sharing session file. The following procedure is used.

1. There must be a Time-Sharing Subsystem Stimulator entry in the EST as described in part II, section 4.3.7.

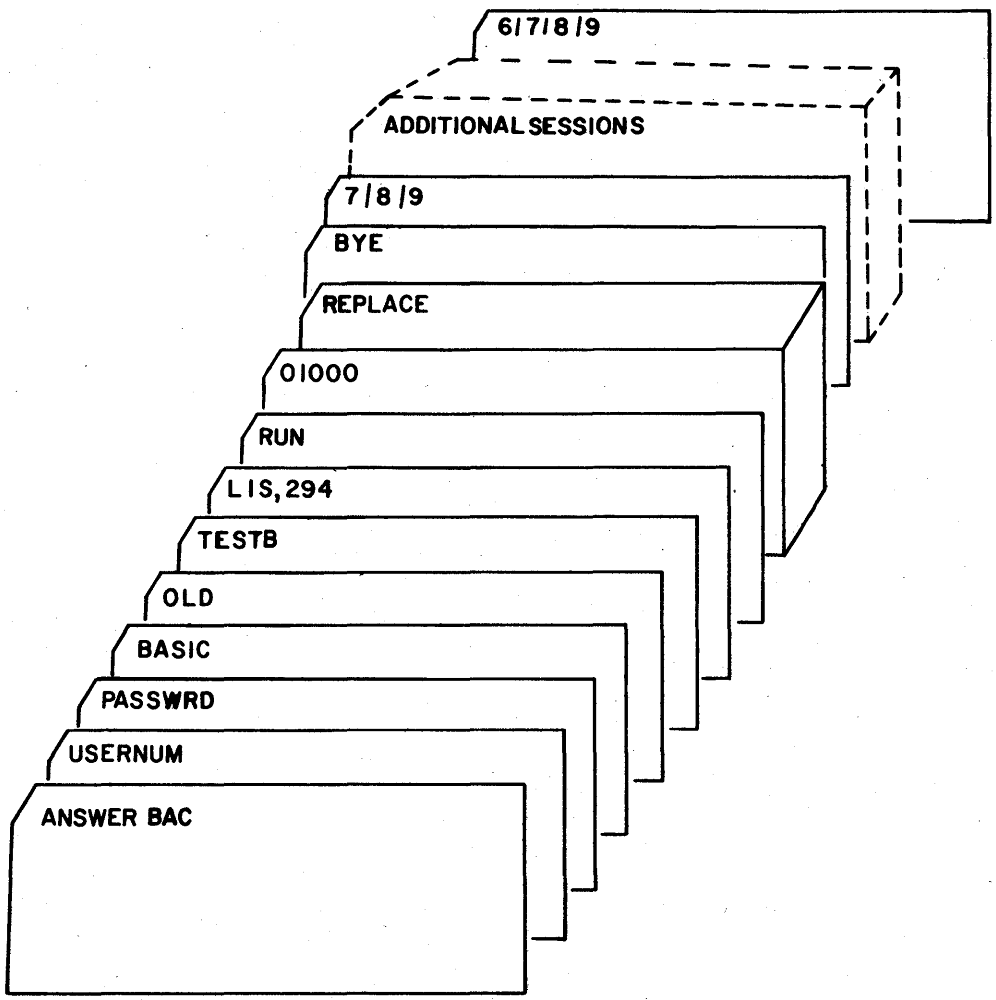


Figure IV-5-1. Sample Session File

2. Use DSD to bring the stimulator software to the last available control point.

   Type STIMULATOR
   Press CR

3. The stimulator requests the K display at the control point.

4. Type K, nn. — nn is the number of the last available control point.
   Press CR

5. The following information appears on the left display.

```
ENTER STIMULATION FILE PARAMETERS IN FORM -
   P = VALUE
   F =          SESSION FILE NAME
   UI = 0B      USER INDEX (OCTAL BASE ASSUMED)
   N = 1        NUMBER OF FILE ON SESSION FILE
   GO           BEGIN PROCESSING OF SESSION FILE
```

6. Type the session file parameters individually as indicated in the following format column or as a group on the same line in the format.

   K. F=lfn, UI=number, N=x, GO.

Format	Default	Significance
K. F=lfn	None	lfn is the indirect permanent file name of the session file.
K. UI=number	0	number is the user index of file lfn. The default family is assumed.
K. N=x	1	x is the number of files to skip on file lfn before data is read from the session file; range is 1 through 377777_8.
K. GO	None	Indicates to the stimulator that all the parameters have been entered.

7. The following information appears on the left display.

```
ENTER STIMULATION PARAMETERS IN FORM -

  P = VALUE        (NUMERIC VALUES ARE ASSUMED
                   DECIMAL BASE. ) (SEE RIGHT SCREEN
                   FOR SPECIAL PARAMETERS. )

NT = #             NUMBER OF TERMINALS TO STIMULATE
                   (3-#)†

LS = 30            LINE SPEED IN CHARACTERS/SECOND
                   (1-1000)

TT = 10            THINK TIME DELAY IN SECONDS (0-127)

TI = 8             UPPER BOUND OF RANDOM THINK TIME
                   INCREMENT 0-64 (MUST BE A POWER OF 2).
                   THINK TIME WILL VARY BETWEEN TT AND
                   TT+TI.

AC = #             ACTIVATION COUNT (1-NT)
                   (NUMBER OF TERMINALS TO ACTIVATE
                   EVERY AD SECONDS)

AD = 0             ACTIVATION DELAY IN SECONDS (0-127)

RC = 0             NUMBER OF TIMES TO REPEAT STIMULATION

RO = OFF           OUTPUT RECOVERY (ON OR OFF)

LF = OFF           LOOP ON SESSION RECORD (ON OR OFF)
     ON            LOOP ON SESSION RECORD
     OFF           LOOP ON SESSION FILE
```

8. To display the special parameters:

   Type KK
   Press CR

9. The following special parameters display appears on the right screen.

```
SPECIAL PARAMETERS.

OPTION LS MAY HAVE PARAMETERS OF THE FORMS
LS=MIXED (T1/LS1,T2/LS2,...,TN/LSN)
WHERE LINE SPEEDS *LSN* ARE ASSIGNED TO *TN* NUMBER OF TERMINALS.
```

10. Type the stimulator parameters individually as indicated in the following format column or as a group on the same line in the format.

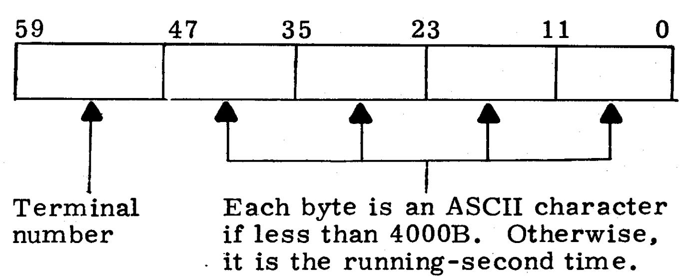
    K. NT=x, LS=y, TT=z, TI=a, AC=b, AD=c, RC=d, RO=ON, GO.

    Enter any special parameters (display in step 9) before the GO. entry.

---

† # is the number of terminals designated in the EST.

Format	Default	Significance
K.NT=#	Current EST value	# is the current maximum number of terminals on which the Time-Sharing Subsystem is initialized. It must be less than or equal to the number specified in the EST entry; the range is 3 through 1000_8.
K.LS=y	30 characters/ second	y is the line speed at which the operator wishes to run the terminals; the range is 1 through 1000 characters per second. Unless the assembled in time in 1TD per scan of multiplexers is changed, any value above 60 characters per second causes a line speed of approximately 60 characters per second. If many terminals (600_8, for example) are run, the maximum realizable line speed is determined by the overhead of 1TD and 1TS. The worst case can be determined by checking the 1TD cycle time message, which is issued when TELEX is dropped.
K.TT=z	10 seconds	z is the number of seconds that each line is delayed before it is sent to the Time-Sharing Subsystem; 0 through 127 seconds.
K.TI=a	8	a must be less than or equal to 64 and must be a power of 2. If a is 0, the think time will vary between the values of z (TT) and z plus a.
K.AC=b	Current EST value	b is the number of terminals that are to be logged in every c seconds (refer to K.AD=c). The AC and AD parameters provide a method to stagger the I/O commands and data to the Time-Sharing Subsystem. The range is 1-# (value of NT).
K.AD=c	0 seconds	c is the delay time until the next login is allowed; range is 0 through 127 seconds.
K.RC=d	0	d is the number of times the sessions will be repeated for each terminal; range is 0 through 31.
K.RO=ON	OFF	When RO=ON, the output from the Time-Sharing Subsystem is recovered on file STIMOUT. The format of this file is:



When RO=OFF, the output is not recovered.

K.LF=OFF ON

LF=OFF causes the stimulator to loop through the session file. The RC parameter must be nonzero in order for the LF parameter to have an effect on the session processing. If LF=ON, the session record is repeated the number of times specified by the RC parameter.

K.GO

Indicates to the stimulator that the parameters are entered

The special parameter format is:

K.LS=MIXED(t_1/x, t_2/y, . . . , tn/n)

ti is equal to the number of terminals assigned the designated line speed. Range is 1 through 1000 characters per second.

If the special line speed parameters are not entered, the line speed for each terminal is taken from the LS parameter.

5.3 MULTIPLE SESSIONS

Each session on the session file is allocated to each terminal line. When each session has been allocated, the stimulator goes to the first session and reallocates the sessions to the next group of terminals. This procedure is repeated until all the terminals are allocated to a session. If a repeat count is not designated (RC parameter), the line is shut off when the session is finished. When the session completes, the repeat count is checked.

The repeat count corresponds to the number of times the first session is executed. Therefore, the current session is completed and then the next, until the last session is executed. The repeat count is decremented, and the first session is executed. If the repeat count is zero at this point, the line is shut off. If not, it will continue through the session until the first one is encountered again.

5.4 STIMULATOR PARAMETER OUTPUT FORMAT

The following parameter information is placed on the output file for printing upon completion of the stimulator run.

```
STIMULATOR PARAMETERS-
SESSION FILE NAME-            F=lfn
USER INDEX-                   UI=number
NUMBER OF THE FILE-           N=x
NUMBER OF TERMINALS-          NT=#
LINE SPEED-                   LS=(1-1000)
THINK TIME-                   TT=(0-127)
RANDOM THINK TIME-            TI=(0-64)
ACTIVATION COUNT-             AC=(1-#)
ACTIVATED DELAY-              AD=(0-127)
REPEAT COUNT-                 RC=(0-31)
OUTPUT RECOVERY-              RO=(ON or OFF)
LOOP ON SESSION-              LF=(ON or OFF)
```

These parameters are explained in section 5.2.

5.5 POSTPROCESSING

The postprocessing of the time-sharing stimulator output is accomplished by reading the file STIMOUT and sorting the data according to terminal number. This is accomplished by using the DEMUX control statement. The data for each terminal is separated by a page eject and an end of line indicator. The terminal data appears as it would on a terminal page. If the ASCII character on the STIMOUT file is 4000_8 or greater, the lower 11 bits are assumed to be time of a carriage return or the first character of output.

An example of output from DEMUX is as follows:

```
ANSWERBAC
**** TTT              (Optional Output)
**** TTT              (Optional Output)
yy/mm/dd.hh.mm.ss.
CDC MULTI-MODE OPERATING SYSTEM     NOS 1
      .
      .
      .
      .
      .
END OF THE LINE---------------------------------------- n
(Page Eject)
ANSWERBAC
      .
      .
```

The n in the previous listing is the decimal encounter number at the end of the time-sharing stimulator run.

The following is the format of the DEMUX control statement.

DEMUX($p_1, p_2, \ldots, p_n$)

The following options are available. If the time-sharing output file STIMOUT will be processed at a later time, it is saved for subsequent processing.

I=lfn	lfn is the time-sharing output data file name. The default is STIMOUT.
O=lfn	lfn is the processed (demultiplexed) output file. OUTPUT is the default file name.
NT=xxx	xxx is the number of terminals to sort and extract (0< xxx< 512). xxx tells DEMUX how much core to allocate to process the input file.
NR	If NR is present, the input and output files are not rewound.
T	If T is present, the time of each carriage return and first output character is put out with the processed data. Subtracting the two time values gives the response time. The time values can also be used to correlate data between each terminal's input and output.

The following dayfile messages are possible.

END DEMUX RUN.	End of the run
ERROR DEMUX ARGUMENTS.	An error in the calling parameter to DEMUX
TOO MANY TERMINALS.	More than 512 terminals were designated with the NT parameter

To allow a selection of different types of postprocessing, the following sequence of control statements is processed after the stimulator run.

```
CALL(PERFORM)
EXIT.
```

These statements call a procedure file which contains user-prepared postprocessing statements. The GET is done to the same user catalog that the session file was obtained from. The following example is a typical procedure file.

PERFORM DEMUX(NT=310, O=NOS, T) GET(RESB) RESB.	RESB is a user-created program to analyze DEMUX output.
ASSIGN(NE, TRASH) DFD(OP=I, L=TRASH)	The system dayfile is dumped to null equipment.
REQUEST(T) DROP TELEX......	Processing stops until TELEX is dropped. Also, ASSIGN, 77 should be performed (refer to part II, section 4.3.3).
DFD(OP=I)	A partial dayfile dump is performed which includes TELEX statistics.
ATTACH(ERRDATA/M=W) COPYEI(ERRDATA, OUTPUT) EVICT(ERRDATA) RETURN(ERRDATA) EXIT. DIS.	Output of all failing jobs is copied to output.

In the previous procedure file, output from failing jobs is copied from the file ERRDATA. In order for ERRDATA to contain this information, another procedure file is necessary to collect data on jobs that fail and place this data in ERRDATA. This procedure file (called ERRCHEK in the following example) should be called in the following manner after each stimulator session.

```
GET,ERRCHEK/UN=usernum.
CALL(ERRCHEK)
```

The following is an example of this procedure file.

ERRCHEK IF(EF=0)GOTO, 99END.	If certain jobs normally cause compilation errors, additional procedures could be included to allow for these special cases.
DISPLAY(EF) ASSIGN(MS, OUTPUT) ENQUIRE(A) DAYFILE.	Data is gathered on which program and which statement caused the error.
ATTACH, ERRDATA/UN=usernum, M=W, NA. SKIPEI(ERRDATA) REWIND(OUTPUT) COPYEI(OUTPUT, ERRDATA) RETURN(OUTPUT, ERRDATA)	This data is placed on file ERRDATA, which is used later during postprocessing.
COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** COMMENT. FAIL******************** EXIT.	These comments warn users that failures have been encountered.
REQUEST(FAIL) ERRCHEK FAILED...	If serious problems have been encountered, DIS could be brought up at this point.
99END(EXIT)	

When the stimulator is used to compare response times between different versions of the operating system, care should be taken. Other factors may have a significant impact on response time. Awareness of the following items is necessary.

1. The entire parameter set should be the same for each stimulator run. The rate of activation of terminals, for example, can significantly impact response time.

2. The hardware configurations must be identical. If an attempt is made to measure small differences, all devices which cannot be restored to their original state (normally temporary devices) should be deadstarted and fully initialized.

3. If the possibility exists that permanent files have been destroyed, these files should be restored before each run. This can be accomplished easily by maintaining a copy of all required files under a user number not used during the stimulation. A procedure file can then be called to restore the files before each stimulator run. This is important when stimulation loads are causing abnormal job failures. A REPLACE shortly after an unexpected abort may cause incorrect information to be written, for example.

4. Job processing must be the same for each stimulator run. When many terminals are stimulated, random job failures may be overlooked. These failures may, however, affect the entire results of the stimulator run. To minimize this problem, it is necessary to check errors in all sessions quickly and easily. This can be done by using the procedures described earlier in this section.

By using the preceding techniques and procedures and noting the restrictions, it becomes easier to use the stimulator to check performance and reliability and to obtain meaningful data from the system.

5.6 STIMULATOR MESSAGES

Message	Fatal (F) or Nonfatal (N)	Significance	Action
DATA LINE TOO LONG	N	Too many characters are in the data line.	Shorten the line.
EMPTY SESSION FILE	N	The session file was empty.	Resupply the correct file name, or put data into the file.
ERROR IN STIMULATOR ARGUMENTS	F	There is a parameter present other than the I parameter, or the parameter is in the wrong format.	
FILE NAME NOT SET	N	A GO was entered, and the file name was not set.	Set the file name.
FORMAT ERROR	N	A character from ≡ to ; (60-77) or a parameter larger than seven characters was detected.	Reenter correct data.
ILLEGAL DATA	N	The data contains an illegal display character. This is commonly caused by not having a carriage return code at the end of the line.	Fix the data.
ILLEGAL FILE NUMBER	N	File number is greater than 18 bits.	Reenter correct decimal file number.

Message	Fatal (F) or Nonfatal (N)	Significance	Action
ILLEGAL OPTION	N	An illegal keyboard entry was made.	Reenter the correct option.
ILLEGAL USER INDEX	N	User index is greater than 18 bits.	Enter the correct user index.
IMPROPER NUMERIC PARAMETER	N	Field was too large, too small, or alphabetic.	Reenter correct data.
LOST STIMOUT DATA	N	Since CIO is not servicing the stimulator output buffer fast enough, the buffer has overrun.	
MEMORY OVERFLOW	F	There is not enough field length for the managed tables for the stimulator.	
MEMORY REQUEST ERROR	F	STIMULATOR and 1TS do not agree on the correct field length.	This could be caused by a system failure.
NO STIMULATOR TERMINALS DEFINED	F	Time-Sharing Subsystem has not initialized using the stimulator EST.	
TELEX ABORT	F	Time-Sharing Subsystem aborted while the stimulation was running.	Look at the dayfile for the cause.
TELEX NOT ACTIVE	F	Time-Sharing Subsystem is not at control point.	Bring Time-Sharing Subsystem to control point before running the stimulator.

STATUS/CONTROL REGISTER SIMULATOR 6

Error logging on a CDC CYBER 170 enables the occurrence of channel parity, memory parity, and other errors identified in the status/control (S/C) register to be detected and logged. The Status/Control Register Simulator (SCRSIM) allows the user to set S/C register bits in order to aid in the testing of error logging and error recovery procedures.

SCRSIM runs on CDC CYBER 170 machines using the S/C register on channel 16, and if more than 10 PPUs are available on the machine, the S/C register on channel 36. On CDC CYBER 70 machines, SCRSIM uses the interlock register on channel 15.

With the aid of a K display, the user can specify commands to set and clear bits, set bytes, and set lines and areas in holding registers. This allows both S/C registers to be set up completely. The contents of the holding registers can then be transferred to the S/C registers (64 or 128 bits are transferred to the interlock register of a CDC CYBER 70).

The bits set through this simulator are logged in the error log if an error bit is set, thus aiding in testing and software checkout. (Refer to the hardware reference manual for a complete description of the significance of each S/C register bit.) The simulator job dayfile lists all simulator commands entered. This error logging does not occur in a CDC CYBER 70, however, unless an appropriate IPRDECK entry has been made (refer to section 6.1.1).

CAUTION

Extreme care should be taken when using the simulator. Improper use may result in serious system malfunctions.

6.1 USING THE SIMULATOR

6.1.1 CDC CYBER 70 USE

Error logging is always enabled on a CDC CYBER 170; however, a zero TC (threshold count) parameter in the DELAY IPRDECK entry (refer to part II, section 5.10) disables error logging on a CDC CYBER 70. Although the default threshold count is zero, at deadstart time this parameter can be set to any value from 0 to 7777_8. Any nonzero threshold count enables error logging on a CDC CYBER 70.

Although the simulator operates on a CDC CYBER 70 if the threshold count is left at zero, it only manipulates bits in the interlock register. No error logging occurs.

6.1.2 CONSOLE OPERATION

The simulator is called from the console by entering:

X.SCRSIM.

The simulator left K display (Figure IV-6-1) appears on the left screen after entering

K, n. CR

where n is the control point number noted on the B display. This K display shows the contents of the temporary holding registers, as well as a central memory buffer. This buffer contains the following.

1. A history of all error status bits since the last level 0 deadstart. If an error status bit has been set in the S/C register, it remains set in the buffer, even though it may have been cleared in the actual S/C register. This history may be useful in diagnosing system malfunctions.

2. All other bits in the buffer reflect actual values in the S/C register at the time the last error bit was set. Each time an error bit is set, the entire buffer is updated.

Unless the simulator is running on a CDC CYBER 170 with more than 10 PPUs, the message

CHANNEL 36 NOT AVAILABLE

also appears. This indicates that no channel 36 S/C register is present on the machine, and thus, no simulation need be done for it.

By entering

KK. CR

the simulator right K display (Figure IV-6-2) is brought up on the right screen. This display gives a brief description of the commands available.

Commands can be entered on the K display by entering

K.comnd.

where comnd is one of the commands shown in Figure IV-6-2. These commands are described in section 6.2.

6.1.3 BATCH INPUT

The simulator may also be called from batch input by the control statement SCRSIM. The input file must have a record containing the commands to be processed, one command per card. The system must be in debug mode and the user must be validated for system origin priviledges.

```
                    TEMPORARY HOLDING REGISTER CONTENTS

      CHANNEL 16 REGISTER - LINES 0-3

BITS 59-    0                                                       BYTE 00
000000000000    000000000000    000000000000    000000000000    000000000000
    0000            0000            0000            0000            0000

BITS 119-  60                                                       BYTE 05
000000000000    000000000000    000000000000    000000000000    000000000000
    0000            0000            0000            0000            0000

BITS 179-120                                                        BYTE 10
000000000000    000000000000    000000000000    000000000000    000000000000
    0000            0000            0000            0000            0000

BITS 203-180                                                        BYTE 15
                                                000000000000    000000000000
                                                    0000            0000

                        CHANNEL 36 NOT AVAILABLE



                     ACTUAL S/C REGISTER ERROR BUFFER

      CHANNEL 16 REGISTER - WORDS 113-116
000000000000    000000000000    000000000000    000000000000    000000000000
000000000000    000000000000    000000000000    000000000000    000000000000
000000000000    000000000000    000000000000    000000000000    000000000000
                                                000000000000    000000000000
```

Figure IV-6-1. Simulator Left Display

SIMULATOR COMMANDS

COMMAND	DESCRIPTION
AREA, A, M, Y.	SET M BITS FROM A TO OCTAL VALUE Y
BYTE, XX, YYYY.	SET BYTE XX TO OCTAL VALUE YYYY
CLEAR, A, B, . . . , Z.	CLEAR BITS A, B, . . . , Z
CYCLE, X, T, R.	SET BIT X EVERY 16*T MS. R TIMES
END.	END CYCLE COMMAND BEFORE R REACHED
LINE, X, Y.	SET LINE X TO OCTAL VALUE Y
SET, A, B, . . . , Z.	SET BITS A, B, . . . , Z
+.	CHANGE REGISTER BEING USED AND K DISPLAY
GO.	ENTER HOLDING REGISTERS IN S/C REGISTERS
STOP.	END THE SIMULATOR

ALL BIT, BYTE, AND LINE NUMBERS ASSUMED DECIMAL.
TIME VALUES ASSUMED DECIMAL
Y AND YYYY VALUES MUST BE OCTAL.

Figure IV-6-2. Simulator Right Display

6.2 S/C REGISTER SIMULATOR COMMANDS

This section lists the commands available to the simulator user. The entire command keyword must be entered, and only one command may be entered at a time. Each command, except +, must end with a terminator. In all cases, a null argument is assumed to be zero.

6.2.1 HOLDING REGISTER COMMANDS

The following commands, except GO., affect only the holding register currently displayed on the left screen. These commands are used to set up the entire 204 bits in the holding registers. GO. transfers the holding register contents to the actual S/C register. (On a CDC CYBER 70, the channel 16 S/C register is simulated by the interlock register. GO. transfers the first 64 or 128 bits of the holding register to the interlock register.) The current contents of the holding register is displayed in binary and octal on the left screen (refer to Figure IV-6-1). The contents of the actual S/C register is also displayed in binary on the left screen.

Command	Description
AREA, a, m, y.	Set m bits in the holding register, from bit a to bit a+m-1, to the octal value y.
	a — Starting bit number, which may range from 0 to 203. a is assumed to be decimal, but a postradix of D or B may also be included.
	m — Number of bits to be set. m is assumed to be decimal, but a postradix of D or B may also be included.
	y — Value to which the bits are to be set. y may be up to m bits of octal value.
BYTE, xx, yyyy.	Set byte xx in the holding register to the octal value yyyy.
	xx — Byte number which may range from 0 to 16. xx is assumed to be decimal, but a postradix of D or B may be included.
	yyyy — Value to which byte xx is to be set. yyyy may be up to 12 bits of octal value.
CLEAR, $a_1, a_2, \cdots, a_n$.	Clear bits $a_1, a_2, \cdots, a_n$ in the holding register. If more than 30 bit numbers are entered, only the first 30 are processed. All others are ignored.
	a_i — Bit number, from 0 to 203, to be cleared. A decimal value is assumed, but a postradix of D or B may be included.

Command	Description
LINE, x, y.	Set line x of the holding register to the octal value y. x — Line number, ranging from 0 to 3, of the holding register shown on the left display screen (refer to Figure IV-6-1). Line 0 is positioned at the top and line 3 is positioned at the bottom. y — Value to which line x is to be set. y may be up to 60 bits of octal value.
SET, $a_1, a_2, \cdots, a_n$.	Set bits $a_1, a_2, \cdots, a_n$ in the holding register. If more than 30 bit numbers are entered, only the first 30 are processed. All others are ignored. a_i — Bit number, from 0 to 203, to be set. A decimal value is assumed, but a postradix of D or B may be included.
GO.	This command causes the contents of the holding register to be transferred to the actual S/C register, or the interlock register of a CDC CYBER 70. No bits in the holding register are changed by this command.

6.2.2 CYCLE COMMANDS

Command	Description
CYCLE, x, t, r.	Set bit x every t periods of time, a total of r times. This command assumes control of the simulator for the total time period specified. During this time, no command is accepted except END. Bit x is set in the holding register by this command. x — Bit number, from 0 to 203, to be set t — Number of periods of time for each cycle, one period being 16 milliseconds. t=32 is approximately 0.5 second. r — Number of times to set bit x. r may not exceed 4095.
END.	End CYCLE command processing before r is reached. Control of the simulator is returned to the operator.

6.2.3 + AND STOP. COMMANDS

Command	Description
+.	This command toggles the K display between the channel 16 and channel 36 register displays and also changes the register currently being worked on, if the simulator is being operated on a CDC CYBER 170 with two S/C registers. If two S/C registers are not present on the machine, no action is taken. The channel 16 register is assumed when the simulator begins. The only holding register which is affected by the holding register and cycle commands is the one currently displayed.
STOP.	This command causes the simulator to end processing.

6.3 SIMULATOR MESSAGES

6.3.1 K DISPLAY MESSAGES

CONVERSION ERROR.

A character has been entered after a postradix, or an 8 or a 9 is present in a number with a postradix of B.

CYCLE STILL PROCESSING.

A command other than END. was entered before the total time limit was reached.

HFM ERROR n

A Hardware Function Manager (HFM) error (code n) has occurred. Here, n may be one of the following.

n	Description
1	An HFM argument error has occurred. An illegal function code was encountered or a parameter-word address was out of range.
2	An HFM illegal request has occurred. HFM was called from other than a special system job, auto recall was not set, or the user did not have system origin privileges.
3	An illegal request for channel 36 has occurred. In this case, channel 36 is not present.

Refer to HFM. To obtain a listing, assemble HFM.

ILLEGAL BIT NUMBER.

The bit number specified was greater than 203.

ILLEGAL BYTE NUMBER.

The byte number specified was greater than 16.

ILLEGAL CHARACTER.

An alphabetic character other than a B or D was entered as a postradix on a decimal value, an alphabetic character, 8, or 9 was entered in an octal value argument, or a character with a display code of 60B or above was entered.

ILLEGAL COMMAND.

The command entered was not a legitimate SCRSIM command.

ILLEGAL LINE NUMBER.

The line number entered was not 0, 1, 2, or 3.

MAXIMUM NUMBER OF ARGUMENTS.

Only the first 30 bit numbers were accepted on a SET. or a CLEAR. command.

NO BITS SPECIFIED.

No bit numbers were specified on a SET. or a CLEAR. command.

NOT ENOUGH ARGUMENTS.

Before the correct number of arguments was specified, a terminator was entered.

NUMBER OF BITS TOO LARGE.

The number of bits entered on the AREA. command was larger than the number of bits from the starting bit to the end of the register.

NUMBER OF CYCLES TOO LARGE.

The number of cycles specified on the CYCLE. command was greater than 4095.

TIME LIMIT UP.

The total time limit on the CYCLE. command has passed. Input can again be accepted by the simulator.

TOO MANY BITS SPECIFIED.

More bits were specified than can be held in the area, line, or byte given.

6.3.2 DAYFILE MESSAGES

END SIMULATOR.

The operator has entered STOP. to drop the simulator.

881/883 PACK REFORMATTING UTILITY 7

Each 881 disk pack used in the 844 Disk Storage Subsystem contains factory-recorded flawing information on cylinder 632B (410D), track 0, sectors 0, 1, and 2. Each 883 pack contains this information on cylinder 1466B (822D), track 0, sectors 0, 1, and 2. The following information is contained.

- Cylinder 632B (or 1466B for 883 packs), track 0, sector 0 contains the factory-recorded manufacturing data. This data consists of the pack serial number and the manufacturing date.
- Cylinder 632B (or 1466B), track 0, sector 1 contains the factory map. This map contains a list of all factory-detected flaws, both correctable and uncorrectable.
- Cylinder 632B (or 1466B), track 0, sector 2 contains the utility map. This map originally contains all factory-detected uncorrectable flaws. This map is updated by the reformatting utility.

FORMAT is a CPU program which operates in conjunction with FDP, a PPU program, to maintain and reformat 881/883 disk packs. It is used to perform the following functions.

- Factory-recorded manufacturing data, factory-recorded flaw data, and utility flaw data can be retrieved from a factory-formatted disk pack.
- Sector and track flaws can be set or cleared on a factory-formatted disk pack.
- Address fields of a previously factory-formatted disk pack can be restored. (This function is used only in the event that addresses on the pack are lost.)

In order to function, FORMAT requires that the factory-recorded data (sectors 0 and 1 of cylinder 632B (or 1466B) be correct and readable. The pack cannot be processed if this data is unreadable. If packs are available which do not contain this factory-recorded information, consult a customer engineer to have this information placed on the packs. Also, the correct level of controlware must be present in order for FORMAT to function. To determine the controlware level and for procedures to install this controlware, refer to part III, section 2.6. Since the operating system requires that the utility map contain the physical flaw information in order for automatic logical flawing to be performed, it is important that the utility map be properly maintained.

The operating system automatically sets logical flaws when initializing 844 equipment. This is done by reading the utility map of the 844 units involved, and mapping this physical flaw information into the corresponding logical track addresses. Logical track flaw reservations are then made in the Track Reservation Table (TRT) for the 844 equipment being initialized. For example, if the 844 equipment being initialized consists of two physical units (such as a DI-2 configuration), the logical flaws set in the TRT are obtained from the physical flaw information recorded in the utility maps of both units making up the DI-2 configuration. This automatic flawing occurs only when an equipment is initialized, regardless of whether the initialization is done during deadstart, on-line, or is the result of running FORMAT.

The operating system allows the manual setting and clearing of flaw information. The RTK, STK, and TTK CMRDECK entries are used to manually set or clear logical track reservations in the equipment's TRT. (Refer to part II, section 4 for information concerning these entries.) If the device is then checkpointed, this flaw information is preserved in the TRT portion of the device label. RTK, STK, and TTK entries may be made during deadstart, during on-line initialization, or by using the FLAW utility (as described in section 7 of the operator's guide). In any case, the flawing done via these entries is only logical; the flaw information remains only in the TRT and is discarded on subsequent deadstart initialization unless manually reentered. This information is also lost during on-line initializations if it was not possible to recover the equipment. The use of RTK, STK, or TTK does not cause any additional information to be recorded in the utility map; only FORMAT is capable of updating the utility map data. Caution should be taken if attempts are made to cancel a logical flaw that was made during automatic flawing, since the physical disk sector is still marked as flawed and attempts to access that sector yield error conditions.

The following sections describe the use of the FORMAT utility for maintaining and reformatting 881/883 disk packs.

7.1 FORMAT CONTROL STATEMENT

Processing of maintenance operations on an 881 type disk pack is initiated by the FORMAT control statement. This program interfaces with the operator and a PPU program FDP. The format of this statement is:

FORMAT($p_1, p_2, \ldots, p_n$)

Each p_i is a parameter or a parameter equated to a value. The following is a list of parameters and values.

Keyword	Description
I-lfn	Specifies the input file. This file contains directives and data. If this parameter is omitted, the default is I=INPUT.
L=lfn	Specifies the output file. This file contains the information extracted from the disk pack. If this parameter is omitted, the default value is L=OUTPUT.
O=lfn	This optional parameter specifies an additional file to contain the output extracted from the disk pack.

NOTE

If output files other than OUTPUT are specified or if optional output files are specified (O=lfn parameter), these files should be created prior to the initiation of FORMAT. If not, these files are destroyed upon completion of FORMAT processing.

Keyword	Description
U=xx	Specifies the EST ordinal in octal of the 844 on which the disk pack is mounted. The unit is checked to ensure that it is available for formatting (refer to section 7.4).
P=sn	Specifies the pack serial number of the pack to be processed. If this decimal number does not exactly match the serial number recorded on the disk pack at the factory, processing does not occur.
G=m	Specifies the relative unit of a multispindle device. This value is checked for validity within the device. For example, if the device is a DI-2 and G=2 is entered, an error results.
MODE=opmode	Specifies the operational mode for this utility. If not specified, the default is MODE=FETCH. Valid declarations are the following.

opmode	Description
ALTER	This parameter indicates that the input file contains directives to control the set or clear flaw operations. These operations are described in section 7.2.
FETCH	This parameter indicates that the information contained on the factory sectors [cylinder 632B (or 1466B), track 0, sectors 0, 1, and 2] is to be obtained and copied to the output file (also the optional output file, if available).
RESTORE	This parameter indicates that addresses, flawed sectors, and tracks are to be restored according to information given in the utility flaw map. The utility flaw map must be intact or the program aborts.

Keyword	Description
V	This parameter causes the utility to verify the addresses recorded on the disk pack. This parameter can only be entered when MODE=FETCH or MODE=RESTORE.

If all default values are used, the following call is made.

FORMAT(I=INPUT, L=OUTPUT, MODE=FETCH, P=0)

At least the U and the P parameter must be correctly specified to initiate processing.

7.2 INPUT FORMATS

Input to FORMAT consists of control directives and data statements. Control directives indicate the type of operation to be performed. Data statements indicate locations on the pack where the operations are to be performed. A number of data statements may follow each control directive. Control directives and data statements are contained on the input file. This file is accessed only when MODE=ALTER has been specified on the FORMAT control statement. The input file (and therefore, control directives and data statements) has no significance when MODE=FETCH or MODE=RESTORE is specified.

7.2.1 CONTROL DIRECTIVES

Control directives begin in column 1. The format is:

directive

The following are acceptable directives.

Directive	Description
SET	Declares that the following data statements contain the addresses of flaws to be set and entered in the utility flaw map.
CLEAR	Declares that the following data statements contain the addresses of flaws to be cleared and deleted from the utility flaw map.
FINIS	Declares the end of the input. No information following this statement is processed. This directive is optional.

SET and CLEAR directives may be intermixed in the input file. However, all CLEAR operations are performed before any SET operation. Any attempt to alter the factory map or to set or clear sector flaws in a previously flawed track, results in an error.

7.2.2 DATA STATEMENTS

Data statements begin in column 1. The format is:

x, cccc, tt, ss

The parameters are described as follows:

Parameter	Description
x	Specifies the type of flaw to be set or cleared. Acceptable values are: **Value** — **Description** S — Indicates that the SET or CLEAR directive is applied to a sector T — Indicates that the SET or CLEAR directive is applied to a track
cccc	An octal number specifying the cylinder; the range is from 0 to 632B (or 1466B).
tt	An octal number specifying the track; the range is from 0 to 22B.
ss	An octal number specifying the sector; the range is from 0 to 27B. This field is ignored for track flaws (x=T).

All input data is checked to ensure that the values are within range. Any errors in input result in the termination of the utility before the disk is accessed. Any attempt to alter the factory map, or to set or clear sector flaws in a previously flawed track results in an error.

A maximum of 157 data statements may appear in the input stream.

7.3 OUTPUT FORMATS

Output generated by FORMAT is always placed on the output file (L=lfn on the FORMAT control statement). This file, for all modes of operation (ALTER, FETCH, and RESTORE), contains the following.

1. A listing of the input stream, if any
2. The pack serial number and date of factory formatting from the manufacturing section (cylinder 632B, or 1466B), track 0, sector 0)
3. A listing of the factor flaw map contained on cylinder 632B (or 1466B), track 0, sector 1
4. A listing of the utility flaw map contained on cylinder 632B (or 1466B), track 0, sector 2
5. A listing of the utility flaw map following any changes resulting from SET or CLEAR directives. This listing appears only when MODE=ALTER is specified on the FORMAT control statement.
6. A listing of the flawed sectors and tracks as read from the disk during address verification. This listing appears only when MODE=FETCH or MODE=RESTORE, and the V parameters are specified on the FORMAT control statement.

Refer to section 7.3.1 for examples of standard output.

The output generated by FORMAT can be directed to an optional output file (O=lfn). This file can then be used as input to another program, or it can be punched or printed.

7.3.1 EXAMPLES

The following three examples illustrate a series of reformatting operations performed on the same pack.

Example 1:

A RESTORE operation is performed on an 881 pack. A control statement similar to the following was entered.

FORMAT(U=xx, P=819545, MODE=RESTORE)

Figure IV-7-1 illustrates the resulting output.

```
DISK PACK MAINTENANCE UTILITY - VERSION 2
MODE = RESTORE

            DISK PACK SERIAL NUMBER
             819545

            DATE OF ORIGINAL FACTORY FORMATTING
             74/04/30

            FACTORY FLAW MAP
            (C=CORRECTABLE ERROR, S=SECTOR FLAW, T=TRACK FLAW)
            S, 632, 00, 00
            S, 632, 00, 01
            S, 632, 00, 02
            T, 302, 16, 00
            T, 362, 01, 00
            T, 373, 21, 00
            S, 626, 15, 15

            UTILITY FLAW MAP
            (S=SECTOR FLAW, T=TRACK FLAW)
            MAP EMPTY

PACK FORMATTING COMPLETE, VERIFICATION FOLLOWS
            S, 632, 00, 00
            S, 632, 00, 02
            S, 632, 00, 01

ADDRESS VERIFICATION COMPLETE
```

Figure IV-7-1. FORMAT Output, MODE=RESTORE

Example 2:

The flaws noted in the factory flaw map from example 1 (Figure IV-7-1) are now set in the utility flaw map. (The sectors containing the factory recorded flawing information, cylinder 632B, track 0, sectors 0, 1, and 2, are not flawed. This also applies to cylinder 1466B, track 0, sectors 0, 1, and 2 of 883 packs.)

Input similar to the following was entered.

```
FORMAT(U=xx, P=819545, MODE=ALTER)
7/8/9
SET
S, 626, 15, 15
T, 302, 16, 0
T, 362, 01, 00
T, 373, 21, 00
FINIS
6/7/8/9
```

Figure IV-7-2 illustrates the resulting output.

```
DISK PACK MAINTENANCE UTILITY - VERSION 2
MODE = ALTER

INPUT DATA
SET
S, 626, 15, 15
T, 302, 16, 00
T, 362, 01, 00
T, 373, 21, 00
FINIS

          DISK PACK SERIAL NUMBER
           819545

          DATE OF ORIGINAL FACTORY FORMATTING
           74/04/30

          FACTORY FLAW MAP
          (C=CORRECTABLE ERROR, S=SECTOR FLAW, T=TRACK FLAW)
          S, 632, 00, 00
          S, 632, 00, 01
          S, 632, 00, 02
          T, 302, 16, 00
          T, 362, 01, 00
          T, 373, 21, 00
          S, 626, 15, 15

          UTILITY FLAW MAP
          (S=SECTOR FLAW, T=TRACK FLAW)
          MAP EMPTY

          UTILITY FLAW MAP     (ALTERED)
          (S=SECTOR FLAW, T=TRACK FLAW)
          S, 626, 15, 15
          T, 302, 16, 00
          T, 362, 01, 00
          T, 373, 21, 00
```

Figure IV-7-2. FORMAT Output, MODE=ALTER

Example 3:

A FETCH with verification operation is performed to ensure proper reformatting. A control statement similar to the following was entered.

```
FORMAT(U=xx, P=819545, MODE=FETCH, V)
```

Figure IV-7-3 illustrates the resulting output.

```
 DISK PACK MAINTENANCE UTILITY - VERSION 2
MODE = FETCH

                    DISK PACK SERIAL NUMBER
                     819545

                    DATE OF ORIGINAL FACTORY FORMATTING
                     74/04/30

                    FACTORY FLAW MAP
                    (C=CORRECTABLE ERROR, S=SECTOR FLAW, T=TRACK FLAW)
                    S, 632, 00, 00
                    S, 632, 00, 01
                    S, 632, 00, 02
                    T, 302, 16, 00
                    T, 362, 01, 00
                    T, 373, 21, 00
                    S, 626, 15, 15

                    UTILITY FLAW MAP
                    (S=SECTOR FLAW, T=TRACK FLAW)
                    S, 626, 15, 15
                    T, 302, 16, 00
                    T, 362, 01, 00
                    T, 373, 21, 00

ADDRESS VERIFICATION FOLLOWS
                    T, 302, 16, 00
                    T, 362, 01, 00
                    T, 373, 21, 00
                    S, 626, 15, 15
                    S, 632, 00, 00
                    S, 632, 00, 02
                    S, 632, 00, 01

ADDRESS VERIFICATION COMPLETE
```

Figure IV-7-3. FORMAT Output, MODE=FETCH

7.4 ACCESSING MASS STORAGE DEVICES

Special procedures must be used when accessing the 844 drive used in the reformatting utility. Since certain FORMAT operations (ALTER and RESTORE) may change addresses on the pack, user access to the pack must be restricted.

FORMAT may operate on the pack in the following two ways.

1. A read operation (FETCH) obtains formatting information from the pack. The integrity of the pack is maintained.

2. Read and write operations (ALTER and RESTORE) may set and clear flaws, and addresses may be rewritten. Users cannot place permanent files on the pack when these operations occur. The integrity of the data on the pack is lost, so a full initialization of the pack must occur before system usage occurs.

7.4.1 ACCESS FOR READ OPERATIONS

To access the pack for read operations, the U parameter must be specified on the FORMAT control statement with the correct EST ordinal of the device containing the pack. In this case, the device must be a single-spindle device unless the G parameter is also specified. The P parameter must also be specified with the correct pack serial number.

In addition, FORMAT must be called from one of the following.

1. A system origin job (from the console)

2. A system privileged job (in this case, engineering mode must have been selected on the system console)

7.4.2 ACCESS FOR READ AND WRITE OPERATIONS

In addition to the information specified for read only operations, the following additional steps must be taken to access a device when write operations (ALTER and RESTORE) are to be performed. (Refer to the operator's guide for a description of all DSD commands.)

1. The pack to be accessed should be mounted on a removable mass storage device.	Refer to part II, section 4.4.10 for a description of the REMOVE CMRDECK entry.
2. One of the following conditions must hold.	
a. The device should not be a shared device.	Refer to SHARE command in part II, section 4.4.13.
b. If the device is shared, global unload should be set.	Refer to section 8.3.5.
3. The device must be declared logically off.	Use the OFF,ee. DSD command or the OFF parameter in the CMRDECK EQ entry. (Refer to part II, section 4.4.2 for a description of the CMRDECK entry.)
4. The MST display must show that the device is not in use.	Refer to the operator's guide for a description of the MST display.

5. One of the following conditions must hold.

   a. The MST display must show that the device is unavailable for permanent file access. — Refer to the operator's guide for a description of the MST display.

   b. All of the following must hold.

      1) The full initialize status and the format pending status must be set. — Use either of the DSD commands, INITIALIZE, ee,FP. or FORMAT,ee.

      2) The direct access file user count should be equal to zero. The MST display gives this information. — Refer to the operator's guide for a description of the MST display.

If all the necessary conditions are satisfied, FORMAT is able to access the pack for reformatting purposes. FORMAT repeatedly checks to ensure that these conditions are satisfied throughout the FORMAT operation.

7.5 CONSOLE MESSAGES

The following console messages are displayed to inform the operator of the status of the function being performed. In all cases, xxxxxx signifies the actual pack serial number as read from the manufacturing data recorded in cylinder 632B (or 1466B), track 0, sector 0.

ALTERING FLAW MAP S/N=xxxxxx

This status message indicates that the utility flaw map is undergoing modification.

RESTORING ADDRESSES S/N=xxxxxx

This status message indicates that the pack is currently undergoing restoration of the address fields. The control point should not be dropped while this message is displayed.

FETCHING FLAW DATA S/N=xxxxxx

This status message indicates that the factory recorded data is being retrieved from cylinder 632B (or 1466B), track 0, sectors 0, 1, and 2.

VERIFYING ADDRESSES S/N=xxxxxx

This status message indicates that a read only pass is being made across the pack. This message is displayed after successfully fetching the factory-recorded data and flaw maps, or after successfully restoring the address fields, if the V (verify) option was specified on the FORMAT control statement.

S/N MISMATCH - xxxxxx JOB ABORTED

This message indicates that FORMAT was terminated due to a mismatch between the serial number specified on the P parameter of the FORMAT control statement and the serial number recorded on the pack. xxxxxx is the serial number read from the pack.

7.6 DAYFILE MESSAGES

In addition to the console messages specified in section 7.5, all of which are entered in the system and control point dayfiles, the following messages are entered in the dayfiles to record conditions causing the program to abort.

INVALID DEVICE SPECIFIED

This message indicates that the device specified is in an improper state for the selected operation to proceed.

FILE EQUIVALENCE MAY NOT BE 0

This message indicates that either the input or the standard output file has been declared empty (that is, set equal to 0).

ILLEGAL FILE NAME - xxxxxxx

This message indicates that a file has been given an illegal or duplicate name (xxxxxxx).

INVALID DATA IN INPUT STREAM

This message indicates that the input file contains data that is incorrect. Refer to the listing of the input stream for statements in error.

INVALID PARAMETER ON PROGRAM CALL CARD

This message indicates that at least one unrecognizable parameter was found on the FORMAT control statement.

MANUFACTURING DATA INVALID

This message indicates that one of the factory-recorded sectors, containing either manufacturing or flaw data, is either unreadable or not present. Refer to the output listing for a detailed status report indicating the actual problem. If the factory-recorded data cannot be read, the pack cannot be processed using this utility. Customer engineering must be contacted to add this format information off-line.

TABLE OVERFLOW ON INPUT

This message indicates that too many flaw entries were available in the input stream; the flaw input limit is 157 flaws.

UNRECOVERABLE ERROR CONDITION OCCURRED

This message indicates that operation was terminated due to a nonrecoverable error. Refer to the general and detailed status described in the output listing for the specific error condition. If this condition occurs, it is extremely probable that the pack and/or disk drive is unusable in its present condition.

The following dayfile messages are issued by the PPU program FDP.

****** FORMAT ERROR xxxx ******

A channel malfunction has occurred, causing FDP to abort the control point. xxxx is one of the following.

xxxx	Malfunction
0001	The coupler was reserved from the opposite access.
0004	The disk drive was hung busy.
0010	An uncorrectable error has occurred.
0014	Status was expected, but none was received.
0015	An uncorrectable error on the channel connection occurred.
0024	An output failure occurred on the FORMAT parameter array.
0026	A read abort occurred.
0027	A detailed status abort occurred.
0032	An uncorrectable error occurred during formatting.

FDP ABORT - ENGINEERING MODE NOT SET.

ENGINEERING mode has not been set on the system console.

MULTIMAINFRAME OPERATION 8

The multimainframe feature provides a mechanism by which up to four† computers may access shared mass storage devices. This allows the mainframes to share preserved files residing on such devices. Preserved files are defined as those which are retained across a level 0 deadstart (nonsystem recovery). Types of preserved files are permanent files, I/O queues, and system dayfiles.

Any combination of one to four CDC CYBER 170, CYBER 70, Models 72, 73, or 74, or 6000 series mainframes may comprise a multimainframe environment. ECS is required with one CPU port for each mainframe. The presence of a DDP on a CPU port decreases by one the total number of mainframes that may run together.

Each mainframe in a complex may operate in a multimainframe mode or in stand-alone mode; however, two machines may not access the same device unless both are in multi-mainframe mode. A device is considered shared if it can be accessed by more than one of the mainframes; it need not be accessable to all the mainframes in the complex. The 841, 844 (single and double density), and ECS devices are the only devices that are supported as shared devices.

The fact that a computer is operating as part of a multimainframe complex is not apparent to the user; however, there are operational changes and additions which are of importance to the operator. These include deadstart commands, displays, on-line commands necessary in the event of an interruption on one of the sharing mainframes, and the UNLOAD/MOUNT process for removable devices.

8.1 OVERVIEW

ECS is used as the means and medium for controlling shared mass storage and inter-mainframe communication. Each mainframe has a CPU port into ECS through which system activity is controlled. In order to control shared mass storage devices, several ECS resident tables are required. The Device Access Table (DAT) contains the logical description (family name/pack name and device number) of each mass storage device (shared or nonshared) which is accessible by any machine in the complex. For each device in the DAT which is to be accessed by more than one machine, a corresponding Mass Storage Table (image of central memory resident MST) and Track Reservation Table (TRT) also reside in ECS. In addition, a Machine Recovery Table (MRT) exists in ECS for each machine and device (that is, there are as many MRTs for each shared device as there are mainframes in the complex).

†This feature has been designed and implemented to run on any combination of one to four mainframes. Due to resource limitations, the testing of this feature was conducted on one and two machine configurations.

Recovery allows a machine to either join other machines operational in a multimainframe environment or to operate in a stand-alone mode. The stand-alone system is not allowed to use the same mass storage devices as other machines. In the event of a system interruption to one machine, it is possible to operate the remaining machines in a multimainframe environment.

Automatic detection of ECS is not provided because it is not possible to determine its absence and continue to run on all machine types. For example, a 6600 will hang if an attempt is made to execute an ECS instruction without ECS. ECS status is checked by CMR (SET) when called upon to process an ECS entry in the CMRDECK.

8.2 DEADSTART

A multimainframe environment is defined at deadstart by CMRDECK entries in each machine. The following entries are used to do this.

CMRDECK Entry	Description
MID=id.	1- or 2-character machine identification. This identification is used to associate a specific machine with its access to a shared device. It is required, however, even if no devices are shared during system operation. This id must be unique for each machine in the complex. (If not specified, the default value is MID=AA.)
SHARE=ord_1,ord_2,...,ord_n.	This entry identifies mass storage devices which will be shared with other machines in the multimainframe environment. The MST/TRT for each device resides in ECS. (Each ord_i refers to an EST ordinal of a mass storage device.)
PRESET,n.	The PRESET entry must be specified on a level 0 deadstart on the first machine to be deadstarted in a multimainframe environment. This entry initializes the flag register and ECS resident tables for multimainframe use. The number of devices to be shared in the complex is specified by n. If n is not supplied, the default value is the number of shared devices defined for the particular machine. This entry can be specified only at the system console.
LINK=ord.	This entry defines the equipment at EST ordinal ord to be the link device. This device must either be of DE or DP type and cannot be defined as removable. The presence or absence of this entry defines whether the machine is to be run as part of a multimainframe complex or as a stand-alone system. If the LINK device is initialized, it must also be PRESET. 0 is not a legal entry for ord.

These entries are also described in part II, section 4.

Since in a multimainframe environment two or more machines can utilize the same mass storage devices, the device assignments and CMRDECKs of all machines are interrelated. Care must be taken to ensure proper CMRDECK settings for each machine to obtain the desired device configuration. The following items are important in obtaining this proper configuration.

1. Each machine must specify a unique machine identification (using the MID entry in the CMRDECK). This id associates a particular machine with its files on a shared device. There are no external characteristics associated with this identification. However, if the machine identification specified on a machine deadstarting into a multimainframe environment is identical to a machine identification on a machine already operating, the deadstart process halts and an appropriate message is displayed.

2. The assignment of shared mass storage devices should be made properly to ensure the recoverability of the device and the proper operation of the system. Assignment of shared mass storage is discussed in more detail in section 8.3.

3. Each machine in the multimainframe environment must specify ECS as the link device, using the LINK CMRDECK entry. An ECS entry must also be present in the CMRDECK. If the CMRDECK entries are present which indicate a multimainframe environment, a check is made to ensure that either a DE or DP equipment entry is also present. If none is found, an error message is given to the operator indicating that no link device has been defined. A link device is automatically designated as a shared device.

4. The first machine deadstarted in a complex must have the PRESET CMRDECK entry specified when deadstarting (level 0 only). This command causes tables to be preset in ECS, and in so doing, assures that no other machine has arrived at the same point in the deadstart sequence and is attempting the same thing. These other machines should not have the PRESET entry specified during deadstart. All other machines which arrive at that point in the deadstart process display a message indicating that they are waiting for deadstart on the preset machine. A machine that does not preset ECS has no means of detecting whether ECS has been preset previously by another machine. Therefore, the operator must ensure that ECS has been preset by a prior deadstart before deadstarting a particular machine without presetting ECS.

When a mainframe joins a multimainframe complex, it is associated with an identification which it utilizes during system operation but which is independent of the machine id. This association is done during deadstart when the machine investigates the MMF tables residing on the link device and places its machine id in an empty slot of the four that are available. Associated with each slot is a unique machine index and a unique machine mask, which the machine uses either to index itself into various MMF tables or to identify itself in these tables. The indices are 1, 2, 3, and 4. The masks are 1, 2, 4, and 8.

When a level 0 recovery deadstart will be performed on one of the machines in a multimainframe environment, the machine recovery utility (MREC) should be performed on some or all of the remaining machines before the deadstart proceeds. The purpose of this utility is to clear interlocks held by the machine to be deadstarted which have not been cleared by CPUMTR. It can also recover mass storage space on a shared device that is currently not accessable because of a machine interruption (necessitating a level 0 deadstart). MREC may have to be run from more than one machine since it affects only shared devices (that is, devices specified on the SHARE CMRDECK entry) of the machine on which MREC is run. If the interrupted machine shares different devices with different machines, MREC must be run from enough machines to account for all devices shared with the interrupted machine. The operator interface to MREC is described in the operator's guide.

Figure 8-1 generally illustrates the steps needed to deadstart a machine in a multimainframe environment.

If a level 3 recovery deadstart will be performed in a multimainframe system, the ECS MMF tables must be intact as well as CMR. For a level 1 or 2 recovery in an MMF environment, the ECS MMF tables must be intact.

8.3 SHARED MASS STORAGE

A major reason for operating a multimainframe complex is to be able to share mass storage devices between machines. Thus, the users of two systems may be able to utilize the same files if these files reside on a shared mass storage device. Having the ability to share files between systems also means that several additional procedures are required when operating a multimainframe complex. Additional consideration must also be given when assigning mass storage. This material, which is unique to multimainframe operation, is discussed in the following sections.

8.3.1 ASSIGNING SHARED MASS STORAGE

Some consideration should be given as to which devices should be made shared devices and which ones should not. Since a shared device contains preserved files which can be accessed by more than one machine, it must be physically connected to and logically defined (as shared) by each machine sharing the device. If one device of a family is defined as shared, then normally all devices in the family should be defined as shared for accessability. If a removable device will be treated as shared, it must be defined as removable in each machine sharing it. If a shared device is not removable, additional use of the device (beyond preserved files) is defined by each machine, independently of any other machine definition (that is, each machine must decide whether the device is to contain such things as a copy of the system or temporary files).

There is no real advantage in having nonpreserved files on a shared device, unless spreading them across shared devices outweighs having them on a dedicated device. A key factor is drive and controller contention. Another factor is whether or not enough drives and accesses to those drives are available to warrant the spreading of temporary files across several devices from all systems. Also, additional overhead is incurred by the system for each shared device. This includes additional CMR space for each device (100 CM words) and additional time that is required to maintain the device-related tables (such as the MST or the TRT) in the link device (ECS). These factors must be weighed to determine how best to configure shared devices.

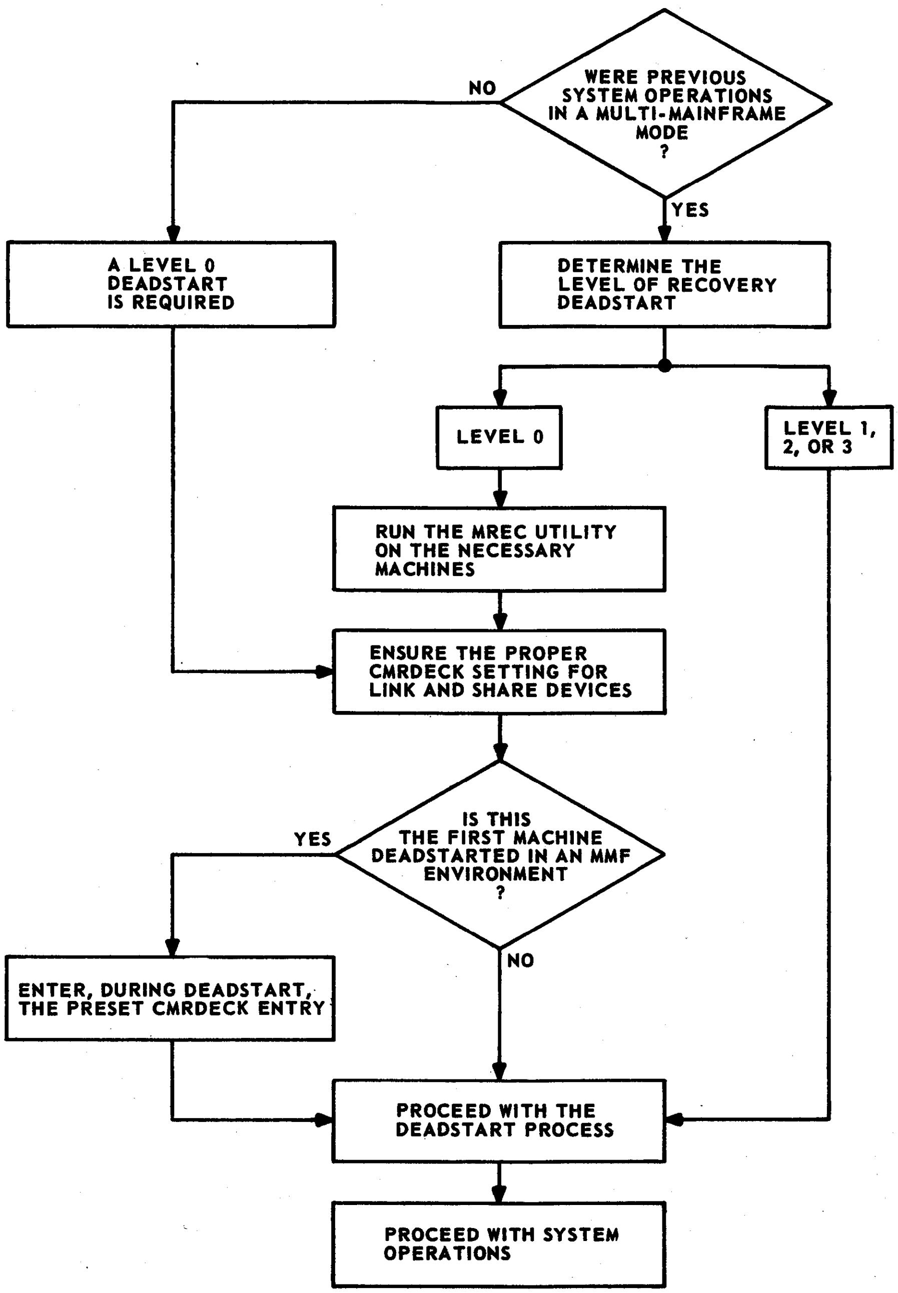


Figure 8-1. Deadstarting a Machine in an MMF Environment

8.3.2 DEVICE ACCESS TABLES

For purposes of device usage determination, tables are maintained in ECS identifying the status of all devices in the multimainframe complex. This includes shared and nonshared devices for all machines. These tables are called the Device Access Tables (DAT).

In order to minimize configuration problems, shared removable equipment should be configured the same way on all machines in the complex. For example, if one system defines three shared units as three MD-1's and another system defines the same units as an MD-3, the first system can accommodate an MD-2 on these units, whereas the second system would consider it an error. Unless the configurations are the same on all machines, any devices mounted on those drives may not necessarily be recoverable on all machines.

NOTE

RESEX considers only the configuration of the machine on which it is executing in its overcommitment algorithm.

8.3.3 MASS STORAGE RECOVERY TABLES

One problem that is created by having more than one machine sharing a mass storage device is that of recovering a machine's mass storage space and interlocks, should it require recovery processing. This problem is solved by defining a table which provides the information needed to recover a machine's mass storage space and by having a utility which performs the recovery. The table is called the Machine Recovery Table (MRT). There is one MRT for each mainframe per device. It tells which tracks are interlocked and which tracks are first tracks of files local to a particular machine. The MRT is utilized by CPUMTR on another machine to clear track interlocks and by the machine recovery utility MREC on another machine to recover the mass storage space of the interrupted machine. For specific information on the MRT, refer to volume 2 of the system reference manual or the system programmer's instant.

8.3.4 DEVICE INITIALIZATION

To initialize a mass storage device, it is necessary first to prevent any new activity from starting up on the device, then to wait until all current activity has completed, and finally to interlock the device and proceed with initialization. To accomplish this on shared mass storage devices in a multimainframe environment, the following steps must be taken.

1.	The INITIALIZE DSD command should be entered on the machine from which the initialization is to take place. If it is found that the INITIALIZE command has been entered from another machine, an error message is displayed.	Refer to section 3 of the operator's guide for information concerning the INITIALIZE command.
2.	All other machines sharing the device must unload it by entering the UNLOAD DSD command to prevent any new activity. (This command can be used for both removable and nonremovable shared devices.)	Refer to section 3 of the operator's guide for information concerning the UNLOAD command. Also refer to section 8.3.5 for additional information.

3. The machine from which the INITIALIZE was entered monitors the status of the other machines that are sharing the device. Once they have unloaded the device and user activity has ceased, initialization proceeds. Tables are also updated in ECS and the device is checkpointed.

4. To activate the device on the other machines, the MOUNT DSD command must be entered from each machine. This command clears the UNLOAD status. If initialization is still in progress on another machine when a MOUNT command is entered, the MOUNT process is terminated with an error.

Refer to section 3 of the operator's guide for information concerning the MOUNT command.

8.3.5 DEVICE UNLOAD

In a multimainframe environment, unloading a device involves more than it does under a one machine system. A device can be unloaded from a machine (referred to as a local unload), or it can be unloaded from the entire multimainframe complex (referred to as a global unload). A device can be physically removed from the complex only after a global unload has been accomplished. The general procedure to complete a global unload is illustrated in the following.

1. Enter the UNLOAD command from each machine. This is an indication to the machine that no new accesses should be initiated. This command must be entered from each machine sharing the device.

2. When all local unloads are set and user access has ceased, global unload status is set if the device is a removable device. This global unload status is displayed on all machines, indicating that there is no activity on the device from any machine and that the device may be physically unloaded.

3. The operator can then switch packs and enter the MOUNT command at the console to initiate recovery of the device. The MOUNT command clears the global unload status and the local unload status on the machine from which it was entered and indicates that this machine is now accessing the device. All other machines continue to ignore the device until the MOUNT command is entered on each machine. The MOUNT command does nothing if local unload status is not set on the machine.

8.3.6 DEVICE RECOVERY

Deadstart and on-line recovery methods use similar logic in recovering mass storage devices. When a device is recovered, the DAT in ECS is interlocked while a check is made to see if an entry exists for this device. The presence of an entry indicates that another machine is also accessing the device. If an entry is found, and the machine recovering the device has not been instructed to share it, an error is indicated and recovery halts with an appropriate message displayed. If the machine already accessing the device is not allowing it to be shared, the same error condition occurs. Therefore, if a device is being accessed, another machine can recover that device only if the recovering machine and the accessing machine use the device in shared mode.

CAUTION

If two devices recovered on separate machines have the same family name/device number or pack name, there is no method of determining whether or not they are the same device if both are shared. If they are different devices, they are both destroyed when used.

8.3.7 DEVICE CHECKPOINT

Local MST information for each machine which shares a mass storage device is maintained on the device. MST information for other machines may also be present on the device. The information for each machine is kept in one sector on the label track following the TRT sectors. Entries up to 31 unique machine ids can exit.

During checkpoint, only the local MST information of the machine performing the checkpoint is updated. Since local MST information for many machines is kept on the device, updating of all these areas by one machine could cause a loss of information needed if the device were to be used with another system.

For shared devices, duplication of checkpointing by more than machine is prevented. If a machine attempts to checkpoint a shared device and determines that another machine is performing the checkpoint (a checkpoint request bit is set in the local MST area of another machine), no action is taken. Only one checkpoint bit is set at any given time for a device.

INSTALLATION ON A 32K MACHINE A

Special procedures are needed to maintain the operating system on a 32K machine. These procedures are needed if the operating system is to be reassembled with or without corrective code on a 32K machine. This appendix outlines these special procedures.

CMRDECK ENTRIES

The following entries must be included in the CMRDECK.

```
NCP=2.
DAYFILE=eq,100.
ACCOUNT=eq,100.
ERRLOG=eq,100.
FNT=100.
LIB=2.
```

eq is the equipment number of the device on which the system dayfile, account dayfile, and error log dayfile are to reside.

For a description of all CMRDECK entries, refer to part II, section 4.

These entries cause the following to occur.

1. The total number of control points is reduced to two.
2. The buffer lengths for the system dayfile, account dayfile, and error log dayfile are reduced.
3. The length of the file name/file status table is reduced.
4. The second LIBDECK on the released deadstart tape is specified for use. This is a special LIBDECK to be used for installation on a 32K machine.

Additional changes to the CMRDECK should be made so that only one mass storage device is configured. Also, no more than one line printer, card reader, or card punch should be configured.

INSTALLATION PARAMETERS

The installation parameter CMXB should be changed from the released default value of 20B and set equal to 1B (refer to part III, section 2.4 for a description of CMXB). This installation parameter is set in the common deck COMPCMX. The latest listing of COMPCMX should be used to obtain the correct line number for CMXB. Since COMPCMX affects CPM, 1AJ, and 0VJ, these decks should be assembled and SYSEDITed into the running system. The following Modify input file can be used to accomplish this.

```
*IDENT MOD32K
*DECK COMPCMX
*DELETE,82
CMXB      EQU      1B
*EDIT,COMPCMX
*EDIT,CPM,1AJ,0VJ
```

NOTE

> CMXB should be changed for installation purposes only. It should be restored to the default value (20B) after installation is complete.

The following should be deleted.

PP/TDL-OVL/HPA7

This saves approximately 200B words by deleting on-line diagnostic routines from the PP and CP routine libraries.

ACTUAL INSTALLATION

The following steps should be taken when actually installing the operating system on a 32K machine.

1. Complete the initial setup of the installation procedures as described in part III, section 1.1.5.
2. Bring DIS to a control point and copy all input tapes required for installation to mass storage files. (For example, if the system OPL is on a magnetic tape, it should be copied to a mass storage file named TAPE.) A mass storage file (either permanent or local) should then be set up for use by the appropriate installation deck as an output tape. The installation decks use the name TAPE for input tapes and the name RELTAPE for output tapes. If these files (TAPE and RELTAPE) are local to the installation job, they are not requested via a VSN statement.
3. Disable the MAGNET and BATCHIO subsystems.
4. Run the job to install the operating system under DIS, which was brought to a control point in step 2. This job is described in part III, sections 1.1.4 and 2.3. The S3=1 option should be included in the CALL statement. This statement is then similar to the following.

   CALL(GENJOB(JOB=NOS, S3=1, S1=x)

   x may be set to 0 or 1.
5. Enable the MAGNET and BATCHIO subsystems upon completion of the installation job.
6. Create a new deadstart tape using the procedures described in part III, sections 1.1.6 and 2.5.
7. Deadstart the system using the new deadstart tape. The CEJ/MEJ switch on the deadstart panel should be turned off.
8. SYSEDIT into the running system only those products recommended in Table III-1-2. This prevents using memory table space for products not necessary for the installation of subsequent products.

9. During the initial setup of installation procedures, when running the job with the following statement

   CALL(GENJOB(JOB=GENFILS)

   an error will occur during the processing of the following control statement

   GTR(SYSTEM, DUMMY)OVL/HPA7

   and the job will fail. This is expected and should be ignored.

10. During the generation of a new deadstart tape using the procedure file GENSYS, the S1=1 option should be set. When this option is set, the deadstart tape is generated from an old deadstart tape and the permanent files created during the installation process. The released deadstart tape (or any subsequently created deadstart tape) should be used when the flashing REQUEST ODS (old deadstart tape) is issued.

INDEX

COMMENT SHEET

MANUAL TITLE CDC NOS 1 Installation Handbook

PUBLICATION NO. 60435700 REVISION D

FROM: NAME:

BUSINESS ADDRESS:

CUT ALONG LINE

AA3419 REV. 7/75 PRINTED IN U.S.A.

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

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

CUT ALONG LINE

FOLD FOLD

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

LITHO IN

CONTROL DATA CORPORATION