CONTROL DATA
CORPORATION

CONTROL DATA®
CYBER 70 SERIES MODELS 72/73/74
6000 SERIES
COMPUTER SYSTEMS

KRONOS® 2.1
INSTALLATION HANDBOOK

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released.
(6-29-73)	

Publication No.
60407500

© 1973
by Control Data Corporation
Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Technical Publications Department
4201 North Lexington Ave.
Arden Hills, Minnesota 55112

or use Comment Sheet in the back of this manual.

CONTENTS

PART I
INTRODUCTION

PART II
DEADSTART

PART III
INSTALLATION OF CONFIGURED SYSTEM AND PRODUCT SET

PART IV
SYSTEM MAINTENANCE

FIGURES

TABLES

SYSTEM AND MANUAL DESCRIPTION 1

1.1 PURPOSE AND ORGANIZATION OF MANUAL

The purpose of the installation handbook is to provide the information that an analyst with at least 6 months of experience with a CONTROL DATA® 6000 or a CDC® CYBER 70 computer system needs to install the KRONOS® 2.1 Operating System.

To meet these needs, the KRONOS® 2.1 Installation Handbook has the following general design.

Part I	Lists and describes all tapes released with the base KRONOS 2.1 system and those optional products that can be ordered separately. Contains the procedure to configure the deadstart tape that is released with the KRONOS 2.1 package so that it includes the product set and the required installation parameter settings.
Part II	Contains the procedure for deadstarting the system.
Part III	Contains specific installation information for the operating system and each product, such as the standard installation parameters and listings 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, VALIDUX, PROFILO (used for account validation), NETWORK (used to describe the communications network), and permanent file utilities. Some of these were formerly covered in the operator's guide.

KRONOS 2.1 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.

1.2 RELEASE DESCRIPTION

1.2.1 LIST OF PRODUCTS

KRONOS 2.1 PRODUCTS

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

KRONOS 2.1 Operating System
COMPASS 3.0
Export/Import 1.0
Modify 1.0
Record Manager 1.0
Direct Access 1.0
Indexed Sequential 2.0
SCOPE 3.4 system text, COMDECKs, and programs
Text Editor
Time-Sharing Subsystem
Update

Maintenance tools are also available for use with the base package, but they must be ordered separately. The maintenance tools consist of STIMULA, 1TS, DFSORT, FTNTRAN, PSAMP, FORTRAN Extended 3.0 object library, COBOL 3.0 object library, and Sort/Merge 3.0 object library.

OPTIONAL PRODUCTS

The following products can be ordered separately. The release materials for these products are defined in part I, sections 2.2 through 2.15.

ALGOL 3.0
APL 1.0
BASIC 2.1
COBOL 4.0
Cyberlink Interchange 1.0
FORTRAN 2.3
FORTRAN Extended 4.0
PERT/Time 1.2
SIMSCRIPT I.5 3.0
SIMULA 1.0
Sort/Merge 4.0
Time-Sharing FORTRAN 1.0
Transaction Subsystem 1.0

1.2.2 DEFINITION OF COMMON PRODUCTS

Some products are common to KRONOS and SCOPE, beginning with the releases of KRONOS 2.1 and SCOPE 3.4. As a result, the same product program library tape (referred to as PLxx) is released to a KRONOS 2.1 as to a SCOPE 3.4 site.

It is necessary that some of these common products be assembled with code that is only applicable to KRONOS if they are to be used with KRONOS 2.1 systems. Therefore, since the binaries for the common products may differ, there are additional release tapes for common products (referred to as KPLxx tapes) that contain binaries assembled to run with KRONOS 2.1 systems. (These tapes are only released with KRONOS 2.1 systems; the binaries to be used with SCOPE 3.4 systems are on the common PLxx tape.)

Each of the KPLxx tapes released for common products contains an empty file that normally would have contained the source program library for the product, which is released on the PLxx tape.

Each product has installation jobs on KPL2 which are listed and described in part I, section 2.1.1. The purpose of a product's job 1 is to generate a new KPLxx containing the program library and assembled binaries. For initial installation of common products, the input to job 1 is its PLxx tape. For subsequent updates to the product program library, KPLxx is input to job 1 and a new KPLxx is created as a result.

The product's job 2 collects the binaries resulting from running job 1 (KPLxx) and generates directives for a subsequent system build. Job 2 is dependent upon the structure (file position of binaries) of the tape created by job 1. Therefore, to allow job 2 to be run before job 1 for initial installation of common product binaries, empty files are placed on the release tape to maintain the same relative file position and file structure as the tape created as a result of running job 1.

For example, if COBOL 4.0 is ordered for use with a KRONOS 2.1 system, the site receives:

PL9 This is the common product tape that contains COBOL 4.0 source code (compiler and library routines in Update format) and the binary code to run COBOL with a SCOPE 3.4 system.

KPL12 This is the additional tape provided to install COBOL 4.0 with a KRONOS 2.1 system; it contains an empty file and the binary code of COBOL 4.0 assembled to run with a KRONOS 2.1 system.

The common products and their release material format for both SCOPE 3.4 and KRONOS 2.1 are:

Common Product	SCOPE 3.4 Materials	KRONOS 2.1 Materials
ALGOL 3.0	PL18	PL18, KPL8
COBOL 4.0	PL9	PL9, KPL12
COMPASS 3.0	PL2	PL2, KPL5
FORTRAN 2.3	PL21	PL21, KPL14
FORTRAN Extended 4.0	PL7, PL8	PL7, PL8, KPL15A, KPL15B
PERT/Time 1.2	PL13	PL13, KPL16
Record Manager 1.0	PL3	PL3, KPL6
Direct Access 1.0	PL3	PL3, KPL6
Indexed Sequential 2.0	PL3	PL3, KPL6
SIMSCRIPT I.5 3.0	PL27	PL27, KPL17
SIMULA 1.0	PL19	PL19, KPL18
Sort/Merge 4.0	PL10	PL10, KPL19
Update	PL1C	PL1C, KPL7

1.2.3 PRODUCTS UNIQUE TO KRONOS 2.1

Products unique to KRONOS 2.1 are released with only KPLxx tapes containing product source program library code and assembled binary code. These products are:

Product	Release Materials
APL 1.0	KPL9
BASIC 2.1	KPL11
Cyberlink Interchange 1.0	KPL13
Time-Sharing FORTRAN 1.0	KPL20
Transaction Subsystem	KPL21

RELEASE MATERIALS DESCRIPTION 2

This section lists the tapes released with the operating system package and each optional product. PL identified tapes are tapes for those products that are common to both SCOPE 3.4 and KRONOS 2.1; KPL identified tapes are tapes that are unique to KRONOS 2.1 and its product set.

2.1 KRONOS 2.1

The tapes released as part of KRONOS 2.1 are:

Deadstart tape		section 2.1.1
KPL1	KRONOS 2.1 source code	
KPL2	Modification, installation, and verification decks	section 2.1.1
KPL4	Binary and source code for SCOPE 3.4 system text, COMDECKs, and programs	section 2.1.1
KPL5	Binary code for COMPASS 3.0	section 2.1.2
PL2	Source code for COMPASS 3.0	section 2.1.2
KPL6	Binary code for Record Manager 1.0, Direct Access (DA) 1.0, and Indexed Sequential (IS) 2.0	section 2.1.3
PL3	Source code for Record Manager 1.0, DA 1.0, and IS 2.0	section 2.1.3
KPL7	Binary code for Update	section 2.1.4
PL1C	Source code for Update	section 2.1.4

The following tape, which must be ordered separately from the KRONOS 2.1 package, is also available.

KPL3	Maintenance tools	section 2.1.5

Materials that are required supplements to the KRONOS 2.1 base package are:

PL7	FORTRAN Extended 4.0 compiler source code	section 2.9
PL8	FORTRAN Extended 4.0 library source code	section 2.9
KPL15A	FORTRAN Extended 4.0 compiler binary code	section 2.9
KPL15B	FORTRAN Extended 4.0 library binary code	section 2.9

2.1.1 OPERATING SYSTEM

Deadstart tape

The KRONOS 2.1 deadstart tape contains binaries of:

- KRONOS 2.1 Operating System
- COMPASS 3.0
- Export/Import 1.0
- Modify 1.0
- Record Manager 1.0
  - Direct Access 1.0
  - Indexed Sequential 2.0
- SCOPE 3.4 system text, COMDECKs, and programs
- Text Editor
- Time-Sharing Subsystem
- Update

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

KPL1

The KRONOS 2.1 system OPL contains the source programs for:

- KRONOS 2.1 Operating System
- Export/Import 1.0
- Modify 1.0
- Text Editor
- Time-Sharing Subsystem

It has the following characteristics: labeled, 7- or 9-track, Modify format, 800 bpi, binary recording mode, one file, KRONOS*2P1 as file id in HDR1 label.

KPL2

KPL2 contains source code in Modify format for the decks necessary to modify and install the products as well as decks to verify that they are correctly installed. There are two or three decks (jobs) on the tape for each product.

Use job 1 to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.

Use job 2 to collect the binaries from the various products' KPLs and generate directive files for later use as input to the procedure file GENSYS (described in part I, section 3.2.2).

Use job 3 to verify that the product is correctly installed.

KPL2 has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, one file, file id in HDR1 label is INSTALLDECKS*2P1.

KRONOS 2.1 Decks

Product	Job Name
KRONOS 2.1 system including Export/Import 1.0, Modify 1.0, Text Editor, and Time-Sharing Subsystem	KR1
7054/844-2 Disk Storage Subsystem Controlware	BCS2
Installation decks tape (KPL2)	INS1
COMPASS 3.0	CMP1 CMP2
Record Manager 1.0	SIXRM1 SIXRM2 SIXRM3
Direct Access 1.0	DA1 DA2 DA3
Indexed Sequential 2.0	IS1 IS2 IS3
Update	UPD1 UPD2
SCOPE 3.4 system text, COMDECKs, and programs	SCP1 SCP2
Maintenance tools	MTN1 MTN2
FORTRAN Extended 3.0 object library	F3L1 F3L2
COBOL 3.0 object library	C3L1 C3L2
Sort/Merge 3.0 object library	S3L1 S3L2

Optional Products Decks

ALGOL 3.0	ALG1 ALG2 ALG3
APL 1.0	APL1 APL2 APL3

Product	Job Name
BASIC 2.1	BAS1 BAS2 BAS3
COBOL 4.0	CBL1 CBL2 CBL3
Cyberlink Interchange 1.0	CYB1 CYB2 CYB3
FORTRAN 2.3	RUN1 RUN2 RUN3
FORTRAN Extended 4.0 compiler	FTNC1 FTNC2 FTNC3
FORTRAN Extended 4.0 library	F4LIB1 F4LIB2
PERT/Time 1.2	PRT1 PRT3
SIMSCRIPT I.5 3.0	SMS1 SMS2 SMS3
SIMULA 1.0	SIM1 SIM2 SIM3
Sort/Merge 4.0	SRM1 SRM2 SRM3
Time-Sharing FORTRAN 1.0	TSR1 TSR2 TSR3
KRONOS Transaction Subsystem	KTS1 KTS2

KPL4

This tape is a program library in Update format that contains the SCOPE 3.4 common decks that are necessary to generate the SCOPE 3.4 system text and additional programs that support the common product set. These decks are from the SCOPE 3.4 tapes named PL1A and PL1B.

KPL4 has the following characteristics: labeled, 7- or 9-track, Update format, 800 bpi, binary recording mode, SCOPE3P4TEXTS*2P1 as file id in HDR1 label, three files.

File 1 Source code in Update format for the following SCOPE PL1A decks: IPARAMS, PPSYS, CPSYS, SISICOM, ACTCOM, COMSRAS, COMACIO, COMAREG, COMAFET, CPCTEXT, IPTEXT, PPTEXT, SCPTEXT, and CPUTEXT.

File 2 Source code in Update format for the following SCOPE PL1B decks: LMACOM, PFCOM, CPCOM, LDRCOM, UCLOAD, IOCOM, CPC, COMCCIO, IORANDM, IO, CHEKPT, SYSEQ, COMCSYS, LDRTEXT, and PFMTEXT.

File 3 Absolute binary code for system text overlays and relocatable binary code for I/O routines.

The common decks included on KPL4 that comprise the SCOPE 3.4 system text are:

ACTCOM	CPU program system action request macros
COMAFET	File environment table generation macros
COMSRAS	System communication symbols
CPSYS	CPU I/O macros using CPC
IPARAMS	Installation parameters for common product set
LMACOM	CPU program loader request macros
PFCOM	Permanent file macros for SCOPE 3.4
PPSYS	SCOPE 3..4 PPU system definitions
SISICOM	Indexed Sequential 2.0 macros
COMACIO	SCOPE 3.4 CPU I/O macros
COMAREG	SCOPE 3.4 replacement for R= pseudo instruction

The 6RMCOM common deck and the IOTEXT deck are not on KPL4; they are on the Record Manager PL3 tape described in section 2.1.3. 6RMCOM contains the Record Manager user macros.

Table I-2-1 indicates which common decks are contained in the eight system text decks.

TABLE I-2-1. SCOPE 3.4 SYSTEM TEXT DECKS

COMDECK Name	Constituent Decks							
	CPCTEXT	IOTEXT	IPTEXT	LDRTEXT	PFMTEXT	PPTEXT	SCPTEXT	CPUTEXT
ACTCOM	X	X					X	X
COMAFET	X						X	X
COMSRAS	X	X				X	X	X
CPSYS	X						X	
IPARAMS			X					
LMACOM				X				
PFCOM					X			
PPSYS						X	X	
SISICOM	X							
6RMCOM		X						
COMACIO								X
COMAREG								X

2.1.2 COMPASS 3.0

KPL5 KPL5 contains binary code for COMPASS 3.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, COMPASS3P0*3P4 as file id in HDR1 label, two files.

File 1	Empty file
File 2	Absolute binary code

PL2 This tape contains the COMPASS 3.0 source code. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, COMPASS3P0*3P4 as file id in HDR1 label.

File 1	Source code in Update format

Other files do not apply to KRONOS 2.1.

2.1.3 RECORD MANAGER 1.0, INDEXED SEQUENTIAL (IS) 2.0, DIRECT ACCESS (DA) 1.0

KPL6 KPL6 contains the absolute and relocatable binary code for Record Manager 1.0, Indexed Sequential 2.0, and Direct Access 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, RM*3P4 as file id in HDR1 label, nine files.

File 1	Empty file
File 2	IOTEXT and TXT6RM binary code
File 3	I/O modules binary code
File 4	FILE control card program absolute binary code
File 5	FILE control card program relocatable binary code
File 6	Empty file
File 7	IS 2.0 relocatable binary code
File 8	Empty file
File 9	DA 1.0 relocatable binary code

PL3 This tape contains the source code for Record Manager 1.0, IS 2.0, and DA 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, RM*3P4 as file id in HDR1 label.

File 1	Source code in Update format for Record Manager 1.0
File 6	Source code in Update format for IS 2.0
File 8	Source code in Update format for DA 1.0

Other files on PL3 do not apply to KRONOS 2.1.

2.1.4 UPDATE

KPL7 KPL7 contains binary code for Update. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode,UPDATE1P2 as file id in HDR1 label, two files.

File 1	Empty file
File 2	Absolute binary code

PL1C PL1C contains the Update source code. It has the following characteristics: labeled, 7- or 9-track, binary recording mode, 800 bpi, UPDATE1P2 as file id in HDR1 label.

File 1	Source code in Update format

Other files do not apply to KRONOS 2.1.

2.1.5 MAINTENANCE TOOLS

KPL3 KPL3 is the maintenance tools tape. It contains the KRONOS Time-Sharing Stimulator (STIMULA and 1TS), the dayfile sort program (DFSORT), the FORTRAN Translator (FTNTRAN), PSAMP (analyzes the P register), and the object libraries for COBOL 3.0, FORTRAN Extended 3.0, and Sort/Merge 3.0.

The purpose of the object libraries is to allow relocatable binary code created under KRONOS 2.0 to satisfy external symbols from the version 3.0 libraries. These libraries do not use Record Manager for input/output; they retain calls to CIO. These libraries are only included to allow the use of 2.0 binary decks during the 2.0 to 2.1 transition phase.

KPL3 has the following characteristics: unlabeled, 7- or 9-track, 800 bpi, binary recording mode, eight files.

File 1	Source code in Modify format for STIMULA, 1TS, DFSORT, FTNTRAN, and PSAMP
File 2	Absolute binary for STIMULA, 1TS, DFSORT, FTNTRAN, and PSAMP
File 3	FORTRAN Extended 3.0 object library in Update format
File 4	FORTRAN Extended 3.0 object library binary
File 5	COBOL 3.0 object library in Update format
File 6	COBOL 3.0 object library binary
File 7	Sort/Merge 3.0 object library in Update format
File 8	Sort/Merge 3.0 object library binary

2.2 ALGOL 3.0

KPL8 This tape contains the ALGOL 3.0 binaries. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, ALGOL3P0*3P4 as file id in HDR1 label, four files.

File 1	Empty file
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines
File 4	Relocatable binary code of compiler overlays

PL18 This tape contains the source code for ALGOL 3.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, ALGOL3P0*3P4 as file id in HDR1 label.

File 1	Source code in Update format

Other files on PL18 do not apply to KRONOS.

2.3 APL 1.0

KPL9 This tape contains the source and binaries for APL 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, APL1P0*2P1 as file id in HDR1 label, five files.

File 1	Source code in Update format
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	Relocatable binary code for terminal and batch processors

2.4 APT

To be released in a subsequent release of KRONOS 2.1.

2.5 BASIC 2.1

KPL11 This tape contains the source and binary code for BASIC 2.1. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, Update format, BASIC2P1*2P1 as file id in HDR1 label, four files.

File 1	Source code in Update format
File 2	Absolute binary code of compiler overlays and user library
File 3	Relocatable binary of library routines
File 4	Relocatable binary code of compiler overlays

2.6 COBOL 4.0

KPL12 This tape contains the binary code for COBOL 4.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, COBOL4P0*3P4 as file id in HDR1 label, five files.

File 1	Empty file
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines
File 4	Relocatable binary code of compiler overlays
File 5	Relocatable binary code of COPYCL overlay

PL9 This tape contains the COBOL 4.0 source code. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, COBOL4P0*3P4 as file id in HDR1 label.

File 1	Source code in Update format

Other files on PL9 do not apply to KRONOS installations.

2.7 CYBERLINK INTERCHANGE 1.0

KPL13 This tape contains the source and binary code for Cyberlink 1.0. It has the following characteristics: labeled, 7- or 9-track, Modify format, 800 bpi, binary recording mode, Cyberlink1P0*2P1 as file id in HDR1 label, two files.

File 1	Source code in Modify format
File 2	Absolute binary code of Cyberlink subsystem routines

2.8 FORTRAN 2.3

KPL14 This tape contains the binary for FORTRAN 2.3. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, RUN2P3*3P4 as file id in HDR1 label, three files.

File 1	Empty file
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines

PL21 This tape contains the source code for FORTRAN 2.3. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, RUN2P3*3P4 as file id in HDR1 label.

File 1	Source code in Update format

Other files on PL21 do not apply to KRONOS installation.

2.9 FORTRAN EXTENDED 4.0

KPL15A This tape contains the FORTRAN Extended 4.0 compiler binary code. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, FTN4P0COMP*3P4 as file id in HDR1 label, four files.

File 1	Empty file
File 2	Relocatable binary code of compiler overlays without DEBUG
File 3	Relocatable binary code of compiler overlays with DEBUG
File 4	Absolute binary code of compiler overlays

KPL15B This tape contains the FORTRAN Extended 4.0 library binary code. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, FTNLIBS*3P4 as file id in HDR1 label, two files.

File 1	Empty file
File 2	Relocatable binary code for library routines

PL7, PL8 These tapes contain the source code for FORTRAN Extended 4.0. They have the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, FTN4P0COMP*3P4 and FTNLIBS*3P4 as file ids in HDR1 labels.

File 1 of PL7	Compiler source code in Update format
File 1 of PL8	Object library source code in Update format

Other files on PL7 and PL8 do not apply to KRONOS installations.

2.10 PERT/TIME 1.2

KPL16 This tape contains the binary code for PERT/Time 1.2. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, PERT*3P4 as file id in HDR1 label, three files.

File 1 Empty file
File 2 Absolute binary code of compiler
File 3 Relocatable binary code of compiler

PL13 This tape contains the source code for PERT/Time 1.2. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, PERT*3P4 as file id in HDR1 label.

File 1 Source code in Update format

Other files on PL13 do not apply to KRONOS installations.

2.11 SIMSCRIPT I.5 3.0

KPL17 This tape contains the binary code for SIMSCRIPT 3.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, SIM153P0*3P4 as file id in HDR1 label, four files.

File 1 Empty file
File 2 Absolute binary code of compiler overlays and user library
File 3 Relocatable binary code of library routines
File 4 Relocatable binary code of compiler overlays

PL27 This tape contains the source code for SIMSCRIPT 3.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SIMI53P0*3P4 as file id in HDR1 label.

File 1 Source code in Update format

Other files on PL27 do not apply to KRONOS installations.

2.12 SIMULA 1.0

KPL18 This tape contains the binary code for SIMULA 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SIMULA1P0*3P4 as file id in HDR1 label, four files.

File 1 Empty file
File 2 Absolute binary code of compiler overlays
File 3 Relocatable binary code of library routines
File 4 Relocatable binary code of compiler overlays

PL19 This tape contains the source code for SIMULA 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SIMULA1P0*3P4 as file id in HDR1 label.

File 1 Source code in Update format

Other files on PL19 do not apply to KRONOS installations.

2.13 SORT/MERGE 4.0

KPL19 This tape contains the binary code for Sort/Merge 4.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SORT4P0*3P4 as file id in HDR1 label, four files.

File 1	Empty file
File 2	Absolute binary code of compiler overlays
File 3	Relocatable binary code of library routines
File 4	Relocatable binary code of compiler overlays

PL10 This tape contains the source code for Sort/Merge 4.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, SORT4P0*3P4 as file id in HDR1 label.

File 1	Source code in Update format

Other files on PL10 do not apply to KRONOS installations.

2.14 TIME-SHARING FORTRAN 1.0

KPL20 This tape contains the source and binary code for Time-Sharing FORTRAN 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, binary recording mode, TSRUN1P0*2P1 as file id in HDR1 label, two files.

File 1	Source code in Modify format
File 2	Absolute binary code of compiler and library routines

2.15 TRANSACTION SUBSYSTEM 1.0

KPL21 This tape contains the source and binary code for the Transaction Subsystem 1.0. It has the following characteristics: labeled, 7- or 9-track, 800 bpi, Modify format, binary recording mode, TRANSACTSYS*2P1 as file id in HDR1 label, three files.

File 1	Source code in Modify format
File 2	Absolute binary code of KTS subsystem and user libraries
File 3	Relocatable binary code of library routines

INSTALLATION OF A CONFIGURED SYSTEM AND PRODUCT SET 3

3.1 GENERAL PROCEDURE

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 VALIDUX (and PROFILO and NETWORK, if needed).

   Part IV, section 1

   To use the system without VALIDUX, turn off job account verification.

   Type DISABLE, VALIDATION.
   Press CR

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 1
      Procedure in part I, section 3.2

   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 2 through 15
      Procedure in part I, section 3.2

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

3.2 PROCEDURE TO CREATE A NEW DEADSTART TAPE

It is necessary to create a new deadstart tape to include customer requirements, such as adding optional products or modifying installation parameters. A new tape must also be created whenever PSR updates are necessary to any products on the deadstart tape.

3.2.1 REQUIRED MATERIALS

To install the product set (composed of one or more products) or to modify and/or update the operating system and product set, use the installation job decks on KPL2 for each product and the three procedure files GENINSD, GENFILS, and GENSYS.

INSTALLATION JOB DECKS

There are either two or three installation jobs for each product on the KPL2 tape described in part I, section 2.1.1.

Job 1 Use this job to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPLxx) containing updated source and binary code.

Job 2 Use this job to collect the binaries from the various products' KPLs and generate directive files for later use as input to the procedure file GENSYS.

Job 3 Use this job to verify that the product is correctly installed.

PROCEDURE FILES

The procedure files GENINSD, GENFILS, and GENSYS are written in KCL, the KRONOS job control language, and are on the KRONOS 2.1 deadstart tape.

GENINSD This program lists and punches the installation decks for all products to be installed.

GENFILS This program creates four system common files to be used by job 2 installation decks and also the GENSYS procedure file program. The four common files created by executing GENFILS are:

File Name	Description
ZZZZCCF	Contains control card records that are generated by the job 2 installation decks. This file is input to the GENSYS procedure file.
ZZZZLDF	Contains LIBEDIT directives generated by the job 2 installation decks.

File Name	Description
ZZZZSDF	Contains SYSEDIT directives generated by the job 2 installation decks.
ZZZPSBF	Contains the product set binaries collected by the job 2 installation decks.

GENSYS This program adds the products' binaries from ZZZPSBF into the running system (SYSEDIT process) and/or creates a new deadstart tape containing these products. GENSYS performs these functions by reading SYSEDIT and/or LIBEDIT directives from the common files ZZZZSDF and ZZZZLDF, reading control card records from ZZZZCCF, and accessing the product set binaries from common file ZZZPSBF.

3.2.2 PROCEDURE

1. Call the GENINSD procedure file program to list and punch the installation decks for all products to be installed. A GENINSD control card call has the following format.

```
GENINSD(JD=xxxx)
```

xxxx is the name of the job deck to be punched.

Use a job similar to the following to punch the required KPL2 decks. It must create a common file with the name MTPL that contains the contents of KPL2, and it must include a GENINSD control card call for each deck to be punched.

```
JOB card
ACCOUNT(usernumber, password)
LABEL(KPL2, R, FI=INSTALLDECKS*2P1, MT, D=HY, F=I)
COPY(KPL2, MTPL)IN
RETURN(KPL2)
LOCK(MTPL)
COMMON(MTPL)
RETURN(MTPL)
GENINSD(JD=xxxx)
     .
     .
     .
GENINSD(JD=xxxx)
6/7/8/9
```

2. Execute the GENFILS procedure file program. To execute GENFILS from a batch job, run a job similar to the following:

```
JOB card
ACCOUNT(usernumber, password)
GENFILS.
6/7/8/9
```

3. If it is necessary to modify the current values of any installation parameters or to add PSR updates, use the following procedure; otherwise continue with step 4.

The standard installation parameters and also installation job listings are in part III, sections 1 through 15.

For each product that requires modification and/or update:

a. Use Modify or Update directives (depending upon the format of the product's source code) to insert any installation parameter changes or PSR updates in the product's job 1.

b. Run job 1 to create a new KPLxx tape that contains the modified and/or updated source and binary code. (If a PLxx tape was input to a job 1, it is no longer useful.)

4. Run the job 2 installation decks for all of the products in the product set that are to be added to the deadstart tape. These jobs collect the binaries from the products' KPLs on ZZZPSBF and generate directives on files ZZZZSDF, ZZZZLDF, and ZZZZCCF for later use as input to procedure file GENSYS.

   Run the jobs according to the following order. Steps a, b, c, and d may be omitted when starting from the released deadstart tape if no changes (installation parameters or PSR updates) are necessary.

   a. SCOPE 3.4 system text and programs

   b. Update

   c. COMPASS 3.0

   d. Record Manager 1.0

   e. FORTRAN Extended 4.0 compiler

   f. FORTRAN Extended 4.0 object library

   g. Sort/Merge 4.0

   h. Indexed Sequential 2.0

   i. Direct Access 1.0

   j. FORTRAN 2.3

   k. Other products in any order

5. Execute GENSYS to add the products' binaries from ZZZPSBF (generated in step 4) into the running system (SYSEDIT process) and/or to create a new deadstart tape containing these products.

   The system must be relatively inactive in order to run GENSYS; in particular, do not run the Time-Sharing Subsystem during this process.

   To execute GENSYS, type the following DSD command at the system console.

```
X.GENSYS(SM=a, DS=b)
```

a Specifies whether or not to add product set binaries to running system (SYSEDIT)

a=0	Do not add binaries to running system
a=1	Add binaries to running system

b Specifies whether new deadstart tape is to be created

b=0	Do not create deadstart tape
b=1	Create new deadstart tape

Both a and b must be specified in the command; otherwise, the job aborts.

When DS is set to 1, the common files are released at the end of the GENSYS run; they are not released if DS is set to 0.

6. If the SYSEDIT option was selected in step 5 (SM=1), run the verification jobs if desired.

7. If a new deadstart tape was created in step 5 (DS=1), deadstart using the new tape according to the procedures in part II, section 1.

   Run verification jobs, if desired.

3.3 INSTALLATION DEPENDENCIES

Table I-3-1 indicates the relationship between the various products.

I	Item is necessary to modify and/or update a product.
B	Item is necessary to modify and/or update as well as execute a product.
O	Item is optional to execute a product.

TABLE I-3-1. INSTALLATION DEPENDENCIES

	Items Required in Running System								
Product To Be Modified, Updated Or Executed	CMP 3.0	RM 1.0	3.4 TXT	UTD	RUN 2.3	FTN 4.0 COMP	FTN 4.0 LIB	SIMSC 3.0	S/M 4.0
KRONOS 2.1, E/I 1.0, Modify, Text Editor Time-Sharing Subsystem	I								
COMPASS 3.0	I			I					
Record Manager 1.0	I		I	I					
Indexed Sequential 2.0	I	B	I	I		I	O①		
Direct Access 1.0	I	B	I	I					I
SCOPE 3.4 Texts	I			I					
Update	I			I					
Maintenance Tools	I				I				
FORTRAN Extended 3.0④ object library									
ALGOL 3.0	I		I	I					
APL 1.0	I			I					
BASIC 2.1	I			I					
COBOL 4.0	I	B	I	I			O②		O③
Cyberlink Interchange 4.0	I								
FORTRAN 2.3	I		I	I					
FORTRAN Extended 4.0 Compiler	I			I		I			
FORTRAN Extended 4.0 Library	I	B	I	I		I			
PERT/Time 1.2				I	I				
SIMSCRIPT 3.0	I	B	I	I				I	
SIMULA 1.0	I		I	I					
Sort/Merge 4.0	I	B	I	I					
Time-Sharing FORTRAN 1.0	I								
Transaction Subsystem 1.0	I								

①IS requires FORTRAN Extended 4.0 library to be in the running system to run the two IS utilities SYSTAT and ESTMAT.

②COBOL requires the FORTRAN Extended 4.0 arithmetic library routines to be in the running system to use the exponentiation feature.

③COBOL requires Sort/Merge 4.0 to be in the running system to use the SORT verb.

④Maintenance of the FORTRAN Extended 3.0 object library requires that job F3L1 be run on a KRONOS 2.0 system with FORTRAN Extended 3.0 installed.

3.4 OBJECT LIBRARIES

The operating system and product installation decks on KPL2 create certain object libraries. The following list defines the contents of the object libraries created if all the products supported by KRONOS 2.1 are installed. The library names that are referenced are reserved for Control Data Corporation.

Library	Section
BASLIB	part I, section 3.4.1
BDMLIB	part I, section 3.4.2
COBOL	part I, section 3.4.3
FORTRAN	part I, section 3.4.4
RUN2P3	part I, section 3.4.5
SIMLIB	part I, section 3.4.6
SYSIO	part I, section 3.4.7
SYSMISC	part I, section 3.4.8

3.4.1 BASLIB

```
      CATALOG OF BASLIB          FILE      1
REC   NAME      TYPE           LENGTH    CKSUM     DATE       COMMENTS

  1   BASLIB    ULIB             363      1502   73/06/12.
  2   BASOGEN   REL             2225      7462  73/06/04.
       BASOPRT
       BASOCON
       BASOCHK
       BASOCLS
       BASOMOV
       BASOTAB
       BASOSRT
       BASOWRT
       RCLF
       BASOMGN
       BATOMGN
       CHKMRGN
       LNLWDCH
       TABFLG
       STRFMT
       VALU
       STARSW
       VBLSTR
       STOP
       OBUFLCL
       SETDGTS
       WNB=
       CIO=
       MSG=
       SYS=
       BASOFMX
       BATOFMX
       BASOSRU
       BATOSRU
       BASOPRU
       BATOPRU
       BASOFLI
       BATOFLI
       BASOUSU
       BATOUSU
       BASOFML
       BATOFML
  3   BASEGEN   REL              602      1301   73/06/04.
       BASESRT
       BASEGEN
       BASEDAT
       BASEDIM
       BASESUB
       BASEXIT
       BASEONS
       BASERET
       BASEZER
       BASRCHK
       FINDLBL
       GETNXTL
       ILLBL
       BASEJMP
       BATEJMP
       CODSTRT
      CATALOG OF BASLIB           FILE      1
REC   NAME        TYPE          LENGTH    CKSUM      DATE        COMMENTS

       CODEND
       BASERRS
       BATERRS
       BASERSS
       BATERSS
       VALESL
       VALESM
       VALNXL
       SUBINCR
  4   BASIGEN     REL            237       7637   73/06/04.
       BASICHK
       BASISRT
       BATISRT
       BASIEND
       BATIEND
       BASIREW
       BATIREW
       BATOSRT
       BATOWRT
       BATOPRT
       BATOCON
       BATEGEN
       BATESRT
       BATEDAT
       BATEDIM
       BATEONS
       BATESUB
       BATERET
       BATEZER
       BATRCHK
  5   BASIINP     REL           1433       1363   73/06/04.
       BASIINP
       BATIINP
       BASIRED
       BATIRED
       BASIDEL
       BATIDEL
       BASINOD
       BATINOD
       BASOSET
       BATOSET
       BASICON
       BASIUNP
       BASISCN
       DLMTNO
       DLMTSW
       DLTKND
       STRINP
       BSTRBUF
       FFCLASS
       BASOFET
       BATOFET
       BASONFT
       BATONFT
       BASOFFT
       BATOFFT
       CATALOG OF BASLIB           FILE      1
REC    NAME      TYPE            LENGTH    CKSUM      DATE       COMMENTS

 6     BASATRI   REL               207      4603   73/06/04.
        BASASIN
        BATASIN
        BASACOS
        BATACOS
        BASATAN
        BATATAN
 7     BASAATN   REL               120      6335   73/06/04.
        BASAATN
        BATAATN
 8     BASALEP   REL               362      4423   73/06/04.
        BASALOG
        BATALOG
        BASALGT
        BATALGT
        BASAEXP
        BATAEXP
        BASAPWR
        BATAPWR
 9     BASARST   REL               472      4575   73/06/04.
        BASASQR
        BATASQR
        BASAABS
        BATAABS
        BASAINT
        BATAINT
        BASAROF
        BATAROF
        BASASGN
        BATASGN
        BASARND
        BATARND
        BASACLK
        BATACLK
        BASATIM
        BATATIM
        BASATAB
        BATATAB
        BASEESL
        BATEESL
        BASEESM
        BATEESM
        BASENXL
        BATENXL
        BASDATE
        BATDATE
        BASCLCK
        BATCLCK
10     BASMOPR   REL              2165      4650   73/06/04.
        BASMADD
        BATMADD
        BASMSUB
        BATMSUB
        BASMSCM
        BATMSCM
        BASMMLT
        CATALOG OF BASLIB          FILE      1
REC     NAME        TYPE          LENGTH     CKSUM       DATE        COMMENTS

          BATMMLT
          BASMTRN
          BATMTRN
          BASMZER
          BATMZER
          BASMCON
          BATMCON
          BASMIDN
          BATMIDN
          BASMRED
          BATMRED
          BASMPRT
          BATMPRT
          BASMINV
          BATMINV
          BASMRFL
          BATMRFL
          BASMINP
          BATMINP
          BASMWRT
          BATMWRT
          BASACSV
          BATACSV
          BASACVS
          BATACVS
          BASXDIS
          BATXDIS
          BASXLEN
          BATXLEN
          BASXVAL
          BATXVAL
          BASXSTR
          BATXSTR
          BASXSAV
          BATXSAV
          BASXSBF
          BATXSBF
          BASXSBS
          BATXSBS
          BASXSBT
          BATXSBT
          BASACMP
          BATACMP
          BASASTR
          BATASTR
          BASIINS
          BATIINS
 11     BASLIB      OPLD               25      3156      73/06/12.

 12     * EOF *         SUM =       11121
```

3.4.2 BDMLIB

```
       CATALOG OF BDMLIB          FILE      1                          73/06/12. 02.42.07.      PAGE       1
PEC    NAMF      TYPE          LENGTH     CKSUM      DATE        COMMENTS

  1    RDMLIB    ULIB            101       5257   73/06/12.
  2    RDMI      REL            7761       1771  73/05/20. 72/08/05. 72/12/18. KTS RATCH DATA MANAGER INTERFACE.
        SYS=
        RCL=
        WNB=
        MSG=
  3    CALLTSK   REL              31       6055  73/05/20. 72/07/09.           REQUEST SCHEDULING OF TASK(S).
        CALLTSK
  4    CEASE     REL              20       3267  73/05/20. 72/07/09.           TERMINATE TASK EXECUTION.
        CEASE
  5    CMDUMP    REL              57       1524  73/05/20. 72/07/24. 72/10/05. DUMP CENTRAL MEMORY.
        CMDUMP
  6    DMGR      REL             267       6377  73/05/20. 72/07/09.           DATA MANAGER REQUESTS.
        ADDR
        RLKGET
        RLKPUT
        GETT
        GETL
        GETN
        GETNL
        GETB

        GETR
        GETRR
        GETRBL
        GETRL
        GETNR
        GETNRL
        LOCKF
        PURGER
        PUT
        PUTF
        PUTI
        PUTIF
        PUTR
        PUTRF
        PUTRI
        PUTRIF
        RECALAL
        RFLES
        RFLESAL
        RECHAIN
        REPOS
        UNLOK
        UNLOKAL
        UNLOKF
  7    DSDUMP    REL              40       2017  73/05/20. 72/07/24. 72/10/05. DIRECT SUBSEQUENT DUMPS.
        DSDUMP
  8    JOURNL    REL              31       7047  73/05/20. 72/07/09.           ISSUE JOURNAL MESSAGE.
        JOURNL
  9    SEND      REL              70       0467  73/05/20. 72/07/09.           SEND MESSAGE TO TERMINAL.
        SEND
 10    TARO      REL              36       5667  73/05/20. 72/08/31.           TERMINAL ARGUMENT OPERATIONS.
        TARO
 11    TSIM      REL              40       2541  73/05/20. 72/08/31.           TERMINAL STATUS AND INFORMATION.
        TSIM
 12    RDMLIB    OPLD             27       5571   73/06/12.

 13    * EOF *        SUM =    11007
```

3.4.3 COBOL

```
        CATALOG OF COBOL          FILE      1
 REC    NAME       TYPE         LENGTH     CKSUM      DATE       COMMENTS

   1    COBOL      ULIB           2063      1267    73/06/12.
   2    DDSUBMV    REL             156      4747   73/06/06.
          D.SUBMV
   3    DDCKETL    REL              32      3332   73/06/06.
          D.CKETL
   4    DDCKLA     REL             345      4571   73/06/06.
          D.CKLA
          D.MVNSL
   5    DDWALAB    REL             233      4506   73/06/06.
          D.WALAB
   6    DDOPWA     REL             477      0212   73/06/06.
          D.CKFLM
          D.CLOWA
          D.CLVWA
          D.CWARL
          D.GTPT
          D.OPIWA
          D.OPMWA
          D.OPOWA
          D.OPXWA
   7    DDSQWA     REL             137      5026   73/06/06.
          D.GTNWA
          D.PTNWA
   8    DDRAWA     REL             212      4401   73/06/06.
          D.GETWA
          D.PUTWA
          D.REMWA
          D.SEKWA
          D.MODWA
   9    CCFINIS    REL             116      7510   73/06/06.
          D.MSOPN
  10    CCCOBIO    REL            1154      7753   73/06/06.
          D.NOCFL
          D.EXALL
          D.GETRT
          D.SVRTN
          D.OPIN
          D.OPOT
          D.OPIO
          D.CLOS
          D.CLVSQ
          D.DX
          D.EX
          D.OPX
          D.ERWRT
          D.STKPT
          D.NEWP
          D.SRTFL
          D.SRT65
          D.RAAE
          D.ATEN
          D.CBFS
          D.FINIS
  11    DDLABLS    REL             376      1336   73/06/06.
          D.LABMV
          D.FTLAB
        CATALOG OF COBOL          FILE     1
 REC    NAME      TYPE           LENGTH   CKSUM      DATE       COMMENTS

          D.UNFRM
          D.LBFET
          D.HDR1
          D.EOV1
          D.EOF1
          D.FRMLB
          D.BLKLB
  12    DDSQLAB   REL              572     6054   73/06/06.
          D.SQLAB
          D.LX
          D.TEMPL
  13    DDMLLIT   REL               34     0017   73/06/06.
          D.MLLIT
  14    DDCMLIT   REL              117     0645   73/06/06.
          D.CMLIT
          D.COLIT
  15    DDSBLIT   REL               22     5563   73/06/06.

  16    DDZPAR    REL               65     5362   73/06/06.
          D.ZPAR
  17    DDSQIO    REL              255     5015   73/06/06.
          D.CLOSQ
          D.GETSQ
          D.GTNSQ
          D.OPISQ
          D.OPMSQ
          D.OPOSQ
          D.OPXSQ
          D.PTNSQ
          D.PUTSQ
  18    DDSTIO    REL              351     7606   73/06/06.
          D.CLOSD
          D.GETSD
          D.OPISD
          D.OPMSD
          D.OPOSD
          D.PUTSD
          D.SEKSD
  19    DDDAIO    REL              266     1333   73/06/06.
          D.CLODA
          D.GETDA
          D.GTNDA
          D.MODDA
          D.OPIDA
          D.OPMDA
          D.OPODA
          D.PTNDA
          D.PUTDA
          D.REMDA
          D.SEKDA
          D.SKPDA
  20    DDISIO    REL              217     0664   73/06/06.
          D.CLOIS
          D.GETIS
          D.GTNIS
          D.MODIS
```

```
       CATALOG OF COBOL          FILE     1
REC    NAME        TYPE        LENGTH   CKSUM      DATE        COMMENTS

         D.OPIIS
         D.OPMIS
         D.OPOIS
         D.PTNIS
         D.PUTIS
         D.REMIS
         D.SEKIS
         D.SKPIS
 21    DDAKIO      REL            224    2655   73/06/06.
         D.CLOAK
         D.GETAK
         D.GTNAK
         D.MODAK
         D.OPIAK
         D.OPMAK
         D.OPOAK
         D.PUTAK
         D.REMAK
         D.SEKAK
         D.SKPAK
 22    DDUPCNT     REL             54    2026   73/06/06.
         D.UPCNT
         D.RERUN
 23    DDADVAN     REL            223    1176   73/06/06.
         D.WBA
         D.WAA
 24    DOUSE       REL            271    5203   73/06/06.
         D.USE
         D.CLUSE
         D.USEX
         D.BBF
         D.ABF
         D.BEF
         D.AEF
         D.BBRF
         D.ABRF
         D.BERF
         D.AERF
         D.ABR
         D.BER
         D.AEBBR
         D.AER
         D.BBR
         D.KEYC
         D.ERRAB
 25    DDEXP       REL             52    7543   73/06/06.
         D.EXP
 26    DDANCM      REL             40    7522   73/06/06.
         D.NUMCM
 27    IFNUMA      REL             52    4140   73/06/06.
         D.NUMCA
 28    DDIFALP     REL             26    7165   73/06/06.
         D.ALPCM
 29    DDALPC1     REL             22    6072   73/06/06.
         D.ALPC1
```

```
        CATALOG OF COBOL            FILE       1
REC     NAME     TYPE             LENGTH     CKSUM      DATE       COMMENTS

 30     DDSUBSC  REL                  23      5117   73/06/06.
          D.SBSC1
 31     DDSBSC2  REL                  31      6426   73/06/06.
          D.SBSC2
 32     DDSBSER  REL                  25      7677   73/06/06.
          D.SBSER
          D.SBSCE
 33     DDSBSC6  REL                  35      0563   73/06/06.
          D.SBSC6
 34     DDSBSC7  REL                  41      4510   73/06/06.
          D.SBSC7
 35     DDSOL    REL                 154      4320   73/06/06.
          D.SOL
          D.SOLP
          D.SOLE
          D.OVRLY
          D.SOLP1
          D.SOLA
 36     DDSORT   REL                1141      0755   73/06/06.
          D.SORT
          D.SORT1
          D.SORT3
          D.SORT4
          D.SRTFN
          D.SRTL1
          D.SRTL2
 37     DDDSPLY  REL                 462      5214   73/06/06.
          D.DSPLY
          D.WRDSP
 38     DDXCEPT  REL                 271      6602   73/06/06.
          D.XCEPT
 39     DDXCPT   REL                 213      4635   73/06/06.
          D.XCPT
 40     DDADSUB  REL                  23      6274   73/06/06.
          D.ADSUB
 41     DDDADD   REL                  37      6132   73/06/06.
          D.DADD
 42     DDFIVED  REL                 111      3057   73/06/06.
          D.TEMPI
          D.FIVED
          D.TEMPN
          D.TEMPC
          D.TEMP
 43     CONSTAN  REL                 176      3207   73/06/06.
          D.TEMP7
          D.ZERO
          D.SIXES
          D.BLANK
          D.TEMPS
          D.SRCH1
          D.CCPAR
          D.MXOFF
          D.FL
          D.TNTH
          D.STOR
          D.STOR1
```

```
       CATALOG OF COBOL              FILE     1
REC    NAME       TYPE             LENGTH    CKSUM      DATE       COMMENTS

         D.MASK
         D.DATE
         D.HOV
         D.STOP
         D.PAUSE
 44    DDTRUBL    REL                 175     2200   73/06/06.
         D.TRUBL
         D.ETD
         D.DFMSG
         D.UNDFP
         D.TRBL2
         D.TRBL1
 45    DDZONE     REL                  73     5753   73/06/06.
         D.ZONE0
         D.ZONE1
         D.ZONE2
         D.ZONE4
 46    DDZN3A     REL                  21     4660   73/06/06.
         D.ZN3A
 47    DDZN5A     REL                  21     7445   73/06/06.
         D.ZN5A
 48    DDZN6A     REL                  21     3721   73/06/06.
         D.ZN6A
 49    DDZNTAB    REL                  21     7701   73/06/06.
         D.ZN
 50    STRPN      REL                  16     1454   73/06/06.
         D.STRPN
 51    DDSTPTB    REL                 117     1160   73/06/06.
         D.STRP
 52    DDZN7A     REL                  21     0273   73/06/06.
         D.ZN7A
 53    DDSTP1A    REL                  41     1636   73/06/06.
         D.STP1A
 54    DDSTP2A    REL                  44     1625   73/06/06.
         D.STP2A
 55    DDSTRP3    REL                  75     3721   73/06/06.
         D.STRP3
         D.STRP4
 56    DDDATCN    REL                 102     5612   73/06/06.
         D.DATCN
 57    DDTENDP    REL                 124     6273   73/06/06.
         D.TENDP
         D.SLASH
         D.SLSH1
         D.QUOT1
         D.QUOT2
         D.DIVS1
         D.DIVS2
         D.DIVD2
         D.DIVD1
 58    DDTENS     REL                  36     0705   73/06/06.
         D.TENS
 59    DDTNTHS    REL                  63     7564   73/06/06.
         D.TNTHS
         D.FIVES
```

```
      CATALOG OF COBOL          FILE      1
REC   NAME      TYPE        LENGTH     CKSUM     DATE      COMMENTS

 60   DDCVBD    REL            200      2372  73/06/06.
       D.CVBD3
       D.CVBD5
       D.CVBD6
       D.CVBD7
       D.CVBD1
       D.CVBD2
 61   DBN1SA    REL             31      7206  73/06/06.
       D.BN1SA
 62   DDBN      REL             32      5113  73/06/06.
       D.BN1SS
 63   DBN6SA    REL             33      0017  73/06/06.
       D.BN6SA
 64   DBN6SS    REL             35      3777  73/06/06.
       D.BN6SS
 65   BN1DSA    REL             34      0320  73/06/06.
       D.B1DSA
 66   DBN1DS    REL             51      2617  73/06/06.
       D.BN1DS
 67   BN6DSA    REL             35      0336  73/06/06.
       D.B6DSA
 68   DBN6DS    REL             51      1557  73/06/06.
       D.BN6DS
 69   DB1DDA    REL             60      7551  73/06/06.
       D.B1DDA
 70   DBN1DD    REL             53      4724  73/06/06.
       D.BN1DD
 71   DB6DDA    REL             60      2740  73/06/06.
       D.B6DDA
 72   DBN6DD    REL             53      2303  73/06/06.
       D.BN6DD
 73   DBN1SB    REL             23      2073  73/06/06.
       D.BN1SB
 74   BN1SBA    REL             27      5116  73/06/06.
       D.B1SBA
 75   DDED      REL            535      5251  73/06/06.
       D.ED3
       D.ED5
       D.ED1
       D.ED2
       D.EDI
       D.AO
 76   AOPTN     REL             62      2703  73/06/06.
       D.A06
       D.A07
 77   DDEDAL    REL            202      7461  73/06/06.
       D.EDAL
 78   DDEOP     REL            225      5116  73/06/06.
       D.EOP1
       D.EOP2
       D.EOP3
       D.EOP4
       D.EOP5
       D.SAVA
 79   DDEXAMO   REL            360      2574  73/06/06.
       D.EXAMO
       CATALOG OF COBOL            FILE      1
REC    NAME       TYPE           LENGTH    CKSUM      DATE       COMMENTS

 80    DDMOVE     REL               156     4053   73/06/06.
        D.MOVE
 81    DDLITCU    REL                56     2347   73/06/06.
        D.LITCU
 82    DDLITCC    REL                61     0310   73/06/06.
        D.LITCC
 83    DDBCDCC    REL               116     7036   73/06/06.
        D.BCDCC
 84    DDCOLSQ    REL               143     2762   73/06/06.
        D.SHFT
        D.COLSQ
        D.CONV
 85    DDBCDCU    REL               106     5175   73/06/06.
        D.BCDCU
 86    DDBCDCM    REL               147     6711   73/06/06.
        D.BCDCM
        D.BCDCP
 87    DDSCM      REL               137     6112   73/06/06.
        D.CMSML
        D.BCDAC
 88    CONV1      REL               117     5406   73/06/06.
        D.CONV1
 89    DDPAGE     REL               142     0326   73/06/06.
        D.PAGE
 90    DDLNCT     REL               105     5040   73/06/06.
        D.LNCT
 91    DDSPACE    REL               134     4444   73/06/06.
        D.SPACE
 92    DDINITL    REL               136     5424   73/06/06.
        D.CFRST
        D.INIT
        D.NIT
        D.INITL
        SVMNR
 93    DDPRINT    REL               276     2355   73/06/06.
        D.PRINT
        D.REPC
        D.REPQ
        D.REPA
        D.RHPH
 94    DDRGEN     REL               372     0615   73/06/06.
        D.RSET
        D.RGEN
        D.CHG00
        D.CHG01
        D.CHG02
        D.TABL2
        D.CHG03
        D.CHG04
        D.CHG05
        D.CHG06
        D.CHG07
        D.CHG08
        D.CHG09
        D.RPGRP
        D.RPTAB
```

```
      CATALOG OF COBOL          FILE      1
REC   NAME      TYPE          LENGTH    CKSUM      DATE      COMMENTS

        D.ADDFL
        D.ADDSM
        D.LEVL
        D.TERM
        D.LEVEL
        D.ERROR
        D.FETR
        D.JSTIN
        D.RESC
        D.RJ
 95   DDBEGRP   REL               23     3741   73/06/06.
        D.BEGRP
 96   DDENDRP   REL               23     4004   73/06/06.
        D.ENDRP
 97   DSMCON    REL               42     3351   73/05/10.
        SMCON7
 98   TSMCON    REL               55     0306   73/05/10.
        SMCONP
        SMCONB
 99   MSMCON    REL               36     0336   73/05/10.
        SMCONM
100   SORTL     REL              222     5512   73/05/10. SORT MACRO ROUTINE LOADER
        SORTL
101   MACPRO    REL              776     1070   73/05/10.
        MACPRO
        S.PTLBF
        S.OWNCD
        S.PARAM
        S.SCF
102   SOCHKR    REL               36     4005   73/05/10.
        SOCHKR
        CKPEOR
103   ENDPRO    REL             4024     4373   73/05/10.
        XLTBUFF
        ENDPRO
        S.ERTBL
        DPARAM
        MACFWA
104   EXTRACT   REL              462     1747   73/05/10.
        EXTRACT
        ASSIGN
        PMFILL
        CTFILL
        XDECBI
        XBIDEC
        LVLBLZ
        ERMSG
        ERMSGS
        S.PRP2
105   KYCPL     REL              232     4443   73/05/10.
        S.EXTR2
        KYCPL
106   KEYCOD    REL             1545     7631   73/05/10.
        KEYCOD
        KED1
        KED2
        CATALOG OF COBOL         FILE      1
REC     NAME       TYPE         LENGTH   CKSUM      DATE       COMMENTS

         KED4
107     BUFALL     REL            500     0671   73/05/10.
         S.BUFAL
108     TRNSRT     REL           1327     0151   73/05/10.
         TRNINT
         S.TRNI
         TRNMRG
         TRM.I01
         TRM.I02
         TRM.I03
         TRS.I01
         TRM.EX
         VECFIL
         TRNSRT
         TRNFLSH
         EX6NLAB
         MDGP.3
         EXIT3
         EX3
         EXAMX0
         MVWDS
109     TSC        REL            527     7166   73/05/10.
         TSC
         SM.TSC
         TSC.DO
         IMTFC
110     TMC        REL            577     5425   73/05/10.
         TMC
111     NEXRCM     REL            116     3205   73/05/10.
         NEXRCM
         MDGP.2
         NX.EOS
         OUTRCM
         MDGP.1
         SSRCM
         MDGP.S
         SS.EX3
         SS.X3
112     FMC        REL           1115     5645   73/05/10.
         FMC
         SM.FMC
113     FMIP       REL            231     1144   73/05/10.
         FMIP
         FMI.EOF
         FMI.KX
         EXIT22
         EX22
         FMIEX22
114     FMOP       REL            125     5017   73/05/10.
         FMOP
         FMO.SEQ
         FMO.EX3
         FMO.E3
115     FSRTGET    REL            105     6021   73/05/10.
         FSRTGET
```

```
      CATALOG OF COBOL         FILE      1
REC   NAME      TYPE         LENGTH     CKSUM      DATE      COMMENTS

116   MRGCON    REL             477      2226   73/05/10.
       MRGCON
       ISIO
       ISION
       SM.MRGC
       S.MSVB7
117   DOC       REL             253      2272   73/05/10.
       DOC
118   MRGDSN    REL             144      4622   73/05/10.
       MRGDSN
       FMRGDSN
119   SRTCON    REL             277      4777   73/05/10.
       SRTCON
       SC.1ST
       S.1ST
       FBR.1
120   SRTPUT    REL              70      2257   73/05/10.
       SRT.PUT
121   SRTGET    REL             110      6356   73/05/10.
       SRT.GET
122   COBOL     OPLD            363      7732    73/06/12.

123   * EOF *        SUM =    44367
```

3.4.4 FORTRAN

```
      CATALOG OF FORTRAN          FILE       1
REC   NAME       TYPE           LENGTH     CKSUM      DATE      COMMENTS

  1   BACKSP=    REL                74      6255  73/06/06. BACKSPACE LOGICAL RECORD.
       BACKSP.
  2   BUFIN=     REL               130      3013  73/06/06.  BUFFER IN PROCESSOR.
       BUFIN.
  3   BUFIO=     REL               133      1200  73/06/06.  COMMON SETUP ROUTINE FOR  BUFIN*/BUFOUT=
       BUFIO.
  4   BUFOUT=    REL                62      7766  73/06/06.    BUFFER OUT PROCESSOR
       BUFOUT.
  5   BUSY       REL                23      4215  73/06/06. STATUS OF BUFFER I/O OPERATION
       BUSY
  6   CLOSMS     REL                36      0520  73/06/06. CLOSE RANDOM FILE LINKAGE ROUTINE.
       CLOSMS
  7   CONDIS     REL                63      7563  73/06/06. CONNECT/DISCONNECT A TERMINAL FILE.
       CONNEC
       DISCON
  8   DECODE=    REL               141      1430  73/06/06. FORMATTED READ FROM CORE.
       DECODI.
       DECODR.
  9   ENCODE=    REL               156      1455  73/06/06. FORMATTED WRITE INTO CORE.
       ENCODI.
       ENCODR.
 10   ENDFIL=    REL               103      7652  73/06/06. WRITE END OF LOGICAL FILE MARK.
       ENDFIL.
 11   EOF        REL                25      5234  73/06/06. TEST FOR END OF FILE STATUS.
       EOF
 12   FLTIN=     REL               226      2474  73/06/06.
       FEIEXP.
       FEIFSC.
 13   FLTOUT=    REL               434      1620  73/06/06. COMMON FLOATING OUTPUT CODE
       FEOFAL.
       FEOEXP.
       FEORND.
       FEOSCA.
       FEOZRO.
 14   FORSYS=    REL              1107      1736  73/06/06. FORTRAN OBJECT LIBRARY UTILITIES.
       CLSLNK.
       DAT.
       END.
       STOP.
       SYSEND.
       SYP=1
       SYP=2
       SYP=3
       AUTOFL.
       FLSCM.
       FLLCM.
       Q8NTRY.
       MEMIT.
       EXIT
       ABNORM.
       SYSARG=
       PACK.
       BURST.
       IOERR.
       SYSERR.
       SYS2=
      CATALOG OF FORTRAN         FILE      1
REC   NAME       TYPE        LENGTH     CKSUM       DATE       COMMENTS

       SYS1=
       CDD.
       COD.
       LINLIM.
       LOT.
       MSGAD.
       DBGFIT.
 15   FTNBIN     REL             11      4341   73/06/06. DUMMY ROUTINE FOR BINARY BLOCKING CALL.
       FTNBIN
 16   GETFIT=    REL             55      7437   73/06/06. LOCATE AN FIT GIVEN A FILE NAME.
       GETFIT.
       NAME.
 17   INCOM=     REL            232      4244   73/06/06. COMMON INPUT FORMATTING CODE
       FEIFSG.
       FEIFST.
       FEIGNC.
       FEINUM.
       FEISBL.
       FEIFST=
       FEIBLK=
       FEIBLK.
 18   INPB=      REL            356      7352   73/06/06.
       INPBI.
       INPBR.
 19   INPC=      REL            320      7742   73/06/06. FORMATTED READ FORTRAN RECORD.
       INPCI.
       INPCR.
 20   INPF=      REL            434      7606   73/06/06. LIST DIRECTED INPUT CONTROL
       FEIERR.
       INPFI.
       INPFR.
 21   IOCHEC     REL             23      5656   73/06/06. PARITY ERROR CHECK.
       IOCHEC
 22   IODATA=    REL             55      7365   73/06/06. COMMON INPUT/OUTPUT CONSTANTS
       IOCON.
 23   KODER=     REL           1651      3674   73/06/06. OUTPUT FORMAT INTERPRETER.
       KODER.
 24   KRAKER=    REL           1765      2416   73/06/06. FORMAT CRACKER FOR INPUT OPERATIONS.
       KRAKER.
       ERRSET
 25   LABEL      REL            232      2412   73/06/06. SET/CHECK USER TAPE LABELS.
       LABEL
 26   LDIN=      REL            360      6744   73/06/06. LIST DIRECTED INPUT FORMATTING
       FEIFAC.
       LDIN.
       LDINR.
 27   LDOUT=     REL            424      7474   73/06/06. LIST DIRECTED OUTPUT FORMATTING
       FEONTL.
       FEOREP.
       LDOUT.
       LDOUTW.
       LDOUTX.
 28   LENGTH     REL             26      4230   73/06/06. RETURN THE LENGTH OF A BUFFER IN.
       LENGTH
 29   NAMIN=     REL           1361      6433   73/06/06. NAMELIST INPUT ROUTINE.
       NAMIN.
      CATALOG OF FORTRAN        FILE       1
REC   NAME       TYPE          LENGTH    CKSUM      DATE      COMMENTS

 30   NAMOUT=    REL             674      1505   73/06/06. NAMELIST OUTPUT ROUTINE.
        NAMOUT.
 31   OPENMS     REL              45      3206   73/06/06. OPEN RANDOM FILE LINKAGE ROUTINE.
        OPENMS
 32   OUTB=      REL             302      2447   73/06/06. BINARY WRITE FORTRAN RECORD.
        OUTBI.
        OUTBR.
 33   OUTC=      REL             273      7067   73/06/06. FORMATTED WRITE FORTRAN RECORD.
        OUTCI.
        OUTCR.
 34   OUTCOM=    REL             234      1326   73/06/06. COMMON OUTPUT CODE
        FEOL.
        FEOI.
        FEOAFM.
        FEOBLS.
        FEOCNV.
        FEORIF.
        FEORIO.
        FEOSTR.
        FEOXFL.
        FEOZRS.
 35   OUTF=      REL             305      1770   73/06/06. LIST DIRECTED OUTPUT CONTROL
        OUTFI.
        OUTFR.
 36   RANMS=     REL             467      7263   73/06/06. RECORD INDEXED (MSIO) FILE PROCESSOR.
        CLOS.RI
        OPEN.RI
        READ.RI
        STIN.RI
        WRIT.RI
 37   READMS     REL              36      5160   73/06/06. READ RANDOM FILE LINKAGE ROUTINE.
        READMS
 38   REWIND=    REL              57      6252   73/06/06. POSITION FILE AT BEGINNING-OF-INFORMATION.
        REWIND.
 39   STINDX     REL              36      4530   73/06/06. CHANGE RANDOM FILE INDEX LINKAGE.
        STINDX
 40   SYSTEM     REL             520      2041   73/06/06. EXTENDED ERROR HANDLING OPTION.
        SYSTEM
        SYSTEMC
        SYSLNK.
 41   UNIT       REL              47      2566   73/06/06.  STATUS A BUFFER I/O FILE.
        UNIT
 42   WRITMS     REL              36      0043   73/06/06. WRITE RANDOM FILE LINKAGE ROUTINE.
        WRITMS
 43   XRCL       REL              25      2053   73/06/06. ISSUE RECALL REQUEST.
        XRCL
 44   CLOCK=     REL              60      3565   73/06/06. ACCESS SYSTEM CLOCKS FOR FORTRAN.
        CLOCK
        TIME
        DATE
        JDATE
        SECOND
 45   DISPLA     REL             222      2676   73/06/06. ISSUE MESSAGE AND VALUE TO DAYFILE.
        DISPLA
 46   DUMP       REL             161      2722   73/06/06. DUMP CORE FOR FORTRAN USER.
        DUMP
```

```
         CATALOG OF FORTRAN          FILE     1
 REC     NAME      TYPE          LENGTH     CKSUM        DATE       COMMENTS

          PDUMP
  47     GOTOER=   REL               27      5537   73/06/06. COMPUTED GO TO ERROR PROCESSOR.
          GOTOER.
  48     LEGVAR    REL               13      1223   73/06/06. TEST FOR INDEF OR INFINITE VALUE.
          LEGVAR
  49     MOVLEV    REL              106      4710   73/06/06. GENERALIZED STORAGE MOVE ROUTINE.
          MOVLEV
  50     OVERLAY   REL              216      2150   73/06/06. OVERLAY LOADING ROUTINE.
          OVERLAY
  51     PAUSE=    REL               37      3151   73/06/06. PAUSE AND ISSUE A MESSAGE TO CONSOLE.
          PAUSE.
  52     READEC    REL              112      5742   73/06/06. READ A BLOCK OF WORDS FROM ECS TO SCM.
          READEC
  53     REMARK    REL               15      0754   73/06/06. ISSUE A MESSAGE TO THE DAYFILE.
          REMARK
  54     SLITE     REL               67      4654   73/06/06. FORTRAN SENSE LIGHT MANIPULATION.
          SLITE
          SLITET
  55     SSWTCH    REL               40      5222   73/06/06. TEST THE SETTING OF A SENSE SWITCH.
          SSWTCH
  56     WRITEC    REL               46      7331   73/06/06. WRITE N WORDS FROM CM INTO ECS.
          WRITEC
  57     ACOS      REL               37      3231   73/06/06. ARCCOSINE OF X.
          ACOS
  58     ASIN      REL               37      6750   73/06/06. COMPUTE THE ARCSINE OF X
          ASIN
  59     ACOSIN=   REL              136      0633   73/06/06. COMPUTE ARCSINE OR ARCCOSINE.
          ACOS.
          ASIN.
  60     ALOG=     REL               54      0015   73/06/06. COMPUTE THE LOG AND LOG10 OF X.
          ALOG.
          ALOG10.
  61     ALOG      REL               37      5064   73/06/06. COMPUTE THE NATURAL LOGARITHM.
          ALOG
  62     ALOG10    REL               37      5134   73/06/06. COMPUTE LOGARITHM BASE 10 OF X.
          ALOG10
  63     ATAN      REL               32      1346   73/06/06. COMPUTE THE ARCTANGENT OF X
          ATAN
  64     ATAN=     REL               76      5554   73/06/06. ARCTANGENT OF X.
          ATAN.
  65     ATAN2     REL               44      4623   73/06/06. ARCTAN OF RATIO OF Y TO X
          ATAN2
  66     ATAN2=    REL              120      6560   73/06/06. ATAN OF THE RATIO OF Y TO X
          ATAN2.
  67     CABS      REL               50      5657   73/06/06. COMPLEX ABSOLUTE VALUE.
          CABS
  68     CABS=     REL               32      4540   73/06/06. COMPLEX ABSOLUTE VALUE
          CABS.
  69     CCOS      REL               72      0032   73/06/06. COMPLEX COSINE WITH ERROR CHECKING.
          CCOS
  70     CCOS=     REL               27      3516   73/06/06. COMPLEX COSINE OF Z.
          CCOS.
  71     CEXP      REL               64      6717   73/06/06. COMPLEX EXPONENTIAL OF Z
          CEXP
  72     CEXP=     REL               25      1461   73/06/06. COMPLEX EXPONENTIAL OF Z.
          CEXP.
```

```
        CATALOG OF FORTRAN          FILE      1
REC     NAME      TYPE            LENGTH     CKSUM      DATE      COMMENTS

 73     CLOG      REL                 40      0465  73/06/06. COMPLEX LOGARITHM OF Z.
          CLOG
 74     CLOG=     REL                 32      7221  73/06/06. COMPLEX LOGARITHM OF Z
          CLOG.
 75     COS       REL                 43      0213  73/06/06. COSINE OF X.
          COS
 76     COS=SIN   REL                 65      2305  73/06/06. COMPUTE SIN AND COS OF X
          COS.SIN
 77     CSIN      REL                 72      3254  73/06/06. COMPLEX SINE OF Z.
          CSIN
 78     CSIN=     REL                 25      7637  73/06/06. COMPLEX SINE OF Z.
          CSIN.
 79     CSQRT     REL                 37      7626  73/06/06. COMPLEX SQUARE ROOT OF Z.
          CSQRT
 80     CSQRT=    REL                 32      3251  73/06/06. COMPLEX SQUARE ROOT OF Z.
          CSQRT.
 81     DATAN     REL                 34      0512  73/06/06. DOUBLE ARCTANGENT.
          DATAN
 82     DATAN2    REL                 45      5040  73/06/06. COMPUTE THE DOUBLE ATAN OF Y/X.
          DATAN2
 83     DATAN=    REL                206      6206  73/06/06. COMPUTE DOUBLE ARCTANGENT.
          DATAN2.
          DATAN.
 84     DEXP      REL                 54      3422  73/06/06. DOUBLE EXPONENTIAL FUNCTION.
          DEXP
 85     DEXP=     REL                130      4473  73/06/06. DOUBLE EXPONENTIAL FUNCTION.
          DEXP.
 86     DLNLOG=   REL                177      1002  73/06/06. COMPUTE DOUBLE LOG AND DLOG10.
          DLOG.
          DLOG10.
 87     DLOG      REL                 40      3425  73/06/06. DOUBLE LOGARITHM BASE E OF X.
          DLOG
 88     DLOG10    REL                 40      2063  73/06/06. DOUBLE LOGARITHM BASE 10 OF X.
          DLOG10
 89     DMOD      REL                 53      5153  73/06/06. DOUBLE MODULUS OF D1, D2.
          DMOD
 90     DMOD=     REL                 35      2657  73/06/06. DOUBLE MODULUS OF D1,D2.
          DMOD.
 91     DCOS      REL                 45      2242  73/06/06. DOUBLE COSINE OF X.
          DCOS
 92     DSIN      REL                 45      3616  73/06/06. DOUBLE PRECISION SINE OF X.
          DSIN
 93     DSNCOS=   REL                175      1134  73/06/06. DOUBLE SINE AND COSINE ROUTINE.
          DSIN.
          DCOS.
 94     DSQRT     REL                 35      2113  73/06/06. DOUBLE SQUARE ROOT OF X.
          DSQRT
 95     DSQRT=    REL                 41      1602  73/06/06. DOUBLE PRECISION SQUARE ROOT.
          DSQRT.
 96     DTOD*     REL                103      6532  73/06/06. DOUBLE BASE TO DOUBSE POWER.
          DTOD$
 97     DTOD=     REL                 25      0151  73/06/06. DOUBLE BASE TO DOUBLE POWER.
          DTOD.
 98     DTOI*     REL                 52      7052  73/06/06. DOUBLE BASE TO INTEGER POWER.
          DTOI$
        CATALOG OF FORTRAN        FILE     1
REC     NAME      TYPE          LENGTH    CKSUM      DATE      COMMENTS

 99     DTOI=     REL               32     5665   73/06/06. DOUBLE TO INTEGER EXPONENTIATION.
          DTOI.
100     DTOX*     REL              103     0525   73/06/06. DOUBLE BASE TO REAL POWER.
          DTOX$
101     DTOX=     REL               24     3506   73/06/06. DOUBLE TO REAL EXPONENTIATION.
          DTOX.
102     DTOZ*     REL              125     6576   73/06/06. DOUBLE TO COMPLEX EXPONENTIATION.
          DTOZ$
103     DTOZ=     REL               33     1365   73/06/06. DOUBLE TO COMPLEX EXPONENTIATION.
          DTOZ.
104     EXP       REL               60     0207   73/06/06. EXPONENTIAL FUNCTION    -  E TO THE X.
          EXP
105     EXP=      REL               57     1464   73/06/06. EXPONENTIAL FUNCTION.
          EXP.
106     HYPERB=   REL               41     5011   73/06/06. HYPERBOLIC SINE AND COSINE.
          HYPERB.
107     ITOD*     REL              103     1557   73/06/06. INTEGER BASE TO DOUBLE POWER.
          ITOD$
108     ITOD=     REL               32     0660   73/06/06. INTEGER TO DOUBLE POWER.
          ITOD.
109     ITOJ*     REL               67     0371   73/06/06. INTEGER BASE TO INTEGER POWER.
          ITOJ$
110     ITOJ=     REL               27     7222   73/06/06. INTEGER TO INTEGER EXPONENTIATION.
          ITOJ.
111     ITOX*     REL              101     7255   73/06/06. INTEGER BASE TO REAL POWER.
          ITOX$
112     ITOX=     REL               24     3764   73/06/06. INTEGER TO A REAL POWER.
          ITOX.
113     ITOZ*     REL              121     1457   73/06/06. INTEGER BASE TO COMPLEX POWER.
          ITOZ$
114     ITOZ=     REL               15     7112   73/06/06. INTEGER TO COMPLEX EXPONENTIATION.
          ITOZ.
115     RANF      REL               26     3442   73/06/06. UNIFORM RANDOM NUMBER GENERATOR.
          RANF
          RANDOM.
          RANMLT.
          RANGET
116     RANSET    REL               21     0056   73/06/06. INITIALIZE THE RANDOM NUMBER SEED.
          RANSET
117     SIN       REL               43     6321   73/06/06. SINE OF X.
          SIN
118     SINCOS=   REL               72     0052   73/06/06. COMPUTE EITHER THE SINE OR COSINE OF X.
          SIN.
          COS.
119     SQRT      REL               33     7633   73/06/06. COMPUTE THE SQUARE ROOT OF X.
          SQRT
120     SQRT=     REL               33     1322   73/06/06. COMPUTE THE SQUARE ROOT OF X.
          SQRT.
121     TAN       REL               43     3274   73/06/06. TANGENT OF X.
          TAN
122     TAN=      REL              111     6445   73/06/06. TANGENT OF X.
          TAN.
123     TANH      REL               32     0774   73/06/06. HYPERBOLIC TANGENT OF X.
          TANH
124     TANH=     REL               50     5211   73/06/06. HYPERBOLIC TANGENT OF X.
          TANH.
        CATALOG OF FORTRAN        FILE     1
REC     NAME      TYPE          LENGTH    CKSUM     DATE      COMMENTS

125     XTOD*     REL              103     2057  73/06/06. REAL BASE TO DOUBLE POWER.
         XTOD$
126     XTOD=     REL               31     1322  73/06/06. REAL TO DOUBLE EXPONENTIATION.
         XTOD.
127     XTOI*     REL               47     0120  73/06/06. REAL BASE TO INTEGER POWER.
         XTOI$
128     XTOI=     REL               21     2225  73/06/06. REAL TO INTEGER EXPONENTIATION.
         XTOI.
129     XTOY*     REL              100     7435  73/06/06. REAL BASE TO REAL POWER.
         XTOY$
130     XTOY=     REL               23     0345  73/06/06. REAL TO REAL EXPONENTIATION.
         XTOY.
131     XTOZ*     REL              121     3360  73/06/06. REAL BASE TO COMPLEX POWER.
         XTOZ$
132     XTOZ=     REL               33     3604  73/06/06. REAL TO COMPLEX EXPONENTIATION.
         XTOZ.
133     ZTOI*     REL               52     6234  73/06/06. COMPLEX BASE TO INTEGER POWER.
         ZTOI$
134     ZTOI=     REL               31     0757  73/06/06. COMPLEX TO INTEGER EXPONENTIATION.
         ZTOI.
135     AND       REL               13     6551  73/06/06. COMPUTE LOGICAL PRODUCT OF ARGUMENTS.
         AND
136     COMPL     REL               11     6326  73/06/06. COMPLEMENT THE ARGUEMNT.
         COMPL
137     LOCF      REL               11     3552  73/06/06. ADDRESS OF ARGUMENT FUNCTION.
         LOCF
138     MASK      REL               33     6672  73/06/06. FORM A MASK.
         MASK
139     OR        REL               13     6501  73/06/06. LOGICAL SUM OF ARGUMENTS.
         OR
140     SHIFT     REL               12     5017  73/06/06. SHIFT ARG 1 BY ARG 2 BITS.
         SHIFT
141     XOR       REL               13     4374  73/06/06. LOGICAL DIFFERENCE OF ARGUMENTS.
         XOR
142     ABS       REL               14     1764  73/06/06. COMPUTE ABSOLUTE VALUE OF ARGUMENT.
         IABS
         ABS
143     AIMAG     REL               12     7444  73/06/06. RETURN IMAG PART OF COMPLEX NUMBER Z.
         AIMAG
144     AINT      REL               12     4534  73/06/06. TRUNCATE REAL NUMBER TO INTEGER.
         AINT
145     AMAX0     REL               16     1357  73/06/06. REAL MAXIMUM OF INTEGER ARGUMENTS.
         AMAX0
146     AMAX1     REL               15     5262  73/06/06. REAL MAXIMUM OF REAL ARGUMENTS.
         AMAX1
147     AMIN0     REL               16     7607  73/06/06. REAL MINIMUM OF INTEGER ARGUMENTS.
         AMIN0
148     AMIN1     REL               15     2704  73/06/06. REAL MINIMUM OF REAL ARGUMENTS.
         AMIN1
149     AMOD      REL               14     2740  73/06/06. COMPUTE THE REAL MODULUS OF ARGUMENTS.
         AMOD
150     CMPLX     REL               12     0440  73/06/06. CONVERT TWO REAL ARGUMENTS TO COMPLEX NUMBER.
         CMPLX
151     CONJG     REL               12     7045  73/06/06. COMPUTE THE COMPLEX CONJUGATE OF Z.
         CONJG
```

```
       CATALOG OF FORTRAN        FILE    1
REC    NAME      TYPE         LENGTH    CKSUM     DATE     COMMENTS

152    COUNT     REL              11     0547  73/06/06.
         COUNT
153    DABS      REL              13     4132  73/06/06. DOUBLE ABSOLUTE VALUE.
         DABS
154    DBLE      REL              12     5403  73/06/06. CONVERT SINGLE PRECISION ARGUMENT TO DOUBLE.
         DBLE
155    DIM       REL              14     1443  73/06/06. COMPUTE POSITIVE DIFFERENCE OF X AND Y.
         DIM
156    DMAX1     REL              22     1034  73/06/06. DOUBLE MAXIMUM OF DOUBLE ARGUMENTS.
         DMAX1
157    DMIN1     REL              22     5647  73/06/06. DOUBLE MINIMUM OF DOUBLE ARGUMENTS.
         DMIN1
158    DSIGN     REL              14     7733  73/06/06. SIGN OF ARGUMENT 2 TO ARGUMENT 1.
         DSIGN
159    FLOAT     REL              12     1136  73/06/06. INTEGER TO REAL CONVERSION.
         FLOAT
160    IDIM      REL              13     1754  73/06/06. INTEGER POSITIVE DIFFERENCE.
         IDIM
161    INT       REL              16     2516  73/06/06. TRUNCATE REAL ARGUMENT TO INTEGER.
         IFIX
         INT
         IDINT
162    ISIGN     REL              15     1564  73/06/06. TRANSFER SIGN OF ARGUMENT 1 TO ARGUMENT 2.
         SIGN
         ISIGN
163    MAX0      REL              15     2572  73/06/06. INTEGER MAXIMUM OF INTEGER ARGUMENTS.
         MAX0
164    MAX1      REL              16     5107  73/06/06. INTEGER MAXIMUM OF REAL ARGUMENTS.
         MAX1
165    MIN0      REL              15     4437  73/06/06. INTEGER MINIMUM OF INTEGER ARGUMENTS.
         MIN0
166    MIN1      REL              16     1611  73/06/06. INTEGER MINIMUM OF REAL ARGUMENTS.
         MIN1
167    MOD       REL              14     6047  73/06/06. INTEGER MODULUS OF J AND K.
         MOD
168    REAL      REL              13     3677  73/06/06. RETURN THE REAL PART OF COMPLEX NUMBER Z.
         SNGL
         REAL
169    BUGARR    REL             114     0656  73/06/06.
170    BUGCLL    REL             336     3165  73/06/06.
171    BUGCTL    REL              62     3175  73/06/06. DEBUG PRINT CONTROLLER.
         BUGCTL
172    BUGFUN    REL             412     5065  73/06/06.
173    BUGGTA    REL             117     7171  73/06/06.
174    BUGSTO    REL            1106     7662  73/06/06.
175    BUGTRC    REL             231     7622  73/06/06.
176    BUGTRT    REL              76     2335  73/06/06.
177    DBGFIT=   REL              27     5113  73/06/06. SUPPLY A FIT AND BUFFER FOR DEBUG FILE.
         DEBUG=
178    FTNERR=   REL              31     2357  73/06/06. FATAL COMPILATION ERROR PROCESSOR.
         FTNERR.
179    TRACEX    REL              36     0145  73/06/06. DEBUG TRACEBACK INFORMATION PROCESSOR.
         TRACEX
         TRCBKX
180    FORTRAN   OPLD            551     1126   73/06/08.
181    * EOF *       SUM =     34605
```

3.4.5 RUN2P3

```
       CATALOG  OF RUN2P3         FILE      1
REC    NAME      TYPE           LENGTH    CKSUM     DATE       COMMENTS

  1    CPC       REL               306     4157   72/10/10.
        CPC
        CPC02
        CPC03
        CPC04
        CPC999
  2    IO        REL               250     2552   72/10/10.
        IOREAD
        IOWRITE
        IOREWRT
        IOIO
        IOSAV
        IOZZ
        IOZW
  3    IORANDM   REL               262     2756   72/10/10.
        IORR
        IORW
        IORRW
  4    ACGOER    REL                26     5275   73/05/05.
        ACGOER
  5    DBLE      REL                13     3473   73/05/05.
        DBLE
  6    EXP       REL                77     5511   73/05/05.
        EXP
  7    IBAIEX    REL                45     6535   73/05/05.
        IBAIEX
  8    INITMS    REL               153     0003   73/05/05.
        OPENMS
        STINDX
  9    LEGVAR    REL                13     1033   73/05/05.
        LEGVAR
 10    LOCF      REL                14     3176   73/05/05.
        LOCF
        XLOCF
 11    SINCOS    REL               123     3651   73/05/05.
        SIN
        COS
 12    SNGL      REL                12     5041   73/05/05.
        SNGL
 13    SQRT      REL                61     1255   73/05/05.
        SQRT
 14    SYSTEM    REL              1236     6452   73/05/05.
        Q8NTRY
        SYSTEM
        SYSTEMC
        SYSTEMP
        END
        STOP
        EXIT
        ABNORML
 15    TAN       REL               133     4560   73/05/05.
        TAN
 16    XRCL      REL                13     5771   73/05/05.
        XRCL
 17    ALNLOG    REL               111     0443   73/05/05.
        ALOG
       CATALOG OF RUN2P3          FILE      1
REC    NAME       TYPE          LENGTH    CKSUM      DATE        COMMENTS

        ALOG10
 18    ASINCOS    REL              166     4010   73/05/05.
        ASIN
        ACOS
 19    ATAN       REL              114     0736   73/05/05.
        ATAN
 20    ATAN2      REL              135     4313   73/05/05.
        ATAN2
 21    CABS       REL               52     1644   73/05/05.
        CABS
 22    CBAIEX     REL               62     1615   73/05/05.
        CBAIEX
 23    CCOS       REL               71     3426   73/05/05.
        CCOS
 24    CEXP       REL               55     3204   73/05/05.
        CEXP
 25    CLOG       REL               52     7023   73/05/05.
        CLOG
 26    CSIN       REL               71     2232   73/05/05.
        CSIN
 27    CSQRT      REL               55     0101   73/05/05.
        CSQRT
 28    DABS       REL               27     4573   73/05/05.
        DABS
 29    DATAN      REL              236     2250   73/05/05.
        DATAN
        DATAN2
 30    DBADEX     REL              127     0302   73/05/05.
        DBADEX
        DBAREX
        RBADEX
 31    DBAIEX     REL               71     3460   73/05/05.
        DBAIEX
 32    DEXP       REL              165     6401   73/05/05.
        DEXP
 33    DISPLA     REL              311     5121   73/05/05.
        DISPLA
 34    DLNLOG     REL              232     7623   73/05/05.
        DLOG
        DLOG10
 35    DMOD       REL               64     1035   73/05/05.
        DMOD
 36    DSIGN      REL               34     5553   73/05/05.
        DSIGN
 37    DSINCOS    REL              227     5633   73/05/05.
        DSIN
        DCOS
 38    DSQRT      REL              100     1766   73/05/05.
        DSQRT
 39    DUMP       REL              151     3434   73/05/05.
        PDUMP
        DUMP
 40    DVCHK      REL               15     1444   73/05/05.
        DVCHK
 41    FTNBIN     REL               44     5342   73/05/05.
        FTNBIN
        CATALOG OF RUN2P3         FILE      1
REC     NAME      TYPE          LENGTH    CKSUM     DATE        COMMENTS

 42     IDINT     REL               41     3451   73/05/05.
         IDINT
 43     INPUTN    REL             1111     0572   73/05/05.
         INPUTN
 44     INPUTS    REL              104     4355   73/05/05.
         INPUTS
 45     LENGTH    REL               37     1004   73/05/05.
         LENGTH
 46     OUTPTN    REL              534     2350   73/05/05.
         OUTPTN
 47     OUTPTS    REL              111     2677   73/05/05.
         OUTPTS
 48     OVERFL    REL               14     5265   73/05/05.
         OVERFL
 49     OVERLAY   REL              155     4541   73/05/05.
         OVERLAY
 50     PAUSE     REL               35     3711   73/05/05.
         PAUSE
 51     RANF      REL               21     3562   73/05/05.
         RANF
 52     RBAIEX    REL               60     2057   73/05/05.
         RBAIEX
 53     RBAREX    REL               76     2456   73/05/05.
         RBAREX
 54     READEC    REL              111     5330   73/05/05.
         READEC
 55     READMS    REL              172     3322   73/05/05.
         READMS
 56     REMARK    REL               16     3343   73/05/05.
         REMARK
 57     BACKSP    REL              447     2553   73/05/05.
         BACKSP
 58     BUFFEI    REL              137     3077   73/05/05.
         BUFFEI
 59     BUFFEO    REL              135     1323   73/05/05.
         BUFFEO
 60     CONDIS    REL              113     2662   73/05/05.
         CONNEC
         DISCON
 61     ENDFIL    REL              107     3061   73/05/05.
         ENDFIL
 62     IOCHEC    REL               62     4452   73/05/05.
         IOCHEC
 63     IFENDF    REL              116     4431   73/05/05.
         IFENDF
 64     INPUTB    REL              336     4123   73/05/05.
         INPUTB
 65     INPUTC    REL              157     4342   73/05/05.
         INPUTC
 66     IOCHEK    REL              120     3312   73/05/05.
         IOCHEK
 67     KODER     REL             1364     3437   73/05/05.
         KODER
 68     KRAKER    REL             1135     3264   73/05/05.
         KRAKER
         CATALOG OF RUN2P3          FILE      1
  REC    NAME      TYPE          LENGTH     CKSUM      DATE      COMMENTS

   69    OUTPTB    REL              326      7432   73/05/05.
           OUTPTB
   70    OUTPTC    REL              124      6543   73/05/05.
           OUTPTC
   71    REWINM    REL              112      4177   73/05/05.
           REWINM
   72    SLITE     REL               27      3506   73/05/05.
           SLITE
   73    SLITET    REL               34      3213   73/05/05.
           SLITET
   74    SSWTCH    REL               31      7314   73/05/05.
           SSWTCH
   75    START     REL               16      6166   73/05/05.
           START
   76    SYS=TIM   REL               56      6701   73/05/05.
           CLOCK
           DATE
           JDATE
           SECOND
   77    TANH      REL               65      7457   73/05/05.
           TANH
   78    TIME      REL               34      6524   73/05/05.
           TIME
   79    WRITEC    REL               52      6654   73/05/05.
           WRITEC
   80    WRITMS    REL              133      2071   73/05/05.
           WRITMS
   81    RUNCALL   REL              315      5124   73/06/08.
           ACSSKF
           ACSSNF
           DLETEF
           FORCEF
           INSRTF
           OPNEWF
           OPNOLF
           RPLACF
           REPOSF
           SEEKLF
           SEEKSF
           STBKDF
           STBKIF
           STCOLF
           STERRF
           STKEYF
           STFETF
           TERMNF
   82    RUN2P3    OPLD             245      3053    73/06/08.

TERMINAL*IDLE.*        SUM =    21320
```

3.4.6 SIMLIB

```
        CATALOG OF SIMLIB              FILE     1
 REC    NAME       TYPE              LENGTH    CKSUM      DATE        COMMENTS

   1    SIMLIB     ULIB                 634     3134   73/06/12.
   2    XX$MAIN    REL                   64     3474  73/05/15.
         XX$MAIN
   3    XX$EVLT    REL                   14     1573  73/05/15.
         XX$XX
         XX$EXOG
         XX$ENOG
   4    XX$NIT     REL                  311     6323  73/05/15.
         XX$NIT
         XX$RNIT
   5    XX$SALL    REL                   32     6151  73/05/15.
         XXSALL
         XX$CORE
   6    XX$L       REL                   13     4534  73/05/15.
         XX$L
   7    XX$FILE    REL                  632     1040  73/05/15.
         XX$FILE
         WRITE.
   8    XX$NITR    REL                  424     5405  73/05/15.
         XX$ERRI
   9    XX$RXXD    REL                  101     5625  73/05/15.
         XX$RXXD
  10    XX$GARG    REL                   22     3105  73/05/15.
         XX$GARG
  11    XX$ABER    REL                  271     1203  73/05/15.
         XX$ABER
  12    XX$SYSV    REL                  141     4777  73/05/15.
         GTIME
         PTIME
         GHOURS
         PHOURS
         GMINS
         PMINS
         GDECHR
         GDPART
         GHPART
         GMPART
         GLINES
         GLXXNS
         GOTAPE
         GPAGE
         OTAPE
         PLINES
         PLXXNS
         POTAPE
         PPAGE
  13    XX$TMR     REL                  301     6112  73/05/15.
         XX$FLET
         XX$EEND
         XX$INT
         XX$STRT
  14    XX$CRDS    REL                  347     7560  73/05/15.
         XX$CMAS
         XX$DMAS
         XX$CSAT
         XX$DSAT
```

```
       CATALOG OF SIMLIB          FILE     1
 REC   NAME      TYPE          LENGTH    CKSUM      DATE      COMMENTS

  15   XX$IN     REL              261     2520   73/05/15.
        XX$IN
        XX$TAPE
  16   XX$ERRS   REL              633     5233   73/05/15.
        XX$ERRS
        XX$END
        XX$STOP
  17   XX$LKUP   REL               60     5352   73/05/15.
        XX$STEP
        XX$LNR
  18   XX$SWCH   REL               21     2777   73/05/15.
        XX$SWCH
  19   XX$KXX    REL               51     3056   73/05/15.
        KXX
        GKXX
        PKXX
  20   XX$LINK   REL               22     7537   73/05/15.
        XX$LINK
  21   XX$CNWC   REL              337     1231   73/05/15.
        XX$CNWA
        XX$CNWT
        XX$CNWO
        XX$CNWI
        XX$CNWH
        XX$CNWD
        XX$CNWM
  22   XX$CNRC   REL              251     6062   73/05/15.
        XX$CNRH
        XX$CNRD
        XX$CNRM
        XX$CNRI
        XX$CNRJ
        XX$CNRO
        XX$CNRA
        XX$CNR3
        XX$BUF
        DAT=
  23   XX$PUFP   REL              160     7001   73/05/15.
        XX$GFP
        XX$PFP
        XX$G2D
        XX$P2D
  24   XX$OUT    REL              226     0402   73/05/15.
        XX$OUT
        XX$EOL
  25   G$RAND    REL               54     5360   73/05/15.
        GRANDI
        GRANDM
        GRANDS
        GRANDR
        PRANDR
  26   XX$INIT   REL             3414     4340   73/05/15.
        XX$INIT
        XXMAXX
        XX$BYP
  27   XX$FMT    REL              307     0636   73/05/15.
        XX$FMT
  28   SIMLIB    OPLD              67     2362    73/06/12.

  29   * EOF *        SUM =    13031
```

3.4.7 SYSIO

```
        CATALOG OF SYSIO          FILE     1
REC     NAME      TYPE          LENGTH   CKSUM      DATE     COMMENTS

  1     CPUCPR    REL              173    4417   73/01/29. 72/11/17.          CONTROL POINT REPRIEVE PROCESSOR.
         SETUP.
         RECOVR
         CPR=
  2     UCLOAD    REL              254    5666   73/05/16. L73136 LOADER USER CALL INTERFACE ROUTINE.
         LOADER=
         LOADER
  3     SYS.RM    REL              100    6120   73/05/16. PROCESS SYSTEM REQUEST.
         MSG=
         RCL=
         SYS=
         WNB=
         CIO=
  4     CHEKPT    REL               62    2303   73/05/16.
         CHEKPTR
         CHEKPTX
         CHEKPT
  5     CPC       REL              306    4153   73/05/16.
         CPC
         CPC02
         CPC03
         CPC04
         CPC999
  6     IORANDM   REL              262    6756   73/05/16.
         IORR
         IORW
         IORRW
  7     IO        REL              250    2552   73/05/16.
         IOREAD
         IOWRITE
         IOREWRT
         IOIO
         IOSAV
         IOZZ
         IOZW
  8     GETBA     REL               30    2503   73/05/05.
         GETBA
  9     SIO$      REL             1424    0670   73/05/05.
         CIO1.
         RCL1.
         DAT.
         SIO.CTL
         INITL.
         SIO.
         SIO.END
         OPEN.
         RDPRU.
         BKSPRU.
         ADVIN.
         POSFI.
         MVWDS.
         SYSERR.
 10     LBUF.SQ   REL               45    5622   73/06/08.
         LBUF.SQ
 11     SQ.RM     REL              103    3055   73/06/08.
         SQ.RM
        CATALOG OF SYSIO            FILE     1
REC     NAME      TYPE          LENGTH     CKSUM      DATE        COMMENTS

 12     WA.RM     REL               31      0147   73/06/08.
         WA.RM
 13     IS.RM     REL              724      0035   73/06/08.
         IS.RM
         GET.IS
         GETN.IS
         PUT.IS
         SEEK.IS
         DLT.IS
         REPL.IS
         REW.IS
         SKFL.IS
         SKBL.IS
 14     DA.RM     REL              720      5447   73/06/08.
         DA.RM
         GET.DA
         GETN.DA
         PUT.DA
         SEEK.DA
         DLT.DA
         REPL.DA
         REW.DA
         SKFL.DA
         SKBL.DA
 15     AK.RM     REL              724      5536   73/06/08.
         AK.RM
         GET.AK
         GETN.AK
         PUT.AK
         SEEK.AK
         DLT.AK
         REPL.AK
         REW.AK
         SKFL.AK
         SKBL.AK
 16     CIO.RM    REL               61      6775   73/06/08.
         CIO.RM
         CIOR.RM
 17     ERR.RM    REL              345      7103   73/06/08.
         ERR.RM
         ERR1.RM
         ERR2.RM
 18     MOVE.RM   REL              103      3360   73/06/08.
         MOVE.RM
 19     CHWR.SQ   REL               22      3402   73/06/08.
         CHWR.SQ
 20     MCT.RM    REL               21      6346   73/06/08.
         MCT.RM
 21     MEM6RM    REL              730      3632   73/06/08.
         FIND.N
         NXT.ID
         STO.FIT
         CHK.ID
         R.ID
         R.FIT
         W.FIT
```

```
      CATALOG OF SYSIO           FILE      1


       W.ID
       REL.SP
       GET.FIT
       GET.SP
       GSP.ERR
 22   CHEK.RM    REL             21        0277   73/06/08.
       CHEK.RM
 23   OPEN.RM    REL            431        7404   73/06/08.
       OPEN.RM
 24   PDF.RM     REL            151        0006   73/06/08.
       PDF.RM
 25   OPEN.SQ    REL            361        5322   73/06/08.
       OPEN.SQ
       OPXX.SQ
 26   OPEX.SQ    REL            212        4524   73/06/08.
       OPEX.SQ
 27   OPEN.WA    REL            141        5477   73/06/08.
       OPEN.WA
 28   OPEN.IS    REL             56        2021   73/06/08.
       OPEN.IS
 29   OPEN.DA    REL             56        2264   73/06/08.
       OPEN.DA
 30   OPEN.AK    REL             56        7272   73/06/08.
       OPEN.AK
 31   PUT.RM     REL             32        4575   73/06/08.
       PUT.RM
       PUTP.RM
 32   RLEQ.RM    REL            133        7652   73/06/08.
       RLEQ.RM
 33   PUT.SQ     REL           1161        1457   73/06/08.
       PUT.SQ
 34   WAR.SQ     REL            302        3263   73/06/08.

       REPO.SQ
 35   PUT.WA     REL            570        3213   73/06/08.
       PUT.WA
       COMM.WA
       FLSH.WA
 36   CLSF.RM    REL             61        0446   73/06/08.
       CLSF.RM
 37   CLSF.SQ    REL            473        1301   73/06/08.
       CLSF.SQ
       FLSH.SQ
       RSPT.SQ
 38   FLSM.SQ    REL             32        0315   73/06/08.
       FLSM.SQ
 39   CLSV.RM    REL             36        2124   73/06/08.
       CLSV.RM
 40   CLSV.SQ    REL            221        7647   73/06/08.
       CLSV.SQ
 41   CLSF.WA    REL            131        2213   73/06/08.
       CLSF.WA
       REW.WA
 42   CLSV.WA    REL             64        5717   73/06/08.
       CLSV.WA
       CATALOG OF SYSIO             FILE      1
 REC   NAME       TYPE            LENGTH     CKSUM      DATE        COMMENTS

  43   CLSF.IS    REL                 47      2272   73/06/08.
         CLSF.IS
  44   CLSF.DA    REL                 47      5650   73/06/08.
         CLSF.DA
  45   CLSF.AK    REL                 47      7330   73/06/08.
         CLSF.AK
  46   REW.RM     REL                 30      5222   73/06/08.
         REW.RM
  47   REW.SQ     REL                 75      2414   73/06/08.
         REW.SQ
  48   GET.RM     REL                 32      3306   73/06/08.
         GET.RM
         GETP.RM
  49   GET.SQ     REL               1151      3545   73/06/08.
         GET.SQ
         SKGT.SQ
         DXIT.SQ
         AMAC.SQ
         GXIT.SQ
         AINI.SQ
         ANBL.SQ
         GRTJ.SQ
         GCNT.SQ
  50   Z.SQ       REL                147      1241   73/06/08.
         GET.Z
  51   R.SQ       REL                216      3177   73/06/08.
         GET.R
         SCR.SQ
  52   W.SQ       REL                113      6303   73/06/08.
         GET.W
  53   DT.SQ      REL                222      7615   73/06/08.
         GET.D
         GET.T
         DCDT.SQ
  54   FSU.SQ     REL                221      0615   73/06/08.
         GET.F
         GET.S
         GET.U
         RMU0.SQ
         RMU1.SQ
         RMU.SQ
  55   BTRT.SQ    REL                223      4424   73/06/08.
         BTRT.SQ
         PUT.I
         PUT.C
         PUT.K
         PUT.E
  56   GET.WA     REL                266      4664   73/06/08.
         GET.WA
  57   PUTL.SQ    REL                255      1575   73/06/08.
         PUTL.SQ
  58   GETL.SQ    REL                301      6540   73/06/08.
         GETL.SQ
         ERETURN
  59   LXER.SQ    REL                311      3442   73/06/08.
         LAB1.SQ
       CATALOG OF SYSIO               FILE      1
REC    NAME      TYPE               LENGTH    CKSUM      DATE      COMMENTS

         RNO.SQ
 60    ENDF.SQ    REL                  116     2003   73/06/08.
         ENDF.SQ
 61    WEOR.SQ    REL                  101     3066   73/06/08.
         WEOR.SQ
 62    WTMK.SQ    REL                   34     2462   73/06/08.
         WTMK.SQ
 63    DLT.RM     REL                   30     1540   73/06/08.
         DLT.RM
 64    DLT.SQ     REL                   30     3320   73/06/08.
         DLT.SQ
 65    DLT.WA     REL                   30     4531   73/06/08.
         DLT.WA
 66    SKFL.RM    REL                   30     5563   73/06/08.
         SKFL.RM
 67    SKFF.SQ    REL                   51     4564   73/06/08.
         SKFF.SQ
         SKBF.SQ
         SKFF.RM
         SKBF.RM
 68    SKFL.SQ    REL                  117     6510   73/06/08.
         SKFL.SQ
 69    SKFL.WA    REL                   40     7637   73/06/08.
         SKFL.WA
         SKFP.WA
         SKFF.WA
 70    GETN.RM    REL                   30     0420   73/06/08.
         GETN.RM
 71    GETN.SQ    REL                   34     2431   73/06/08.
         GETN.SQ
 72    GETN.WA    REL                   34     4471   73/06/08.
         GETN.WA
 73    REPL.RM    REL                   37     1627   73/06/08.
         REPL.RM
 74    REPL.SQ    REL                   34     0140   73/06/08.
         REPL.SQ
 75    REPL.WA    REL                   34     2200   73/06/08.
         REPL.WA
 76    SEEK.RM    REL                   30     6261   73/06/08.
         SEEK.RM
 77    SEEK.SQ    REL                   34     5306   73/06/08.
         SEEK.SQ
 78    SEEK.WA    REL                   30     2436   73/06/08.
         SEEK.WA
 79    SKBL.RM    REL                   30     5443   73/06/08.
         SKBL.RM
 80    SKBL.SQ    REL                 1077     4262   73/06/08.
         SKBL.SQ
         SKBP.SQ
         SKBP.RM
 81    SKFP.SQ    REL                   20     3177   73/06/08.
         SKFP.SQ
         SKFP.RM
 82    SKBL.WA    REL                   34     4005   73/06/08.
         SKBL.WA
       CATALOG OF SYSIO          FILE      1
REC    NAME      TYPE           LENGTH    CKSUM     DATE       COMMENTS

 83    MSG.RM    REL              102      4343  73/06/08.
        MSGLINL
        MSGLINE
 84    R6RM.RM   REL              230      7547  73/06/08.
        R.6RM
        EPW
 85    ACOMMON   REL               15      7044  73/06/08.
        CPYRITE
 86    BUFALOC   REL              412      3610  73/06/08.
        BUFALOC
 87    DATM      REL             2566      0612  73/06/08.
        SISPUT
        PUTDUMY
        SISGET
        SISDELT
        SISREPL
        CKFILE
        MJKYCK
        CHECKEY
 88    DIAGNOS   REL              521      4546  73/06/08.
        DIAGNOS
        BINDIS
 89    FINDIT    REL              443      3536  73/06/08.
        FINDIT
        KYCMPAR
        MOVEKEY
        TRNSLAT
        REVERS
 90    INDEXM    REL             1130      4732  73/06/08.
        IDXINS
        RPOSX
        IDXRPL
        IDXDEL
        RECLMI
 91    IOMGR     REL              772      2247  73/06/08.
        SI$IO
 92    PPCALL    REL              410      1575  73/06/08.
        PPCALL(
 93    CEOI6RM   REL               26      3435  73/06/08.
        CEOI6RM
 94    WRIT6RM   REL               43      1616  73/06/08.
        WRIT6RM
 95    REGSAV    REL              111      3034  73/06/08.
        SAVREG
        RESREG
        SAVREGM
        RESREGM
 96    SAAM.IS   REL              166      2111  73/06/08.
        IS.RTRN
        SAAM.IS
 97    SISCLSE   REL              536      2124  73/06/08.
        SISCLSE
        SISCLOS
 98    SISKEY    REL              106      4144  73/06/08.
        SISKEY
       CATALOG OF SYSIO            FILE      1
REC    NAME      TYPE            LENGTH    CKSUM      DATE        COMMENTS

 99    SISOPEN   REL               1434     2447   73/06/08.
        SISOPEN
        SISRWIN
100    SISRPV    REL                150     6710   73/06/08.
        SISRPV
        DIAGRTN
101    SISSEEK   REL                141     0130   73/06/08.
        SISSEEK
102    SISSKIP   REL                351     1047   73/06/08.
        SISSKIP
103    SIZES     REL                350     7673   73/06/08.
        SIZES(
104    KWAKER    REL                324     6705   73/06/08.
        KWAKER
        KWAKER2
        FILLUP(
        BOMB
105    GETPAR    REL                201     4621   73/06/08.
        GETPAR
        GETFPAR
106    NEWCONS   REL                 72     4762   73/06/08.
        NEWCONS
        GETLFN
        NUMPM
107    FTNMAIN   REL                173     4103   73/06/08.
108    READCRD   REL                374     4232   73/06/08.
109    PRINTER   REL                470     5362   73/06/08.
110    NRER1     REL                631     3571   73/06/08.
111    DEFAULT   REL                231     6147   73/06/08.
112    ACCESS    REL                130     3671   73/06/08.
        ACCESSK
        ACCESSN
        CKFTERR
113    ACCPRCK   REL                270     3044   73/06/08.
        MASKER2
        AXEPIK(
        OLDCMP1
        RETRNRL
        RETRNES
114    CHKFET    REL                240     7727   73/06/08.
        CHKFET(
        V1FATAL
115    DELETE    REL                 53     1632   73/06/08.
        DELETE
116    FORCEW    REL                 64     7631   73/06/08.
        FORCEW
117    INSERT    REL                 62     2770   73/06/08.
        INSERT
118    FTNCALL   REL                363     5250   73/06/08.
        ACSSKF
        ACSSKF$
        ACSSNF
        ACSSNF$
        DLETEF
        DLETEF$
        FORCEF
       CATALOG OF SYSIO            FILE     1
REC    NAME      TYPE            LENGTH     CKSUM      DATE        COMMENTS

         FORCEF$
         INSRTF
         INSRTF$
         OPNEWF
         OPNEWF$
         OPNOLF
         OPNOLF$
         RPLACF
         RPLACF$
         REPOSF
         REPOSF$
         SEEKLF
         SEEKLF$
         SEEKSF
         SEEKSF$
         STBKDF
         STBKDF$
         STBKIF
         STBKIF$
         STCOLF
         STCOLF$
         STERRF
         STERRF$
         STFETF
         STKEYF
         TERMNF
         TERMNF$
 119   OPENNEW   REL               563      3561    73/06/08.
         OPENNEW
         BILDFIT
 120   OPENOLD   REL               267      5704    73/06/08.
         OPENOLD
 121   REPLACE   REL                62      4551    73/06/08.
         REPLACE
 122   REPOS     REL               103      1332    73/06/08.
         REPOS
 123   SEEK      REL                76      5420    73/06/08.
         SEEKL
         SEEKS
 124   SETBLKD   REL               126      0523    73/06/08.
         SETBLKD
 125   SETBLKI   REL               142      3733    73/06/08.
         SETBLKI
 126   SETCOLL   REL               110      4655    73/06/08.
         SETCOLL
 127   SETERR    REL               200      2031    73/06/08.
         SETERR
 128   SETKEY    REL               211      5771    73/06/08.
         SETKEY
 129   STFETF$   REL               116      0451    73/06/08.
         STFETF$
 130   STKEYF$   REL               203      1200    73/06/08.
         STKEYF$
 131   TERMNAT   REL                63      6214    73/06/08.
         TERMNAT
```

```
      CATALOG OF SYSIO          FILE      1
REC   NAME      TYPE          LENGTH     CKSUM     DATE        COMMENTS

132   BINDSDA   REL               61      6114   73/06/08.
        SD$BIND
133   CLOSE     REL              414      5471   73/06/08.
        SD$CLOS
134   SD$DEL    REL              112      2413   73/06/08.
        SD$DLTE
135   D.DIAGN   REL              367      6752   73/06/08.
        SD$DIAG
136   SD$FIND   REL              336      1453   73/06/08.
        SD$FIND
        SD$FRSK
        SD$DVID
        SD$ALKY
        SD$FTPR
137   SD$GETN   REL              164      2171   73/06/08.
        SD$GETN
        SD$REW
138   SDAHASH   REL               24      4306   73/06/08.
        SD$HASH
139   SD$ADD    REL              126      7671   73/06/08.
        SD$ADD
140   D.INREC   REL               64      6267   73/06/08.
        SD$INRC
141   SD$IO     REL              411      5340   73/06/08.
        SD$CBLK
        SD$CKSM
        SD$EMTY
        SD$EXIO
        SD$FLSH
        SD$READ
        SD$WRIT
        SD$CEOI
        SD$FSCK
        SD$WFST
142   SD$KEY    REL               53      0447   73/06/08.
        SD$KEY
143   RECHOM    REL               40      0744   73/06/08.
        SD$RCHM
144   D.OPN     REL              710      1757   73/06/08.
        SD$OFFT
        SD$OPEN
        SD$MPLY
145   OVFM      REL              327      5174   73/06/08.
        SD$OVFM
        SD$SET
146   SD$RPLC   REL              261      3754   73/06/08.
        SD$RPLC
147   RETRV     REL               76      0437   73/06/08.
        SD$RETV
148   SD$RLCV   REL               32      1057   73/06/08.
        SD$RLIN
        SD$RLOT

        SD$RPVD
150   SAAM.DA   REL              127      6417   73/06/08.
        SAAM.DA
        CATALOG OF SYSIO           FILE      1
REC     NAME      TYPE           LENGTH    CKSUM      DATE         COMMENTS

          SD$SVB6
151     SD$SEEK   REL                42     5351   73/06/08.
          SD$SEEK
152     SPACE     REL                26     0104   73/06/08.
          SD$SPAC
153     SQUEEZE   REL                54     0770   73/06/08.
          SD$SQZ1
          SD$SQUZ
154     SEARCH    REL               136     4061   73/06/08.
          SD$SRCH
          SD$TALY
155     SD$TRC    REL               321     0442   73/06/08.
          SD$TRC
          SD$TCLS
          SD$OCT
156     FILEDA    REL                20     0773   73/06/08.
          FILEDA
157     FILEIS    REL                27     0466   73/06/08.
          FILEIS
158     FILESQ    REL                20     2162   73/06/08.
          FILESQ
159     FILEWA    REL                20     5216   73/06/08.
          FILEWA
160     FILEAK    REL                20     7753   73/06/08.
          FILEAK
161     FITCOM    REL               555     2673   73/06/08.
          STOREF
          IFETCH
          SETFIT
          SYMSRCH
162     ERRPROC   REL               155     6113   73/06/08.
          ERRPROC
          ERR1
163     OPNCLS    REL               106     7515   73/06/08.
          CLOSEM
          OPENM
164     GET       REL                72     5562   73/06/08.
          GET
165     PUT       REL               101     2376   73/06/08.
          PUT
166     REPLC     REL                62     0740   73/06/08.
          REPLC
167     DLTE      REL                63     2704   73/06/08.
          DLTE
168     GETN      REL                43     4503   73/06/08.
          GETN
169     SEEKF     REL                50     4443   73/06/08.
          SEEKF
170     SKIP      REL                46     2311   73/06/08.
          SKIP
171     REWND     REL                31     1103   73/06/08.
          REWND
172     GETP      REL                57     7256   73/06/08.
          GETP
173     PUTP      REL                50     6117   73/06/08.
          PUTP
```

```
       CATALOG OF SYSIO           FILE     1
REC    NAME      TYPE          LENGTH    CKSUM      DATE       COMMENTS

174    SQANDWA   REL              127     1663   73/06/08.
        WEOR
        WTMK
        ENDFILE
        CHECK
175    KYAN      REL             1730     2573   73/06/08.
        SDAHASH
        SD$KEYH
        SDAKYAN
        SDAKEYH
        SDAENDH
        SD$KYAN
        SD$ENDH
176    DCREATE   REL              544     4431   73/06/08.
        SDACRT
        SDACRTU
        SDAENDC
177    SYSIO     OPLD             541     1515    73/06/12.

178    * EOF *       SUM =      64047
```

3.4.8 SYSLIB

```
      CATALOG OF SYSLIB   FILE           1                         73/05/15. 01.08.13.      PAGE      1
REC   NAME      TYPE        LENGTH     CKSUM     DATE     COMMENTS

  1   SYSLIB    ULIB          1301      1407   73/05/15.
  2   LOADER    REL             65      3014   73/05/25.
       LOADER
  3   LDRUSX    REL            103      4622   73/05/25.
       LDRUSX=
  4   CPUCIO    REL             27      3602   73/05/25. 70/09/21.           I/O FUNCTION PROCESSOR.
       CIO=
  5   CPUCPM    REL             16      7710   73/05/25. 70/09/21.           CONTROL POINT MANAGER PROCESSOR.
       CPM=
  6   CPULFM    REL             24      7146   73/05/25. 70/09/21.           LOCAL FILE MANAGER PROCESSOR.
       LFM=
  7   CPUMVE    REL             31      3366   73/05/25. 70/09/21.           MOVE BLOCK OF DATA.
       MVE=
  8   CPUPFM    REL             24      0467   73/05/25. 70/09/21.           PERMANENT FILE PROCESSOR.
       PFM=
  9   CPUOPE    REL             25      4250   73/05/25. 70/09/21.           OPEN FILE PROCESSOR.
       OPE=
 10   CPUOVL    REL             24      4302   73/05/25. 70/09/21.           OVERLAY LOAD PROCESSOR.
       OVL=
 11   CPURDC    REL             27      3245   73/05/25. 70/09/21.           READ CODED LINE, -C- FORMAT.
       RDC=
 12   CPURDH    REL             50      2576   73/05/25. 70/09/21.           READ CODED LINE, -H- FORMAT.
       RDH=
 13   CPURDO    REL             40      1643   73/05/25. 70/09/21.           READ ONE WORD.
       RDO=
 14   CPURDS    REL             62      6450   73/05/25. 70/09/21.           READ CODED LINE TO STRING BUFFER.
       RDS=
 15   CPURDW    REL            141      6433   73/05/25. 70/09/21.           READ WORDS TO WORKING BUFFER.
       LCB=
       RDW=
       RDX=
 16   CPUSYS    REL             51      4031   73/05/25. 70/09/21.           PROCESS SYSTEM REQUEST.
       MSG=
       RCL=
       SYS=
       WNB=
 17   CPUWTC    REL             26      7624   73/05/25. 70/09/21.           WRITE CODED LINE, -C- FORMAT.
       WTC=
 18   CPUWTH    REL             46      7364   73/05/25. 70/09/21.           WRITE CODED LINE, -H- FORMAT.
       WTH=
 19   CPUWTO    REL             33      2536   73/05/25. 70/09/21.           WRITE ONE WORD.
       WTO=
 20   CPUWTS    REL             63      1725   73/05/25. 70/09/21.           WRITE CODED LINE FROM STRING BUFFER.
       WTS=
 21   CPUWTW    REL            115      4662   73/05/25. 70/09/21.           WRITE WORDS FROM WORKING BUFFER.
       DCB=
       WTW=
       WTX=
 22   ACGOER    REL             27      7234   73/05/25. 70/06/17.           RUN - *GO TO* ERROR PROCESSOR.
       ACGOER
 23   ALNLOG    REL            111      0443   72/05/14.
       ALOG
       ALOG10
 24   EXP       REL             77      5511   72/05/14.
      EXP
      CATALOG OF SYSLIB          FILE      1                          73/05/15. 01.08.13.      PAGE      2
REC   NAME       TYPE          LENGTH    CKSUM      DATE      COMMENTS

 25   INPUTC     REL              123     4325    73/05/25. 70/06/21. 71/03/01. RUN - CODED INPUT.
        INPUTC
 26   KODER      REL             1301     3755    73/05/25. 01/16/70. 73/10/18. RUN - CODED DATA ENCODE ROUTINE.
        KODER
 27   KRAKER     REL             1123     3671    73/05/25. 70/06/21. 73/10/18. RUN - CODED DATA DECODE ROUTINE.
        KRAKER
        RUNDCF=
 28   OUTPTC     REL              106     7065    73/05/25. 70/06/21. 72/05/19. RUN - CODED OUTPUT.
        OUTPTC
 29   RBAIEX     REL               60     2057    72/05/14.
        RBAIEX
 30   RBAREX     REL               76     2456    72/05/14.
        RBAREX
 31   RUNIOP     REL              133     4136    73/05/25. 70/06/22. 70/09/29. RUN - INPUT/OUTPUT PROCESSOR.
        RUNIOE=
        RUNSBF=
        RUNSFP=
        RUNSFS=
 32   RUNSYS     REL              420     2512    73/05/25. 70/06/28. 73/05/23. RUN - OBJECT TIME UTILITY ROUTINES.
        END
        EXIT
        STOP
        Q8NTRY
        RUNABT=
        RUNPTR=
        ABNORML
 33   SQRT       REL               61     1255    72/05/14.
        SQRT
 34   SYSTEM     REL              405     3110    73/05/25. 70/06/17. 73/10/18. RUN - EXECUTION ERROR PROCESSOR.
        SYSTEM
        SYSTEMC
        SYSTEMP
        RUNLES=
 35   ABORT      REL               26     2054    73/05/25. 70/06/17.           RUN - ABORT PROGRAM.
        ABORT
 36   ASINCOS    REL              166     4010    72/05/14.
        ASIN
        ACOS
 37   ATAN       REL              114     0736    72/05/14.
        ATAN
 38   ATAN2      REL              135     4313    72/05/14.
        ATAN2
 39   BACKSP     REL              113     1652    73/05/25. 70/06/28. 73/10/18. RUN - BACKSPACE FILE.
        BACKSP
 40   BUFFEI     REL              107     6213    73/05/25. 70/06/17.           RUN - BUFFER IN FILE.
        BUFFEI
 41   BUFFEO     REL               65     4721    73/05/25. 70/06/28. 70/08/12. RUN - BUFFER OUT FILE.
        BUFFEO
 42   CABS       REL               52     1644    72/05/14.
        CABS
 43   CBAIEX     REL               62     1615    72/05/14.
        CBAIEX
 44   CCOS       REL               71     3426    72/05/14.
        CCOS
 45   CEXP       REL               55     3204    72/05/14.
        CEXP
        CATALOG OF SYSLIB         FILE     1                         73/05/15. 01.08.13.       PAGE      3
REC     NAME        TYPE         LENGTH     CKSUM      DATE      COMMENTS

 46     CLOCK       REL             20      0507     73/05/25. 70/06/17.          RUN - REQUEST TIME OF DAY.
          CLOCK
 47     CLOG        REL             52      7023     72/05/14.
          CLOG
 48     CSIN        REL             71      2232     72/05/14.
          CSIN
 49     CSQRT       REL             55      0101     72/05/14.
          CSQRT
 50     DABS        REL             27      4573     72/05/14.
          DABS
 51     DATER       REL             20      6352     73/05/25. 70/06/17.          RUN - REQUEST DATE.
          DATER
 52     DATAN       REL            236      2250     72/05/14.
          DATAN
          DATAN2
 53     DBADEX      REL            127      0302     72/05/14.
          DBADEX
          DBAREX
          RBADEX
 54     DBAIEX      REL             71      3460     72/05/14.
          DBAIEX
 55     DBLE        REL             13      3473     72/05/14.
          DBLE
 56     DEXP        REL            165      6401     72/05/14.
          DEXP
 57     DISPLA      REL            307      3014     72/05/14.
          DISPLA
 58     DLNLOG      REL            232      7623     72/05/14.
          DLOG
          DLOG10
 59     DMOD        REL             64      1035     72/05/14.
          DMOD
 60     DSIGN       REL             34      5553     72/05/14.
          DSIGN
 61     DSINCOS     REL            227      5633     72/05/14.
          DSIN
          DCOS
 62     DSQRT       REL            100      1766     72/05/14.
          DSQRT
 63     DUMP        REL            151      3434     72/05/14.
          PDUMP
          DUMP
 64     DVCHK       REL             15      1444     72/05/14.
          DVCHK
 65     ENDFIL      REL             27      3273     73/05/25. 70/06/17.          RUN - WRITE FILE MARK.
          ENDFIL
 66     IBAIEX      REL             45      6535     72/05/14.
          IBAIEX
 67     ICOUNT      REL             15      7657     73/05/25. 70/06/17.          RUN - COUNT FUNCTION.
          ICOUNT
 68     IDINT       REL             41      3451     72/05/14.
          IDINT
 69     IFENDF      REL             22      5215     73/05/25. 70/06/17.          RUN - END OF FILE CHECK.
          IFENDF
 70     INPUTB      REL            116      1430     73/05/25. 70/06/21. 72/05/19. RUN - BINARY INPUT.
          INPUTB
```

```
       CATALOG OF SYSLIB         FILE    1                           73/05/15. 01.08.13.      PAGE      4
REC    NAME      TYPE          LENGTH   CKSUM     DATE     COMMENTS

 71    INPUTN    REL             1142    5204   73/05/25. 70/06/21. 73/10/18. RUN - NAMELIST INPUT.
        INPUTN
 72    INPUTS    REL              104    3777   73/05/25. 70/06/17.           RUN - DECODE.
        INPUTS
 73    IOCHEK    REL               40    4126   73/05/25. 70/06/28. 70/08/12. RUN - CHECK BUFFER FILE STATUS.
        IOCHEK
 74    ISHIFT    REL               15    0052   73/05/25. 70/06/17.           RUN - SHIFT VARIBLE.
        ISHIFT
 75    LEGVAR    REL               13    1033   72/05/14.
        LEGVAR
 76    LENGTH    REL               21    0317   73/05/25. 70/06/17. 70/08/12. RUN - RETURN BUFFER FILE LENGTH.
        LENGTH
 77    LOCF      REL               14    3176   72/05/14.
        LOCF
        XLOCF
 78    OUTPTB    REL               55    1546   73/05/25. 70/06/17. 70/08/12. RUN - BINARY OUTPUT.
        OUTPTB
 79    OUTPTN    REL              550    4752   73/05/25. 70/06/21. 71/01/27. RUN - NAMELIST OUTPUT.
        OUTPTN
 80    OUTPTS    REL              106    5147   73/05/25. 70/06/17. 70/08/12. RUN - ENCODE.
        OUTPTS
 81    OVERFL    REL               14    5265   71/01/14.
        OVERFL
 82    OVERLAY   REL               65    3662   73/05/25.
        OVERLAY
        OVERLIB
 83    PAUSE     REL              3EL              30       6774   73/05/25. 70/06/21. 70/09/29. RUN - PAUSE FOR OPERATOR.
        PAUSE
 84    RANF      REL               21    3562   72/05/14.
        RANF
 85    READB     REL               32    0566   73/05/25. 70/06/21.           RUN - READ WORDS.
        READB
 86    RECALL    REL               16    6721   73/05/25. 70/06/17.           RUN - PLACE PROGRAM ON RECALL.
        RECALL
 87    REMARK    REL               17    4646   73/05/25. 70/06/17.           RUN - ENTER MESSAGE IN DAYFILE.
        REMARK
 88    REWINM    REL               32    3347   73/05/25. 70/06/21.           RUN - REWIND FILE.
        REWINM
 89    RTIME.    REL               17    1124   73/05/25.
        RTIME
 90    SECOND    REL               27    7614   73/05/25. 70/06/17.           RUN - REQUEST CPU TIME USED.
        SECOND
 91    SINCOS    REL              123    3651   72/05/14.
        SIN
        COS
 92    SLITE     REL               32    1137   73/05/25. 70/06/21.           RUN - TURN ON SENSE LIGHT.
        SLITE
 93    SLITET    REL               33    3727   73/05/25. 70/06/21. 70/09/29. RUN - TEST SENSE LIGHT.
        SLITET
 94    SNGL      REL               12    5041   72/05/14.
        SNGL
 95    SSWTCH    REL               32    3405   73/05/25. 70/06/21.           RUN - TEST SENSE SWITCH.
        SSWTCH
 96    START     REL               22    2672   73/05/25. 70/06/17.           RUN - SEND START MESSAGE.
        START
```

```
         CATALOG OF SYSLIB         FILE     1                              73/05/15. 01.08.13.       PAGE       5
REC      NAME        TYPE        LENGTH      CKSUM      DATE     COMMENTS
 97      TANH        REL             65       7457   72/05/14.
           TANH
 98      TIME        REL             25       7405   73/05/25. 70/06/21.           RUN - SEND TIME MESSAGE.
           TIME
 99      TAN         REL            133       4560   72/05/14.
           TAN
100      WRITEB      REL             32       3265   73/05/25. 70/06/21.           RUN - WRITE WORDS.
           WRITEB
101      SYSLIB      OPLD           311       5450   73/05/15.

102      * EOF *        SUM =     20015
```

3.4.9 SYSMISC

```
      CATALOG OF SYSMISC        FILE     1
REC   NAME      TYPE        LENGTH     CKSUM     DATE     COMMENTS

  1   SYSMISC   ULIB          2136      1360   73/06/12.
  2   ALGORUN   REL           3314      7631  73/05/09.
       ALGORUN
  3   ALGLB00   REL           3325      6420  73/05/09.
       ALGLB00
  4   ALGLB01   REL            720      4541  73/05/09.
       ALGLB01
  5   ALGLB02   REL           5640      7422  73/05/09.
       ALGLB02
  6   ALGLB03   REL            776      1715  73/05/09.
       ALGLB03
  7   ALGLB04   REL            621      0772  73/05/09.
       ALGLB04
  8   ALGLB05   REL           1023      4664  73/05/09.
       ALGLB05
  9   ALGLB06   REL           1154      2156  73/05/09.
       ALGLB06
 10   ALGLB07   REL            621      1006  73/05/09.
       ALGLB07
 11   ALGLB10   REL            154      4457  73/05/09.
       ALGLB10
 12   SIMURUN   REL           7306      1446  73/05/09.
       SIMURUN
 13   SIMLB00   REL           3072      7765  73/05/09.
       SIMLB00
 14   SIMLB01   REL           2036      4332  73/05/09.
       SIMLB01
 15   SIMLB02   REL           1514      5114  73/05/09.
       SIMLB02
 16   SIMLB03   REL           3647      5404  73/05/09.
       SIMLB03
 17   SIMLB05   REL            671      5670  73/05/09.
       SIMLB05
 18   SYSTEM$   REL           1206      5751   03/08/73
       FLSCM.
       FLLCM.
       Q8NTRY.
       END.
       EXIT$
       STOP.
       ABNORM.
       SYSTEME
       SYSTEM$
       SYSTEM.
       SYSTEM:
       SYS1:
       SYS2:
       LOT:
       ERRFLG.
       DBGFET.
 19   ACGOER$   REL             27      3032   03/08/73
       ACGOER.
 20   BACKSP$   REL            423      3557   03/08/73
       BACKSP.
 21   BUFFEI$   REL            127      0070   03/08/73
       BUFFEI.
```

```
        CATALOG OF SYSMISC          FILE     1
 REC    NAME      TYPE            LENGTH   CKSUM      DATE       COMMENTS

  22    BUFFEO$   REL                124    4041    03/08/73
         BUFFEO.
  23    BUGARR$   REL                117    7241    03/08/73
         BUGARR$
  24    BUGCLL$   REL                405    4420    03/08/73
         BUGCLL$
         BUGCLR$
         STRACE$
  25    BUGCTL$   REL                 56    0700    03/08/73
         BUGCTL$
  26    BUGFUN$   REL                545    7663    03/08/73
         BUGFUN$
         BUGFNN$
  27    BUGGTA$   REL                153    2204    03/08/73
         BUGGTA$
  28    BUGSTO$   REL               1226    2767    03/08/73
         BUGSTO$
  29    BUGTRC$   REL                311    4725    03/08/73
         BUGTRC$
         BUGTRA$
         BUGTRU$
  30    BUGTRT$   REL                 75    2533    03/08/73
         BUGTRT$
  31    CONDIS$   REL                110    0101    03/08/73
         CONNEC$
         DISCON$
  32    DBGFET$   REL                 22    1241    03/08/73
         DEBUG=
  33    DUMP$     REL                116    7562    03/08/73
         DUMP$
         PDUMP$
  34    ENDFIL$   REL                106    2174    03/08/73
         ENDFIL.
  35    FTNBIN$   REL                 46    6747    03/08/73
         FTNBIN$
  36    FTNERR$   REL                 31    2622    03/08/73
         FTNERR.
  37    IFENDF$   REL                110    6470    03/08/73
         EOF$
  38    INITMS$   REL                154    1605    03/08/73
         OPENMS$
         STINDX$
  39    INPUTB$   REL                344    3725    03/08/73
         IPUTBI.
         INPUTB.
  40    INPUTC$   REL                177    5511    03/08/73
         IPUTCI.
         INPUTC.
  41    INPUTN$   REL               1263    1744    03/08/73
         INPUTN.
  42    INPUTS$   REL                103    6240    03/08/73
         DECODI.
         DECODE.
  43    IOCHEC$   REL                 60    3454    03/08/73
         IOCHEC$
        CATALOG OF SYSMISC          FILE      1
REC     NAME     TYPE          LENGTH      CKSUM      DATE        COMMENTS

 44     IOCHEK$   REL             104       4611    03/08/73
         UNIT$
 45     KODER$    REL            1524       2565    03/08/73
         KODER.
 46     KRAKER$   REL            1646       7032    03/08/73
         KRAKER.
         ERRSET$
 47     LENGTH$   REL              33       2034    03/08/73
         LENGTH$
 48     MASK$     REL              31       4244    03/08/73
         MASK$
 49     OUTPTB$   REL             351       7210    03/08/73
         OPUTBI.
         OUTPTB.
 50     OUTPTC$   REL             132       6520    03/08/73
         OPUTCI.
         OUTPTC.
 51     OUTPTN$   REL             567       2516    03/08/73
         OUTPTN.
 52     OUTPTS$   REL             127       1302    03/08/73
         ENCODI.
         ENCODE.
 53     READEC$   REL             121       5267    03/08/73
         READEC$
 54     READMS$   REL             175       7535    03/08/73
         READMS$
 55     REWINM$   REL             112       2070    03/08/73
         REWINM.
 56     TRACEX$   REL              42       2376    03/08/73
         TRACEX$
         TRCBKX$
 57     WRITEC$   REL              60       0616    03/08/73
         WRITEC$
 58     WRITMS$   REL             135       4176    03/08/73
         WRITMS$
 59     ABS$      REL              12       2560    03/08/73
         ABS$
 60     IABS$     REL              12       4412    03/08/73
         IABS$
 61     DABS$     REL              45       3344    03/08/73
         DABS$
 62     AINT$     REL              12       1106    03/08/73
         AINT$
 63     INT$      REL              12       7402    03/08/73
         INT$
 64     IDINT$    REL              44       7340    03/08/73
         IDINT$
 65     AMOD$     REL              14       4726    03/08/73
         AMOD$
 66     MOD$      REL              14       5011    03/08/73
         MOD$
 67     AMAX0$    REL              16       2370    03/08/73
         AMAX0$
 68     AMAX1$    REL              15       5047    03/08/73
         AMAX1$
        CATALOG OF SYSMISC        FILE     1
REC     NAME      TYPE          LENGTH    CKSUM      DATE        COMMENTS

 69     MAX0$     REL               15     0756    03/08/73
         MAX0$
 70     MAX1$     REL               16     0456    03/08/73
         MAX1$
 71     DMAX1$    REL               21     4744    03/08/73
         DMAX1$
 72     AMIN0$    REL               16     0174    03/08/73
         AMIN0$
 73     AMIN1$    REL               15     7177    03/08/73
         AMIN1$
 74     MIN0$     REL               15     6723    03/08/73
         MIN0$
 75     MIN1$     REL               16     6560    03/08/73
         MIN1$
 76     DMIN1$    REL               20     7176    03/08/73
         DMIN1$
 77     FLOAT$    REL               12     6316    03/08/73
         FLOAT$
 78     IFIX$     REL               12     2654    03/08/73
         IFIX$
 79     ISIGN$    REL               13     1366    03/08/73
         ISIGN$
 80     SIGN$     REL               13     3553    03/08/73
         SIGN$
 81     DSIGN$    REL               50     5024    03/08/73
         DSIGN$
 82     DIM$      REL               14     4401    03/08/73
         DIM$
 83     IDIM$     REL               13     1021    03/08/73
         IDIM$
 84     SNGL$     REL               11     5155    03/08/73
         SNGL$
 85     REAL$     REL               11     0700    03/08/73
         REAL$
 86     AIMAG$    REL               12     7716    03/08/73
         AIMAG$
 87     DBLE$     REL               12     1765    03/08/73
         DBLE$
 88     CMPLX$    REL               12     4620    03/08/73
         CMPLX$
 89     CONJG$    REL               12     7452    03/08/73
         CONJG$
 90     SHIFT$    REL               12     0177    03/08/73
         SHIFT$
 91     ACOS$     REL               42     6722    03/08/73
         ACOS$
 92     ALNLOGE   REL               54     5611    03/08/73
         ALOG.
         ALOG10.
 93     ALOG$     REL               50     0473    03/08/73
         ALOG$
 94     ALOG10$   REL               50     2727    03/08/73
         ALOG10$
 95     AND$      REL               13     1474    03/08/73
         AND$
```

```
          CATALOG OF SYSMISC          FILE      1
 REC      NAME      TYPE          LENGTH      CKSUM      DATE        COMMENTS

  96      ASIN$     REL               42       2440    03/08/73
            ASIN$
  97      ASNCOSE   REL              137       5337    03/08/73
            ASIN.
            ACOS.
  98      ATAN$     REL               36       2152    03/08/73
            ATAN$
  99      ATANE     REL               76       0536    03/08/73
            ATAN.
 100      ATAN2$    REL               42       3631    03/08/73
            ATAN2$
 101      ATAN2E    REL              120       6562    03/08/73
            ATAN2.
 102      CABS$     REL               45       5532    03/08/73
            CABS$
 103      CABSE     REL               31       6426    03/08/73
            CABS.
 104      CCOS$     REL               65       5310    03/08/73
            CCOS$
 105      CCOSE     REL               27       5457    03/08/73
            CCOS.
 106      CEXP$     REL               63       2306    03/08/73
            CEXP$
 107      CEXPE     REL               25       6004    03/08/73
            CEXP.
 108      CLOG$     REL               43       4757    03/08/73
            CLOG$
 109      CLOGE     REL               32       2764    03/08/73
            CLOG.
 110      COMPL$    REL               11       3426    03/08/73
            COMPL$
 111      COS$      REL               50       0404    03/08/73
            COS$
 112      COSESIN   REL               65       5217    03/08/73
            COS.SIN
 113      CSIN$     REL               65       3251    03/08/73
            CSIN$
 114      CSINE     REL               26       2475    03/08/73
            CSIN.
 115      CSQRT$    REL               36       2156    03/08/73
            CSQRT$
 116      CSQRTE    REL               33       0212    03/08/73
            CSQRT.
 117      DATAN$    REL               40       4061    03/08/73
            DATAN$
 118      DATAN2$   REL               44       3727    03/08/73
            DATAN2$
 119      DATANE    REL              210       5235    03/08/73
            DATAN.
            DATAN2.
 120      DCOS$     REL               45       0150    03/08/73
            DCOS$
 121      DEXP$     REL               52       4402    03/08/73
            DEXP$
 122      DEXPE     REL              130       2525    03/08/73
            DEXP.
```

```
      CATALOG OF SYSMISC         FILE      1
REC   NAME      TYPE          LENGTH     CKSUM      DATE        COMMENTS

123   DLNLOGE   REL              175      1241    03/08/73
       DLOG.
       DLOG10.
124   DLOG$     REL               52      0121    03/08/73
       DLOG$
125   DLOG10$   REL               52      4422    03/08/73
       DLOG10$
126   DMOD$     REL               50      5645    03/08/73
       DMOD$
127   DMODE     REL               35      3565    03/08/73
       DMOD.
128   DSIN$     REL               45      5211    03/08/73
       DSIN$
129   DSNCOSE   REL              175      2767    03/08/73
       DSIN.
       DCOS.
130   DSQRT$    REL               43      6377    03/08/73
       DSQRT$
131   DSQRTE    REL               42      6245    03/08/73
       DSQRT.
132   DTOD$     REL              103      6635    03/08/73
       DTOD$
133   DTODE     REL               31      4632    03/08/73
       DTOD.
       DTOD$.
134   DTOI$     REL              107      7407    03/08/73
       DTOI$
135   DTOIE     REL               34      1640    03/08/73
       DTOI.
       DTOI$.
136   DTOX$     REL              102      4405    03/08/73
       DTOX$
137   DTOXE     REL               26      0147    03/08/73
       DTOX.
       DTOX$.
138   DTOZ$     REL              117      2164    03/08/73
       DTOZ$
139   DTOZE     REL               35      2612    03/08/73
       DTOZ.
       DTOZ$.
140   DISPLA$   REL              265      2140    03/08/73
       DISPLA$
141   EXP$      REL               62      4003    03/08/73
       EXP$
142   EXPE      REL               57      7351    03/08/73
       EXP.
143   HYPERBE   REL               41      7555    03/08/73
       HYPERB.
144   ITOD$     REL              102      3475    03/08/73
       ITOD$
145   ITODE     REL               32      1174    03/08/73
       ITOD.
       ITOD$.
146   ITOJ$     REL               63      1063    03/08/73
       ITOJ$
       CATALOG OF SYSMISC          FILE      1
REC    NAME       TYPE           LENGTH    CKSUM      DATE        COMMENTS

147    ITOJE      REL                27     3061    03/08/73
        ITOJ.
        ITOJ$.
148    ITOX$      REL                75     0767    03/08/73
        ITOX$
149    ITOXE      REL                26     5016    03/08/73
        ITOX.
        ITOX$.
150    ITOZ$      REL               116     2322    03/08/73
        ITOZ$
151    ITOZE      REL                17     1446    03/08/73
        ITOZ.
        ITOZ$.
152    LABEL$     REL                44     1657    03/08/73
        LABEL$
153    LEGVAR$    REL                14     3447    03/08/73
        LEGVAR$
154    LOCF$      REL                11     1353    03/08/73
        LOCF$
155    OR$        REL                13     5546    03/08/73
        OR$
156    OVERLA$    REL               173     3035    03/08/73
        OVERLA4
157    PAUSE$     REL                31     6723    03/08/73
        PAUSE.
158    RANDOME    REL                13     3633    03/08/73
        RANDOM.
        RANMLT.
159    RANF$      REL                17     0271    03/08/73
        RANF$
160    RANGET$    REL                15     1250    03/08/73
        RANGET$
161    RANSET$    REL                23     1463    03/08/73
        RANSET$
162    REMARK$    REL                37     7252    03/08/73
        REMARK$
163    SEGMEN$    REL               163     1525    03/08/73
        SEGMENR
164    SIN$       REL                50     4457    03/08/73
        SIN$
165    SINCOSE    REL                72     4663    03/08/73
        SIN.
        COS.
166    SLITE$     REL                35     1711    03/08/73
        SLITE$
167    SLITET$    REL                36     0310    03/08/73
        SLITET$
168    SQRT$      REL                42     3703    03/08/73
        SQRT$
169    SQRTE      REL                33     4661    03/08/73
        SQRT.
170    SSWTCH$    REL                36     7007    03/08/73
        SSWTCH$
171    TAN$       REL                44     0341    03/08/73
        TAN$
```

```
      CATALOG OF SYSMISC          FILE      1
REC   NAME      TYPE            LENGTH    CKSUM      DATE       COMMENTS

172   TANE      REL                111     2350    03/08/73
       TAN.
173   TANH$     REL                 36     0200    03/08/73
       TANH$
174   TANHE     REL                 45     2043    03/08/73
       TANH.
175   TIME$     REL                 47     6344    03/08/73
       TIME$
       DATE$
       SECOND$
176   XTOD$     REL                102     3275    03/08/73
       XTOD$
177   XTODE     REL                 31     1420    03/08/73
       XTOD.
       XTOD$.
178   XTOI$     REL                 67     3174    03/08/73
       XTOI$
179   XTOIE     REL                 26     5301    03/08/73
       XTOI.
       XTOI$.
180   XTOY$     REL                 75     4045    03/08/73
       XTOY$
181   XTOYE     REL                 25     1100    03/08/73
       XTOY.
       XTOY$.
182   XTOZ$     REL                116     2750    03/08/73
       XTOZ$
183   XTOZE     REL                 35     1406    03/08/73
       XTOZ.
       XTOZ$.
184   ZTOI$     REL                110     1570    03/08/73
       ZTOI$
185   ZTOIE     REL                 33     2073    03/08/73
       ZTOI.
       ZTOI$.
186   DDSUBMV   REL                166     5161   03/17/73
       D.SUBMV
187   DDCOBIO   REL               5531     5031   03/17/73
       D.OPRAN
       D.OPIN
       D.OPOT
       D.CLOS
       D.WRITE
       D.READ
       D.MOCKR
       D.MOCKW
       D.WAA
       D.CRELR
       D.WBA
       D.RERUN
       D.SEEK
       D.TBIG
       D.TSMAL
       FRDGJO
       D.NEWP
```

```
        CATALOG OF SYSMISC        FILE     1
REC     NAME       TYPE         LENGTH     CKSUM      DATE       COMMENTS

188     DDEXP      REL              52      7543   03/17/73
          D.EXP
189     DDANCM     REL             205      5674   03/17/73
          D.ALPCM
          D.NUMCM
190     DDSUBSC    REL             210      3473   03/17/73
          D.SBSC1
          D.STOR
          D.STOR1
          D.SBSC2
          D.SBSC6
          D.SBSC7
          D.SBSCK
191     DDSOL      REL             135      5351   03/17/73
          SOL
          SOLP
          SOLE
          OVRLYNM
192     DDSORT     REL             642      4550   03/17/73
          D.SORT
          D.SRTFL
193     FINIS      REL             136      3373   03/17/73
          D.FINIS
          D.FILES
194     DDDSPLY    REL             447      0660   03/17/73
          D.DSPLY
          D.WRDSP
          D.STOP
          D.DATE
          D.PAUSE
195     DDXCEPT    REL             320      0477   03/17/73
          D.XCEPT
          TERMINL
196     DDDADD     REL              36      1735   03/17/73
          D.DADD
197     DDFIVED    REL             141      3447   03/17/73
          D.TEMPI
          D.FIVED
          D.TEMP7
          D.ZERO
          D.SIXES
          D.TEMPN
          D.TEMPC
          D.TEMP
          D.BLANK
          D.TEMPS
          D.SRCH1
          D.MXOFF
198     DDTRUBL    REL             143      4746   03/17/73
          D.UNDFP
          D.TRUBL
199     DDZONE     REL             163      5250   03/17/73
          D.ZONE3
          D.ZONE5
          D.ZONE6
          D.ZONE7
        CATALOG OF SYSMISC          FILE     1
REC     NAME      TYPE           LENGTH     CKSUM     DATE        COMMENTS

          D.ZONE0
          D.ZONE1
          D.ZONE2
          D.ZONE4
200     DDSTRP    REL               173      1421   03/17/73
          D.STRP1
          D.STRP2
          D.STRP3
          D.STRP4
201     DDDATCN   REL               102      5612   03/17/73
          D.DATCN
202     DDTENDP   REL               124      6273   03/17/73
          D.TENDP
          D.SLASH
          D.SLSH1

          D.QUOT2
          D.DIVS1
          D.DIVS2
          D.DIVD2
          D.DIVD1
203     DDTENS    REL                36      0705   03/17/73
          D.TENS
204     DDTNTHS   REL                63      7564   03/17/73
          D.TNTHS
          D.FIVES
205     DDCVBD    REL               165      1057   03/17/73
          D.CVBD3
          D.CVBD5
          D.CVBD6
          D.CVBD7
206     DDBN      REL               237      3264   03/17/73
          D.BN1SS
          D.BN6SS
          D.BN1DS
          D.BN6DS
          D.BN1DD
          D.BN6DD
          D.BN1SB
207     DDED      REL               470      3325   03/17/73
          D.ED3
          D.ED5
          D.ED6
          D.ED7
          D.EDI
          D.A0
          D.A06
          D.A07
208     DDEDAL    REL               163      1610   03/17/73
          D.EDAL
209     DDEDIT    REL               534      6467   03/17/73
          D.EDIT3
          D.EDIT5
          D.EDIT6
          D.EDIT7
          D.EDITI
```

```
       CATALOG OF SYSMISC          FILE      1
REC    NAME      TYPE            LENGTH    CKSUM       DATE       COMMENTS

         D.AOPT
         D.AOPT6
         D.AOPT7
210    DDEDALP   REL                167     4043    03/17/73
         D.EDALP
211    DDEOP     REL                232     0076    03/17/73
         D.EOP1
         D.EOP2
         D.EOP3
         D.EOP4
         D.EOP5
         D.SAVA
212    DDEXAMO   REL                401     3747    03/17/73
         D.EXAMO
213    DDMOVIO   REL                200     6154    03/17/73
         D.MOVIO
214    DDBCDCM   REL                326     4255    03/17/73
         D.BCDCM
         D.BCDCP
215    CONV1     REL                117     1614    03/17/73
         CONV1
216    DDPAGE    REL                140     6226    03/17/73
         D.PAGE
217    DDLNCT    REL                105     5040    03/17/73
         D.LNCT
218    DDSPPR    REL                137     1153    03/17/73
         D.SPPR
219    DDSPACE   REL                207     1412    03/17/73
         D.SPACE
220    DDINITL   REL                130     5466    03/17/73
         D.CFRST
         D.NIT
         D.INITL
221    DDPRINT   REL                230     3132    03/17/73
         D.PRINT
         D.REPC
         D.REPQ
         D.REPA
         D.RHPH
         NOSPACF
222    DDRGEN    REL                362     0650    03/17/73
         D.RSET
         D.RGEN
         D.CHG00
         D.CHG01
         D.CHG02
         D.TABL2
         D.CHG03
         D.CHG04
         D.CHG05
         D.CHG06
         D.CHG07
         D.CHG08
         D.CHG09
         D.RPGRP
         D.RPTAB
```

```
        CATALOG OF SYSMISC        FILE      1
REC     NAME      TYPE          LENGTH     CKSUM      DATE        COMMENTS

          D.ADDFL
          D.ADDSM
          D.LEVL
          D.TERM
          D.LEVEL
          D.ERROR
          D.FETR
          D.JSTIN
          D.RESC
          D.RJ
223     DDBEGRP   REL              23       2202   03/17/73
          D.BEGRP
224     DDENDRP   REL              23       3141   03/17/73
          D.ENDRP
225     DECK10    REL            5604       4312   03/17/73
          PARS
          FEIL
          CEMPTY
          CIOERR
          T.RROPN
          OCTBCD
          RFNAME
          RWRITE
          FBLA
          T.RANFD
          T.GTRCD
          T.XTRCT
          ERFLAG
          RWOPEN
          CRECALL
          CHKPT
          CSEQERR
          MDSKFUL
          VBKA
          VDKA
          T.MPTY
          VBRA
          VBTA
          VBSA
          RANREAD
          VDRA
          VDTA
          VDSA
          FDBA
          VDRU
          VBRU
          FILNAM
          CTOOBIG
          ERFILE
          S.CIOUP
          CTOOFUL
226     DECK12    REL           10652       6250   03/17/73
          SMCON
227     SYSMISC   OPLD            705       5545    73/06/12.

228     * EOF *        SUM =   137031
```

3.4.10 TRANLIB

```
         CATALOG OF TRANLIB         FILE     1                                  73/06/12. 02.39.20.      PAGE       1
REC      NAME       TYPE         LENGTH    CKSUM      DATE      COMMENTS

  1      TRANLIB    ULIB             63     5030    73/06/12.
  2      CALLTSK    REL              31     6055   73/05/20. 72/07/09.            REQUEST SCHEDULING OF TASK(S).
           CALLTSK
  3      CEASE      REL              20     3267   73/05/20. 72/07/09.            TERMINATE TASK EXECUTION.
           CEASE
  4      CMDUMP     REL              57     1524   73/05/20. 72/07/24. 72/10/05. DUMP CENTRAL MEMORY.
           CMDUMP
  5      DMGR       REL             267     6377   73/05/20. 72/07/09.            DATA MANAGER REQUESTS.
           ADDR
           BLKGET
           BLKPUT
           GETT
           GETL
           GETN
           GETNL
           GETB
           GETBL
           GETR
           GETRB
           GETRBL
           GETRL
           GETNR
           GETNRL
           LOCKF
           PURGER
           PUT
           PUTF
           PUTI
           PUTIF
           PUTR
           PUTRF
           PUTRI
           PUTRIF
           RECALAL
           RELES
           RELESAL
           RECHAIN
           REPOS
           UNLOK
           UNLOKAL
           UNLOKF
  6      DSDUMP     REL              40     2017   73/05/20. 72/07/24. 72/10/05. DIRECT SUBSEQUENT DUMPS.
           DSDUMP
  7      JOURNL     REL              31     7047   73/05/20. 72/07/09.            ISSUE JOURNAL MESSAGE.
           JOURNL
  8      SEND       REL              70     0467   73/05/20. 72/07/09.            SEND MESSAGE TO TERMINAL.
           SEND
  9      TARO       REL              36     5667   73/05/20. 72/08/31.            TERMINAL ARGUMENT OPERATIONS.
           TARO
 10      TSIM       REL              40     2541   73/05/20. 72/08/31.            TERMINAL STATUS AND INFORMATION.
           TSIM
 11      TRANLIB    OPLD             25     5260    73/06/12.

 12      * EOF *       SUM =       1006
```

ADDITIONAL MANUALS 4

The following manuals contain additional information on the KRONOS 2.1 supported software.

4.1 MANUALS FOR PRODUCTS IN KRONOS 2.1 PACKAGE

Publication	Publication Number
KRONOS 2 1 General Information Manual	60407100
KRONOS 2.1 Reference Manual	60407000
KRONOS 2.1 Instant Manual	60407200
KRONOS 2.1 Operator's Guide	60407700
COMPASS Version 3 Reference Manual	60360900
COMPASS Version 3 General Information Manual	60343400
COMPASS Version 3 Instant	60361000
COMPASS Version 3 Instruction Card	60361700
Modify Reference Manual	60281700
Modify Instant	60283000
KRONOS 2.1 Export/Import Reference Manual	59150500
Record Manager Version 1 Reference Manual	60307300
Record Manager File Organization User's Guide	60359600
KRONOS 2.1 Text Editor Reference Manual	60408200
KRONOS 2.1 Time-Sharing User's Reference Manual	60407600
KRONOS 2.1 Terminal User's Instant	60407800
Update Reference Manual	60342500
Update Instant	60360200

4.2 MANUALS FOR OPTIONAL PRODUCTS

Publication	Publication Number
ALGOL Version 3 Reference Manual	60329000
APL*CYBER Reference Manual	19980400
KRONOS 2.1 BASIC Reference Manual	19980300
COBOL Version 4 Reference Manual	60384100
COBOL Version 4 General Information Manual	60327800
COBOL Version 4 Instant	60328400

Publication	Publication Number
Cyberlink Interchange, Version 1 Reference Manual	60373300
Cyberlink Interchange, Version 1 Operator's Guide	60408400
FORTRAN Version 2.3 Reference Manual	60174900
FORTRAN Version 2.3 Instant	60189500
FORTRAN Extended Version 4 Reference Manual	60305600
FORTRAN Extended Version 4 General Information Manual	60327900
FORTRAN Extended Version 4 Instant	60357900
FORTRAN Extended Debug User's Guide	60329400
PERT/Time Version 1.2 Reference Manual	60133600
PERT/Time Version 1.2 General Information Manual	60133300
SIMSCRIPT Version 3 Reference Manual	60178300
SIMSCRIPT Version 3 General Information Manual	60173500
SIMULA Version 1 Reference Manual	60234800
SIMULA Version 1 General Information Manual	60251900
SIMULA Version 1 Instant	60235100
Sort/Merge Version 4 Reference Manual	60343900
Sort/Merge Version 4 General Information Manual	60342400
Sort/Merge Version 4 Instant	60344000
KRONOS 2.1 Time-Sharing FORTRAN Reference Manual	60408600
KRONOS 2.1 Transaction Subsystem Reference Manual	60407900
KRONOS 2.1 Transaction Subsystem Operator's Guide Addendum	60408000

SUMMARY OF DEADSTART 1

Deadstart is the process of loading the current version of the KRONOS 2. 1 deadstart tape into core memory and mass storage. The KRONOS 2. 1 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 1. 1. 9, for the system catalog.

1.1 PROCEDURE

The deadstart process includes the following actions.

- 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 are not to be modified
- Activating the DEADSTART switch
- Initializing the system
- Initiating job processing

The following outline lists the procedure necessary to deadstart a KRONOS 2. 1 system. It references sections containing detailed instructions.

Mount Deadstart Tape

1. Ensure that all mass storage devices are mounted and/or available.

2. Mount current deadstart tape on tape unit to be specified on deadstart panel (word 2 specifies deadstart tape unit and controller); ready the unit.

   Refer to section 2 of the operator's guide for instructions for mounting the deadstart tape.

Set Deadstart Panel

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

   Part II, section 2, describes deadstart panel selections.

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

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 3.5 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 bits 0 through 5 of word 14 are currently set, the system library is loaded onto the system devices. Continue with step 9; there are no further options before initializing the system.

Refer to part II, section 3.2.1.

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.

   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.)

   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. 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.
    ```

11. Type yy/mm/dd.
    Press CR

yy	Year; 0 through 99
mm	Month; 1 through 12
dd	Day; 1 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.

12. When DSD accepts the date entry, it requests the time.

    ```
    ENTER TIME HH.MM.SS.
    ```

13. Type hh.mm.ss.
    Press CR

hh	Hour; 0 through 23
mm	Minute; 0 through 59
ss	Second; 0 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

14. 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 and is flashed after date and time are entered. (Each DSD command in the IPRDECK flashes while it is processed.) 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 17.

    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 VALIDUX (account validation), PROFILO (project validation, if available), RESEXDF and RESEXVF (resource management control files):

    Type X.ISF.
    Press CR

15. 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.

16. 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.

17. 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.

1.2 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.
Tape does not move.	● 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.

DEADSTART PANEL SETTINGS 2

2.1 GENERAL DESCRIPTION

The deadstart program is a 12-instruction program that is transferred from the deadstart panel to locations 1 through 14_8 in PPU 0 memory to execute when the DEADSTART switch is activated.

The deadstart program:

Specifies and activates the equipment to read the deadstart tape (words 1 through 12)

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 13, rrr)

Figure II-2-1 illustrates the deadstart panel switch positions for the KRONOS 2.1 deadstart program.

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

Figure II-2-1. KRONOS 2.1 Deadstart Program

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.

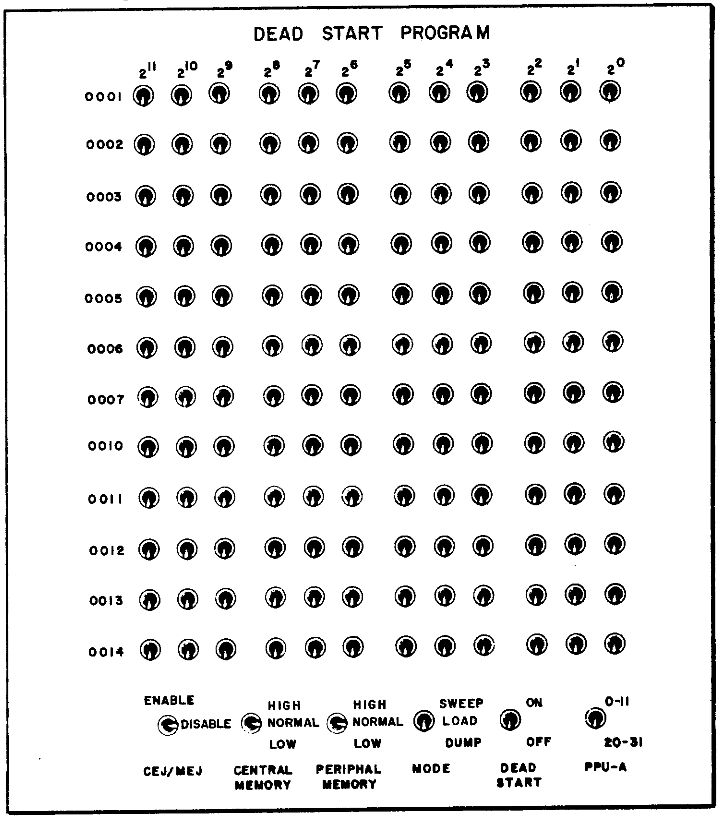


Figure II-2-2. Deadstart Panel

In reference to Figure II-2-2, move switch up to set in 1 or ON position; move switch down to set in 0 or OFF position. Activate the DEADSTART switch by briefly holding in ON position.

2.2 WORDS 0000 THROUGH 0012

Set the switches for words 0000 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.

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

ccc ccc Channel used to access the magnetic tape equipment

eee Controller associated with magnetic tape unit uuu

uuu Magnetic tape unit on which the deadstart tape is mounted

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 causes the deadstart program (PRL) that was read from the deadstart panel to execute. PRL has two functions: to display and process any maintenance options that are selected and to load the proper records from tape that are selected. (For example, if the PP0 or EJT options are selected, PRL loads these records.)

2.3 WORD 0013

Word 13 specifies the deadstart function and the CMRDECK to be used at deadstart.

0013	000 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 KRONOS 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, bypasses maintenance tests PP0, DDD, PMM, MM2, and EJT, and activates SET (processes CMRDECK entries).

The first display to appear at the system display after the deadstart switch is activated is determined by the setting of word 14, bits 0 through 5.

yyy — Deadstart Function

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. 1.)

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 3.8).

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 maintenance deadstarts may be selected during a system deadstart with options displayed (yyy=001). These are EJT, EDD, MM2, and PMM.

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.

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 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.

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.

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 KRONOS 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 switch labeled CEJ/MEJ at the bottom of the deadstart panel. Refer to Figure II-2-2 in this section.

rrr These switches designate the level of system deadstart. For additional information concerning levels of deadstart, refer to section 3 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 KRONOS 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 and active files are lost and the system is reloaded from the deadstart tape.

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 are also recovered. At level 1, recovery deadstart is normally performed following a system malfunction that destroys the contents of central memory. Level 1 recovery is effective only if the DSD command CHECK POINT SYSTEM (refer to operator's guide) has been issued prior to the system malfunction. 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. It may be desirable to perform a level 1 recovery deadstart temporarily (where possible) in order to dump accounting information (refer to DSD command ACCOUNT in operator's guide).

rrr Level of System Deadstart

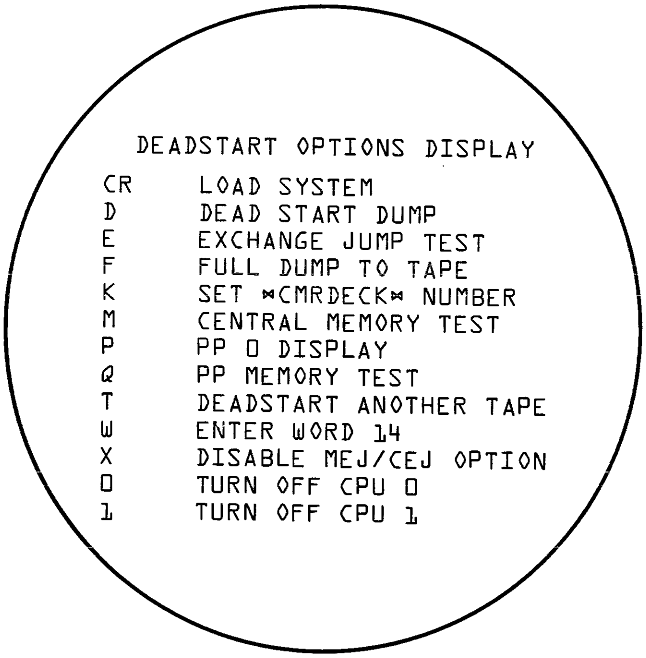
Unless the operator is certain that the content of central memory was destroyed by the system malfunction, he should always attempt to perform a level 3 recovery deadstart (rrr=011) before attempting level 1 recovery.

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.

011 Indicates a level 3 recovery deadstart whereby the system, all jobs, and all active files are recovered from central memory tables. Permanent files are also recovered. 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 level 1 or level 0 deadstart.

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.



```
   DEADSTART OPTIONS DISPLAY
CR      LOAD SYSTEM
D       DEAD START DUMP
E       EXCHANGE JUMP TEST
F       FULL DUMP TO TAPE
K       SET *CMRDECK* NUMBER
M       CENTRAL MEMORY TEST
P       PP 0 DISPLAY
Q       PP MEMORY TEST
T       DEADSTART ANOTHER TAPE
W       ENTER WORD 14
X       DISABLE MEJ/CEJ OPTION
0       TURN OFF CPU 0
1       TURN OFF CPU 1
```

Description

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

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.

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. 5 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. 10 to specify the system devices.

3. At the system console, press CR.

   The system library is loaded onto the system devices. The next display requests the date. For subsequent procedure, refer to section 1, step 10.

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. 5 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. 10 to do so.

3. At the system console, press CR.

   The next display 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. 2. 1.
Dump PPU 0 saving all memory destroyed by DDD	Refer to section 3. 2. 2.
Dump central memory	Refer to section 3. 2. 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 O MEMORY DURING DUMP

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

Locations 0 through 15_8 of PPU 0	Deadstart program
Locations 6600 through 7777 of PPU 0	Deadstart dump program
Locations 0 through 3 of all other PPUs	PPU idler routine

Location 6600 through 7777 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.

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 Normal PPU Memory Dump procedure. If no further dumps are desired, activate the DEADSTART switch.

3.4 EJT: EXCHANGE JUMP TEST OPTION

This option displays EJT, the exchange jump deadstart maintenance test which is used to determine the reliability of the hardware.

CAUTION

Entering E destroys central memory.

1. Type E
2. The EJT display appear on the system console listing possible options; perform any necessary tests.
3. Activate the DEADSTART switch. Do not use a level 3 recovery deadstart (word 14, rrr) because central memory is destroyed.

3.5 DUMP TO TAPE OPTION

EDD (express deadstart dump) is a maintenance program that can be used to dump all PPUs' memories as well as central memory 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 15_8 of PPU 0	Deadstart program
Locations 6000 through 7777 of PPU 0	EDD program
Locations 0 through 3 of all other PPUs	PPU idler routines

Locations 6000 through 7777 of 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 to standard setting (part II, section 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
uu	Two-digit number of tape unit

   Enter e0uu from right to left.

7. EDD requests the dump identifier; this identifier will be placed in the tape label 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 xx
    Press CR

    xx Type a two-digit nonzero entry to prevent tape rewind before and after dump; enter 00 (or press CR) to cause tape rewind before and after dump.

    Dump of central and PPU memories occurs.

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

    If ECS is dumped, an area of central memory that is approximately 1040_8 words in length is destroyed after central memory is dumped. The first word address of this area of central memory is defined by CMEC (an EDD assembly symbol). The released value of CMEC is $40,000_8$.

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 CSaa DSbbbb 00

err Indicates error type

CON	Connect reject error
FCN	Function reject error
WRT	Write error

aa	Channel converter status
bbbb	Controller status

Pressing CR causes EDD to retry the dump operation.

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 is the 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 MM2: CENTRAL MEMORY TEST OPTION

The central memory test option is a deadstart maintenance test.

CAUTION

Entering M destroys central memory.

1. Type M

2. The MM2 display appears at the system console listing possible options; perform necessary tests.

3. Activate the DEADSTART switch. Do not use a level 3 recovery deadstart (word 14, rrr) because central memory is destroyed.

3.8 PPO: DISPLAY PPU O 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.

3.9 PMM: PPU MEMORY TEST OPTION

The PPU 0 memory test (PMM) is a deadstart maintenance test that tests memory reliability for each PPU in the system.

1. Type Q
2. The PMM display appears listing possible options; perform necessary operations.
3. Activate the DEADSTART switch. (Central memory is left intact.)

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; ready the unit.
4. Type e0uu
   Press CR

   e One-digit controller number of tape unit containing newly selected deadstart tape.

   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.

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 KRONOS 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 OPTION

Use this option to turn off CPU 0 (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.

To specify CPU 0 as available, redeadstart.

3.14 TURN OFF CPU 1 OPTION

Use this option to turn off CPU 1 (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.

To specify CPU 1 as available, redeadstart.

CMRDECK 4

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.

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:

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. Several of the entries listed are assigned system default values. These values are assumed if the entries do not appear in the CMRDECK being used.

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 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 job KR1. — Part III, section 1. 1. 3

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

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. Specify all parameters unless they are noted as optional.

3. Separate each entry with a comma when indicated in the entry format.

4. Terminate each entry 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 KRONOS.

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.

DFB=400.  SET DAYFILE BUFFER LENGTH = 400.
ACF=400.  SET ACCOUNT FILE BUFFER LENGTH = 400.
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.
IPD=0.    ASSEMBLE INSTALLATION PARAMETER DECK 0.
STL=6.    SET SYSTEM EVENT TABLE LENGTH = 6.
ERL=100.  SET ERROR LOG BUFFER LENGTH = 100.
CM=XXXX.  SET CENTRAL MEMORY SIZE TO XXXX HUNDRED WORDS.

EQXX=TY,ST,EN,UN,A,B,C,D. DEFINE EQUIPMENT XX AS FOLOWS
          TY = TYPE (2 LETTERS)
          N = NUMBER OF UNITS CONSECUTIVE STARTING AT *UN*.
          ST = STATUS (ON, OFF)
          EN = EQUIPMENT NUMBER
          UN = UNIT NUMBER
          A - D = CHANNELS
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-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.  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.
TK=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=07,C2000.         DH 07, 3553/821.
     RTK=10,C623,T22,S27.  DI 10, 7054/844-N.
     RTK=11,C2000,S2411.   MD 11, 3553/841-N.
```

```
      THE FOLOWING ENTRIES ARE CLEARED IF EQ IS REDEFINED.
ASR=X,Y,...,Z. SET ALTERNATE SYSTEM DEVICES.
FAMILY=NN. SET EQUIPMENT NUMBER OF DEFAULT FAMILY.
PF=XX,TY,DM,FN,DN.
          DEFINE DEVICE XX FOR PERMANENT FILES.
          REQUIRED ONLY WHEN INITIALIZING.  PARAMETERS ARE SET
          FROM LABEL WHEN NOT INITIALIZING.
          TY = TYPE OF PERM. FILE RESIDENCE.
                *D* = DIRECT ACCESS ALLOWED
                *I* = ONLY INDIRECT ACCESS FILES
                *X* = AUXILIARY (DIRECT AND INDIRECT)
          DM = DEVICE MASK.
          FN = FAMILY NAME (1-7 CHARACTERS)
          DN = DEVICE NUMBER.

INITIALIZE,XX,NC.
          BLANK LABEL MASS STORAGE DEVICE XX.
          ENTRY SHOULD BE USED FOR ALL DEVICES ON WHICH P.F.
          ARE NOT DESIRED AND FOR ALL DEVICES WHICH ARE BEING
          DEFINED AS P.F. DEVICES BY *PF* ENTRY.
                   NC = CATALOG TRACKS (POWER OF 2 .LE. 200)

REMOVE=X,Y,...Z. SET X,Y,...Z AS REMOVABLE DEVICES.
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.
```

<u>Format of the CMRDECK on the Released Deadstart Tape :</u>

```
CMRDECK
NAME= 6400 S/N 127. CDC TIME SHARING SYSTEM.
NCP=27.
EQ0=DB,ON,1,0,0.
EQ1=DB,ON,1,0,3.
EQ2=MD-1,ON,6,0,4.
EQ3=MD-1,ON,6,1,4.
EQ10=DS,ON,7,,10.
EQ11=CR,ON,4,,13.
EQ12=CP,ON,5,,13.
EQ20=LP,ON,6,,13.
EQ21=LP,ON,3,,13.
EQ22=LP,OFF,7,,13.
EQ30=ST,ON,7,0,2.
EQ37=TT,OFF,7,1,5,0,0,1.
EQ40=TT,ON,7,,1,,40.
EQ50=MT-4,ON,5,0,12.
TEMP=0,1.
```

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
ACF=length.	400_8	This entry sets the 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. length — 3 or 4 octal digit length of account file buffer; must be a multiple of 100_8
CM=size.	Actual core size	The actual size of core memory is automatically determined by the SET program during deadstart. This value is a minimum of 32K and a maximum of 131K. 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.

Entry format	Released Default Value	Significance
DFB=length.	400_8	This entry sets the dayfile buffer 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. length — 3 or 4 octal digit length of dayfile buffer in CMR; must be a multiple of 100_8
ERL=length.	100_8	This entry sets the 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. 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=entry.	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
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_{10} 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.

Entry Format	Released Default Value	Significance
NAME=date line.	6400 S/N 127. CDC TIME SHARING SYSTEM	This parameter specifies the system date line that is displayed on the system console display. It is stored in words 32 through 37 of CMR. date line — Any length alphanumeric character system date line; only first 60 characters are actually used.
NCP=number.	12_8	This entry sets the number of control points available for job processing to a value other than the default value. Changing the default setting depends upon the type of job mix and environment desired. Each control point uses about 300_8 words of additional CMR space. 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 not available	The released system assumes 10 PPUs. To indicate that 20 PPUs are available, enter this command in the CMRDECK. Reentering the command toggles the setting so that the upper 10 PPUs are unavailable.
PPU=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 PPUs 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 no other entries of this type) deactivates PPU 3 and PPU 4. PPU=3, 4.

4.3 EQUIPMENT ASSIGNMENTS: NONMASS STORAGE

The following EST entries are described in this section.

Clear EST assignment	section 4.3.1
Undefined equipment EST entry	section 4.3.2
Null 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.
```

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-volume serial number association is established with an ASSIGN, LABEL, REQUEST, or VSN control card, 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-volume serial number association established by a VSN control card 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. ASSIGN77. (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

KRONOS 2.1 requires at least one system console to be available for use.

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

ord — 1 or 2 octal digit EST ordinal of display; 1 through 75_8; most sites set to 10

DS — 6612 display console equipment type

status — 2 or 3 digit indicator of whether equipment is available for system use

ON	Available
OFF	Can only use console for the DSD commands

controller — 1 or 2 digit number of system console controller; 0 through 7

unit — Not applicable to the system console EQ entry; enter 0 or leave a blank space

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,,10.

4.3.5 UNIT RECORD EQUIPMENT EST ENTRY

The recommended unit record equipment requirement is one line printer and one card reader.

```
EQord=type,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 KRONOS 2.1.

Equipment	Type
Card reader	
405-3447/3649	CR
Card punch	
415-3446/3644	CP
Line printer	
501-3256/3659	LP
505-3256/3659	LP
512-1—3555-1	LP or LQ

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

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 KRONOS 2.1 requires is two 607's, 657's, or 659's.

EQord=MT-n, status, controller, 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 parameter.

MT-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 KRONOS 2.1.

Controller	Units	n
3x2x	604/607	1-16
3518-1/2/3	657/659	1-8
3528-1/2/3	657/659	1-8

To clear an MT-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

controller — 1-digit controller number of magnetic tape device; 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 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

A controller can be connected to from one to four channels, depending upon the controller model.

When more than one channel access is available to some tapes, the configuration for all units that may be accessed by multiple channels must be the same. Also, all other tapes to be supported must be on one of the multiple channels. For example, an installation site has a 3528-3 and a 3228. If the 3528-3 is accessed through channels 7 and 13, then the 3228 must be on either channel 7 or 13. Also, if another 3528-3 is added, it must be on channels 7 and 13.

option Optional hardware feature parameter.

0 or omitted	No significance
1	Status 2 hardware feature not available
2	Set option to 2 if the conversion memory (code conversion) feature is not available; this would only be used for 3518-1 and 3528-1.

4.3.7 MULTIPLEXER EQUIPMENT EST ENTRY

There are four types of EST entries for communication equipment.

- Time-sharing and Transaction Subsystem entries
- Time-sharing and Transaction Subsystem stimulator entries
- Export/Import 1.0 entry
- Cyberlink Interchange 1.0 entry

TIME-SHARING AND TRANSACTION SUBSYSTEMS 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 and transaction subsystems; 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 subsystem 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. KRONOS 2.1 supports all of these up to 600 baud.

KRONOS 2.1 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 KRONOS 2.1 Transaction Sybsystem Reference Manual.

The stimulator EST entry format is:

```
EQord=TT, status, controller, no., 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

no. 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 transaction subsystem stimulator, the value specified must be the same as the number of lines specified in NETWORK or SIMFILE.

If no. 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 subsystem 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 1.0 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

CYBERLINK INTERCHANGE 1.0 EST ENTRY

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

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

SA — 6673/6674 data set controller equipment type

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 — Parameter not used by Cyberlink 1.0

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

Examples:

EQ15=SA,ON,1,0,4. — Specifies that a 6674 to be used for Cyberlink is available

EQ21=SA,OFF,0,0,3. — Specifies that a 6674 to be used for Cyberlink is not available

4.4 MASS STORAGE EQUIPMENT ASSIGNMENTS

This section describes the following mass storage assignments.

Mass storage equipment EST entry	EQ	section 4.4.2
ECS equipment EST entry	EQ	section 4.4.3
Temporary files device assignment	TEMP	section 4.4.4
Permanent files device assignment	PF	section 4.4.5, 4.4.6
System library device assignment	SYSTEM	section 4.4.7
Alternate system library device assignment	ASR	section 4.4.8
Default family name assignment	FAMILY	section 4.4.9
Removable device assignment	REMOVE	section 4.4.10
Initialization command	INITIALIZE	section 4.4.11

4.4.1 KRONOS 2.1 MASS STORAGE CONCEPTS

The following descriptions and chart summarize the KRONOS 2.1 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 SYSTEMDEVICE

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 4.4.8) specifies which mass storage devices are to contain system library routines; the *AD LIBDECK entry (part III, section 1.1.5) 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 a 6638, 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.

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

System	Ord	Device	Access Used	Contents
X	0	6638	A	Direct access files
X	1	841-4	A	Master device, indirect access files
Y	0	6638	B	Direct access files
Y	1	841-4	B	Master device, indirect access files

If system Y became inoperative, the B accesses for the 6638 and the 841-4 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=DB,ON,1,0,0.        (defines access A)
EQ1=MD-4,ON,4,0,2.      (defines access A)
EQ2=DB,OFF,1,0,1.       (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:

```
EQ0=DB,ON,1,0,1.          (defines access B)
EQ1=MD-4,ON,4,0,3.        (defines access B)
EQ2=DB,OFF,1,0,0.         (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. Note 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 the KRONOS 2.1 Reference Manual and the KRONOS 2.1 Operator's Guide for detailed descriptions of PRIVATE and PUBLIC.

Example:

An 854 to be used as an auxiliary device could be defined as follows:

```
EQ3=DD,ON,5,2,7.
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 doesn't specify one, the default FAMILY entry is used.

A family device can contain direct and/or indirect access files. These files are defined in the KRONOS 2.1 Reference Manual. The files that will be allowed are set with the type parameter on the PF entry.

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 so as 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 the KRONOS 2. 1 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.

For example, an 841-4 to be used as one device could be defined in the CMRDECK as:

```
EQ3=MD-4,ON,3,1,2.
```

Refer to section 4. 4. 2 (type and unit parameters) for specific information on assigning multispindle 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 common files) cannot be removed from the system. It can be an auxiliary device or an alternate permanent file family device.

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.

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	Nonremovable device	Auxiliary device	Family device
System device	no	---	yes	yes	yes	yes	yes	no	yes	yes	yes
Alternate system device	---	no	yes	yes	yes	yes	yes	no	yes	yes	yes
Device containing temporary files	yes	yes	---	yes	yes	yes	yes	no	yes	yes	yes
Device containing direct access files	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
Auxiliary device	yes	yes	yes	yes	yes	yes	no	yes	yes	---	no
Default family device	yes	yes	yes	yes	yes	yes	yes	no	yes	no	---

4.4.2 MASS STORAGE EQUIPMENT EST ENTRY

The purpose of the mass storage EQ entries is to describe all mass storage peripheral equipment. KRONOS 2.1 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 1.1.3.) Each 841, 844, 853, or 854 entry, however, may refer to more than one physical unit. For example, the 844-2 may be defined as two units with two EQ entries, or it could be defined as one unit with one EQ entry.

The 6603 can have only one entry. Define a 6638 with one entry if it is a half-size 6638; define it with two entries if it is a full-size 6638. The 821 has two fixed units and must always have two EQ entries; they cannot be accessed as one unit.

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.

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. 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, chan$_1$, chan$_2$, R.

ord

1 or 2 octal digit EST ordinal of mass storage equipment; 0 through 17_8

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. There are two times to specify which mass storage devices are to be system devices.

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 KRONOS 2.1; n is the number of units connected to a controller.

Equipment	Type	n
813/814—3234	DF	
821—3553-1	DH	
841—3553-1	MD-n	1-8
844—7054-1	DI-n	1-8
853/854—3234	DD	
853/854—3234 (group)	DD-n	1-8
863—3647	DC	
6603	DA	
6638	DB	

The 841, 844, 853, and 854 units can be defined in different ways. Each unit can be defined separately with separate EQ entries just as with the other mass storage units.

If more continuous storage is needed than is possible with one unit, then 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 consecutively numbered units defined with the EQ entry, and the unit parameter identifies the lowest numbered unit of n consecutively numbered units.

For example, an 844-2 with 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.
```

An 844-2 with 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 (624_8 words for the 844). 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; 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; 0 through 7

Controller Name	Number
3234/3553-1/3647	4-7
6603/6638	0-7
7054-1	0

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

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 consecutive 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 consecutively numbered 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 consecutive 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.
```

This parameter does not pertain to the 6603 or to the 6638; set to 0 or omit with a comma for this equipment.

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.

Equipment	Number of Channels
841—3553-1	1 or 2
844—7064-1	1 or 2
853/854—3234	1 or 2
813/814—3234	1
821—3553-1	1 or 2
6603	1
6638	1
863—3647	1

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 — Optional parameter for a dual access unit. The two controllers must have the same controller number (0 through 7) when R is part of the EQ entry. A release unit reserve function occurs whenever a channel is dropped. KRONOS 2.1 will not usually release controller reservations when an I/O operation is complete (or a single access device unless R is specified).

If the R parameter is omitted from dual access (two channels specified) EQ entry, the second channel automatically sets the R option.

4.4.3 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. KRONOS 2.1 supports up to three DDPs.

The format of the entry is:

```
EQord=type,status,size,chan1,chan2.
```

ord — 1 or 2 octal digit EST ordinal of ECS; 0 through 17_8

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 — 3 or 4 octal digit number representing size of ECS unit

Size	ECS Available
400	128K (1 bank)
1000	256K (2 banks)
2000	512K (4 banks)
4000	1024K (8 banks)
10000	2058K (16 banks)

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.
```

4.4.4 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

This assignment was indicated with the EQ entry status (ON/OFF) parameter in KRONOS 2.0.

```
TEMP=ord,...,ord.
```

ord 1 or 2 octal digit EST ordinal of mass storage device; 0 through 17_8; one or more ordinals may be specified with one entry.

4.4.5 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,$\substack{\text{I}\\\text{D}}$,mask, name, device.

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

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

ord — 1 or 2 octal digit EST ordinal of mass storage device; 0 through 17_8

I,D,X — This parameter specifies the type of files that the unit is to contain.

I — Unit can only contain indirect access files; master device

D — Unit can contain only direct access files if its mask parameter is set to 0; it could not in this case be a master device

It could contain indirect and direct access files if the mask parameter is set to 1 through 377_8; it is a master device in this case.

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.

mask — 1, 2, or 3 octal digit parameter specifying the unit's 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.

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.

Examples:

```
PF=2,I,125,LINE1,3.
PF=17,X,UFO.
```

TABLE II-4-2. PF ENTRY

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

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

4.4.6 KRONOS 2.0 PERMANENT FILE CONVERSION TO KRONOS 2.1

Permanent files created under KRONOS 2. 0 systems that are updated through PSR level 10 may be recovered by KRONOS 2. 1 systems.

If the permanent file devices are not reconfigured under 2. 0 or 2. 1, the installations can alternate between 2. 0 and 2. 1 systems without reloading permanent files. Auxiliary devices are not accessible under 2. 0 systems.

The only difference in the EQ entries is that under 2. 0, the status parameter referred to whether or not the device was available for temporary files; under 2. 1, it refers to whether or not the device is available for access. The TEMPord. entry (section 4. 4. 4) specifies whether or not the device is to be used for temporary files under 2. 1.

Only the identical EQ entries are necessary in the 2. 1 CMRDECK to recover permanent files created with 2. 0 systems. The PF entry is not required to recover KRONOS 2. 1 permanent files. However, the PF entries are identified in the following example to indicate what the PF entry characteristics are for 2. 1.

To determine the permanent file definitions needed for the KRONOS 2. 1 CMRDECK, use the following information.

The KRONOS 2. 0 PF entry format was:

```
PF=ord,status,file type,mask.
```

The KRONOS 2. 1 PF entry format is:

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

status — Delete the status parameter on the KRONOS 2. 1 PF entry

file type — The 2. 0 variables were D for device dedicated to indirect access permanent files and S for devices with direct and indirect access permanent files.

To use the permanent files created under a 2. 0 system with a 2. 1 system, convert the parameters as follows:

2. 0 parameter	2. 1 parameter
S	D
D	I

mask — Same as for 2. 0 and 2. 1

name — Omit for 2. 1 because there was no family or pack name under 2. 0. No pack or family name can be specified as long as there is to be an alteration between 2. 0 and 2. 1 systems.

device — Catalog ordinal of 2. 0 device biased by 40_8

Example:

```
If 2.0 CMRDECK is:             2.1 CMRDECK is:

EQ0=DB,ON,1,0,0.               EQ0=DB,ON,1,0,0.
EQ1=DB,ON,1,0,1.               EQ1=DB,ON,1,0,1.
EQ2=MD-4,OFF,4,0,2.            EQ2=MD-4,ON,4,0,2.
EQ3=MD-2,ON,4,4,2.             EQ3=MD-2,ON,4,4,2.
                               TEMP=0,1,3.
PF=1,ON,S,.                    PF=1,D,,,40.
PF=2,ON,D,377.                 PF=2,I,377,,41.
PF=3,ON,S,.                    PF=3,D,,,42.
```

4.4.7 SYSTEM: SYSTEM LIBRARY DEVICE ASSIGNMENT

The SYSTEM entry specifies which mass storage devices are to contain copies of the KRONOS 2.1 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 DB or 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; the EQ entry must be set to ON status. One or more ordinals may be specified with one SYSTEM entry.

4.4.8 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, and PP 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 1.1.5.

Note the following restrictions.

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.

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 alternate SYSTEM entry.

4.4.9 FAMILY: FAMILY NAME ASSIGNMENT

If there is only one permanent file family in the active system, specifying a family name is not necessary. However, 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.

4.4.10 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, or FAMILY assignment.
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; one or more ordinals may be specified with one REMOVE entry.

4.4.11 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. It 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 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, all flaw reservations specified for a device are lost and must be reentered. Refer to part II, section 4. 5.

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

The entry format is:

```
INITIALIZE,ord,tracks.
```

ord — 1 or 2 octal digit EST ordinal of mass storage device to be initialized; 0 through 17_8

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.

tracks — Optional 1, 2, or 3 octal digit parameter that specifies the number of catalog tracks allowed for a master device if the default number is not to be used; 1 through 200_8; must be a power of 2; omit if not a master device. Note that the fewer the catalog tracks available, the longer each permanent file request will take since the catalog search will be longer.

The defaults for each mass storage equipment type are:

Device	Equipment Type	Default No. Catalog Tracks
813/814	DF	100
821	DH	100
841	MD	20
844	DI	20
853/854	DD	4
863	DC	2
6603	DA	10
6638	DB	10
ECS	DE	2
ECS with DDP	DP	2
	Private device	1

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.

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; 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

address — Physical address of mass storage area to be cancelled; refer to the formats for each mass storage device as shown by the RTK entry descriptions that follow.

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

track — 4 octal digit logical track to be reserved; 4000 through 7777_8

4.5.4 6603 DISK SYSTEM TRACK RESERVATION ENTRY

Use the following entry to reserve a track on a 6603 disk system.

```
RTK=ord, Ttrack, Ggroup, Ssector.
```

ord — 1 or 2 octal digit EST ordinal of 6603 disk system; 0 through 17_8

track — 1, 2, or 3 octal digit physical track number to be reserved; 0 through 200_8

group — 1 or 2 octal digit head group number; 0 through 10_8

sector — 1, 2, or 3 octal digit sector number; 0 through 144_8

4.5.5 6638 DISK SYSTEM TRACK RESERVATION ENTRY

Use this entry format to reserve tracks on the 6638 disk system.

```
RTK=ord, Ptrack, Hgroup, Ssector.
```

ord	1 or 2 octal digit EST ordinal of 6638; 0 through 17_8
track	1 or 2 octal digit number of track to be reserved; 0 through 40_8
group	1 or 2 octal digit head group number; 0 through 40_8
sector	1 or 2 octal digit sector number; 0 through 44_8

Examples:

```
RTK=0,P11,H17,S34.
RTK=0,P21,H1,S3.
RTK=1,P31,H12,S44.
RTK=1,P34,H26,S45.
```

4.5.6 863 DRUM RESERVATION ENTRY

Use the following entry to reserve areas on the 863 drum.

```
RTK=ord,Uunit,Ggroup.
```

ord	1 or 2 octal digit 863 EST ordinal; 0 through 17_8
unit	1 or 2 octal digit unit number; 0 through 10_8
group	1, 2, or 3 octal digit head group number; 0 through 100_8

4.5.7 853/854 DISK STORAGE DRIVE RESERVATION ENTRY

Use the following entry to reserve areas on the 853 or 854 disk storage drive.

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

ord	1 or 2 octal digit 853 or 854 EST ordinal; 0 through 17_8
cylinder	1, 2, or 3 octal digit cylinder area; 0 through 310_8

Enter flaws for an 853/854 according to how it is to be used (whether as two DD-1's or as one DD-2, for example).

4.5.8 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

address — 1 through 6 octal digit logical address in a track of ECS; track containing absolute address is reserved

Example:

```
RTK=4,A714140.
```

4.5.9 813/814 DISK FILE RESERVATION ENTRY

Use the following entry to reserve areas on a 813 or 814 disk file.

```
RTK=ord,Ccylinder,Ttrack.
```

ord — 1 or 2 octal digit 813 or 814 EST ordinal; 0 through 17_8

cylinder — 1 through 3 octal digit cylinder number; 0 through 200_8

track — 1 through 5 octal digit track number; 0 through 10000_8

Example:

```
RTK=2,C0,T2435.
RTK=2,C100,T30.
RTK=3,C10,T4011.
RTK=3,C21,T1162.
```

4.5.10 821 DATE FILE RESERVATION ENTRY

Use the following entry to reserve areas of an 821 data file.

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

ord — 1 or 2 octal digit 821 EST ordinal; 0 through 17_8

cylinder — 1 through 4 octal digit cylinder number; 0 through 2000_8

4.5.11 841 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

cylinder 1 through 4 octal digit cylinder number; 0 through 2000_8

sector 1 through 4 octal digit sector number; 0 through 2411_8

Enter flaws for an 841 according to how it is to be used (whether as two MD-1's or as one MD-2, for example).

4.5.12 844-2 DISK STORAGE SUBSYSTEM RESERVATION ENTRY

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

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

ord 1 or 2 octal digit 844-2 EST ordinal; 0 through 17_8

cylinder 1 through 3 octal digit cylinder number; 0 through 632_8

track 1 or 2 octal digit track number; 0 through 22_8

sector 1 or 2 octal digit sector number; 0 through 27_8

Enter flaws for an 844-2 according to how it is to be used (whether as two DI-1's or as one DI-2, for example).

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).

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 supercedes the value currently specified in the IPRDECK. However, the modified IPRDECK remains in effect only until the next 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 1.1.2).

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>RELEASED FORMAT OF IPRINST</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.
LOCK.      TOGGLE CONSOLE LOCK STATUS.
PRIORITY.  TOGGLE PRIORITY EVALUATION DISABLE.
VALID.     TOGGLE JOB VALIDATION DISABLE.
BATCHIO.   TOGGLE AUTO STATUS OF *BATCHIO*.
EI200.     TOGGLE AUTO STATUS OF *EI200*.
MAGNET.    TOGGLE AUTO STATUS OF *MAGNET*.
TELEX.     TOGGLE AUTO STATUS OF *TELEX*.
TRANEX.    TOGGLE AUTO STATUS OF *TRANEX*.

DELAY,T1XXX,T2XXX,....,TNXXX.
           SET DELAY TIME *TN* = XXX.
     TN = AR      AUTO RECALL (MILLISECONDS)
          CS      CPU JOB SWITCH (MILLISECONDS)
          CR      CPU PROGRAM RECALL (MILLISECONDS)
PU PROGRAM RECALL (MILLISECONDS)
          JA      JOB ADVANCE (MILLISECONDS)
          JS      JOB SCHEDULER (SECONDS)
```

```
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
T = IN   INPUT     QN = OP   ORIGINAL PRIORITY
N = OP   ORIGINAL PRIORITY
          RO   ROLLOUT        LP   LOWER BOUND FOR PRIORITY AGE
          OT   OUTPUT         UP   UPPER BOUND FOR PRIORITY AGE
                              IN   AGE INCREMENT

TCSC=CC.
          SET ASSUMED MAGNETIC TAPE CHARACTER SET CONVERSION.
     CC = 63   63 CHAR SET      64   64 CHAR SET
TCVM=CC.
          SET ASSUMED MAGNETIC TAPE CONVERSION MODE.
     CC = NL   NONE             AS   ASCII
          US   USASI            EB   EBCDIC

TDEN=CC.
          SET ASSUMED TAPE DENSITY ACCORDING TO CC.
     CC = NL   NONE             HI   556
          LO   200              HY   800
          HD   800              PE   1600

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

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

FORMAT OF RELEASED IPRDECKS

```
IPRDECK
TDEN=HY.
LOCK.
EI200.
VALID.
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.
DSD,3,AUTO.
```

```
IPRDCK1
TDEN=HY.
LOCK.
EI200.
VALID.
TRANEX.
BATCHIO.
TELEX.
MAGNET.
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,AUTO:X.ISF.
DSD,3,AUTO.
```

```
IPRDCK2
TDEN=HY.
LOCK.
VALID.
ACCOUNT.
AUTOROLL.
PRIORITY.
BATCHIO.
TELEX.
EI200.
TRANEX.
MAGNET.
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.
```

5.2 ACCOUNT.

Format	Default	Significance
ACCOUNT.	enabled	Use ACCOUNT. and VALID. to specify whether or not ACCOUNT card processing and account validation are to occur during system operation. The default status of ACCOUNT. is to check the job stream for an ACCOUNT card. This is the usual status if running under a production environment. The system assumes the ACCOUNT feature is enabled if there is no ACCOUNT entry in the current IPRDECK.

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

	ACCOUNT. enabled	ACCOUNT.disabled
VALID. enabled	An ACCOUNT entry in the job stream must follow the JOB entry; it will be validated. If a CHARGE entry is required†, it must follow the ACCOUNT entry; it will also be validated.	It is necessary that a job stream include an ACCOUNT entry, which is issued to the dayfile but is not validated.
VALID. disabled	It is not necessary that a job stream include an ACCOUNT entry but it is validated if it is available in the job stream. If a CHARGE entry is required†, it must follow the ACCOUNT entry; it will be validated, if available.	It is not necessary that a job stream include an ACCOUNT entry. If it is available in the job stream, it is issued to the dayfile; but it is not validated.

5.3 VALID.

Format	Default	Significance
VALID.	enabled	Use this entry to enable or disable user validation. Disabling this option causes the ACCOUNT card to be ignored even if it is present. This allows jobs to run with no VALIDUX (account validation) file. Account validation is normally enabled when running in a production environment. Refer to the ACCOUNT. and VALID. relationship in section 5.2 The system assumes that VALID. is enabled if there is no entry in the current IPRDECK.

† When an ACCOUNT entry is validated, part of the validation parameters indicate whether a user is required to have a CHARGE entry.

5.4 AUTOROLL.

Format	Default
AUTOROLL.	enabled

Significance

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.

5.5 BATCHIO.

Format	Default
BATCHIO.	enabled

Significance

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.

5.6 DEBUG.

Format	Default
DEBUG.	disabled

Significance

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.

5.7 JOB CONTROL INFORMATION

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

5.7.1 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 1.0, BATCHIO, and TELEX 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 5.7.2). 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.

5.7.2 CONTROL POINT CONTROL

The fourth word in the job control 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; the second is discussed in section 5.7.3.

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.

Whenever a TELEX 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.

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.

5.7.3 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 TELEX 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 that type; AM parameter in SERVICE entry

Bytes 3 and 4 — Not used

Byte 0 only has meaning for TELEX origin jobs. Byte 1 for TELEX 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 TELEX 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.

5.7.4 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.

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

JOB ORIGIN TYPE	QUEUE TYPE	QUEUE PRIORITY			INCREMENT	TIME SLICE		INITIAL CPU PRIORITY
		ENTRY PRIORITY	LOWER BOUND PRIORITY	UPPER BOUND PRIORITY		CPU	CM	
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
1	10	200	10	10

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 TELEX origin jobs. The TELEX 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 TELEX input queue entry priority is given to a job when it is entered into the system. Thus, because the TELEX 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 (TELEX 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 (TELEX rollout entry priority minus input queue lower bound priority). This is only true if there are other jobs in the TELEX input queue.

- The second and each succeeding time a job exceeds its time slice, its priority is set to the TELEX 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 TELEX 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 TELEX 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 swapping too often, the time slices for TELEX 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.

5.8 QUEUE

Format

QUEUE, ot, qt, qp$_1$xxxx, . . . , qp$_n$xxxx.

Significance

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	TELEX
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 $\leq$ xxxx $\leq$ 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.

5.9 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	TELEX
EI	Export/Import (remote batch)
MT	Multiterminal

pi Service Limits

PRxx CPU priority ($01 \le xx \le 70_8$). 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 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 scheduling a job to a control point exceeds the field length specified for its job origin type, 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.

FCxxxxxx Number of permanent files allowed. The value of xxxx can range from 1 to 777777 octal. The value entered is rounted down to the nearest 100_8.

CSxxxxxx Cumulative size in PRUs allowed for all indirect access permanent files. The maximum value that can be entered is 777777 octal. This value is multiplied by 100 octal to determine the actual limit allowed.

FSxxxxxx Size in PRUs allowed for individual indirect access permanent files. The maximum value that can be entered is 777777 octal. This value is rounded down to the nearest 1000 octal PRUs.

5.10 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. Note that the scheduler may also be called at other times.

CRxxxx CPU program recall period in milliseconds. This parameter specifies the amount of time a job remains in recall when an RCL request is placed in RA+1.

ARxxxx PPU auto recall interval in milliseconds. This parameter specifies the time interval at which PPUs in auto recall are recalled.

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.

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.

5.11 DSD

Format

DSD, level, command...

Significance

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, such as a colon.

Example:

DSD, 0, MAI:X. ISF.

5.12 EI200.

Format	Default
EI200.	enabled

Significance

Use this entry to specify whether or not Export/Import 1.0 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 Subsystem (TELEX entry).

The system assumes that Export/Import is to be initialized if there is no entry in the current IPRDECK.

5.13 LOCK.

Format	Default	Significance
LOCK.	unlocked	Use this entry to specify whether the 6612 system console is to be locked or unlocked.

locked — The 6612 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.
TIM.hh.mm.ss.
DISABLE,VALIDATION.
ENABLE,VALIDATION.
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.

5.14 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.

5.15 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

5.16 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.

5.17 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.

5.18 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. HI — 556 bpi (7 track) HY — 800 bpi (7 track) LO — 200 bpi (7 track) NL — None HD — 800 cpi (9 track) PE — 1600 cpi (9 track) The system assumes that the density is HY if there is no TDEN entry in the current IPRDECK.

5.19 TELEX.

Format	Default	Significance
TELEX.	enabled	Use this entry to specify whether or not the Time-Sharing Subsystem 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.

5.20 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.

5.21 TCSC

Format	Default	Significance
TCSC=cc.	63	Use this entry to specify the magnetic tape character set conversion to be assumed during system operation. 63 63-character set 64 64-character set The system assumes 63-character set conversion if there is no TCSC entry in the current IPRDECK.

5.22 TCVM

Format	Default	Significance
TCVM=cc.	AS	Use this entry to set the tape conversion mode to be assumed during system operation. NL None AS ASCII 9-track conversion US USASI 9-track conversion EB EBCDIC 9-track conversion

KRONOS 2.1 1

The KRONOS 2.1 base package consists of the following products; modification information for these products is in the sections indicated.

Operating System: KRONOS 2.1 operating system, Export/Import 1.0, Modify 1.0, Text Editor, Time-Sharing Subsystem	part III, section 1.1
COMPASS 3.0	part III, section 1.2
Record Manager 1.0	part III, section 1.3
Direct Access 1.0	part III, section 1.4
Indexed Sequential 2.0	part III, section 1.5
Update	part III, section 1.6
SCOPE 3.4 Texts	part III, section 1.7

The maintenance tools, which must be ordered separately, are described in part III, section 1.8

1.1 OPERATING SYSTEM

1.1.1 RELEASE DESCRIPTION

HARDWARE REQUIREMENT

The minimum hardware requirement for KRONOS 2.1 is:

CDC CYBER 70/Model 72-12
Seven PPUs
6.0 million words of mass storage
One line printer
One card reader
Two 607, 657, or 659 magnetic tape units

The released KRONOS 2.1 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 1.1.5. Generally, the minimum configuration only supports the Time-Sharing Subsystem (including the Time-Sharing Executive (TELEX), BASIC 2.1, Time-Sharing FORTRAN and Text Editor). To use additional subsystems and products, such as Export/Import 1.0 and KRONOS Transaction Subsystem, additional memory is necessary.

The target hardware configuration for KRONOS 2.1 is:

CDC CYBER 70/model 73-14
Ten PPUs
One 512 line printer
One 405 card reader
One 415 card punch
One 844-2 disk system (or one 6638 disk (with 10037) plus one 841-3 disk system)
Four 65x magnetic tape units (x = 7 or 9)
Two 6676 and one 6671 communication multiplexers

CORRECTIONS

All appropriate corrective code through the KRONOS 2.0 PSR level 12 minitape release is included in the release of KRONOS 2.1.

DEFICIENCIES

The deficiencies for the KRONOS 2.1 system will be supplied with the release materials.

1.1.2 INSTALLATION PROCEDURE

The general procedure to install, modify, and update the KRONOS 2.1 operating system (as well as Export/Import 1.0, Modify, Text Editor, and Time-Sharing Subsystem) is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). Note the following additional procedure. The materials necessary to do so are:

KPL2 contains the jobs to modify, update, install, and verify installation; the job required for the operating system is KR1.

KR1	Use to modify installation parameters, modify CMRDECKs, IPRDECKs, and LIBDECK, add PSR updates to source code, and create a new deadstart tape and KPL1.
BCS2	Use to collect the binary card deck for the 844-2 subsystem controlware program BCS and generate directives for later use as input to the procedure file GENSYS. Note the additional information and alternate procedure in this section on the 844-2 subsystem controlware.
INS1	Use to update installation decks on KPL2.

KPL1 contains the source code.

ADDITIONAL PROCEDURE

Note the following additional information when modifying the KRONOS 2.1 system.

1. If modifying a parameter listed in part II, section 1.1.3, list its deck to obtain its sequence number.

2. Run a job similar to the following example to place modifications on the SYSMOD file and to create files CMRDECK, IPRDECK, and LIBDECK for job KR1.

```
JOB,CM10000,P30,T100.
ACCOUNT card.
COPYBR,INPUT,SYSMOD.
COPYBR,INPUT,CMRDECK.
COPYBR,INPUT,IPRDECK.
COPYBR,INPUT, LIBDECK.
SAVE,SYSMOD,CMRDECK,IPRDECK, LIBDECK.
7/8/9
*DECK COMSPFM            )
*D,nnn.                  |
 DFPT EQU  3RDI1         |
*EDIT COMSPFM            |
*DECK PPCOM              |
*D,nnn.                  }      Local modifications and installation
 NPFS EQU  3             |      parameter (assembly constants) changes
*EDIT PPCOM              |      from part III, section 1.1.3
.                        |
.                        |
.                        )
7/8/9
CMRDECK                  )
EQ0=DB,ON,1,0,3.         |
EQ20=LP,ON,5,0,12.       |      Entire CMRDECK using entries from
.                        }      part II, section 4
.                        |
.                        )
7/8/9
IPRDECK                  )
SERVICE,TX,NJ100.        |      Entire IPRDECK using entries from part II,
.                        }      section 5
.                        |
.                        )
7/8/9
LIBDECK                  )
*CM PP/PFM,CIO,1AJ       |      Entire LIBDECK using entries from part III,
*PROC TEXT/PFD           }      section 1.1.5
.                        |
.                        |
.                        |
6/7/8/9                  )
```

3. If adding a CMRDECK, IPRDECK, or LIBDECK, add the appropriate cards as indicated in job KR1 in part III, section 1.1.4. If adding all three decks, the cards would be:

```
GET(CMRDECK, IPRDECK, LIBDECK)
•
•
•
*FILE  CMRDECK
*FILE  IPRDECK
*FILE  LIBDECK
```

4. Run job KR1 to add the modifications to the deadstart tape, as described in part I, section 3.2.

844-2 SUBSYSTEM CONTROLLWARE

DESCRIPTION OF RELEASED CONTROLWARE

The 844-2 subsystem controlware that is released on the KRONOS 2.1 deadstart tape is identified by card deck part number 5382700. It is appropriate for the following controllers.

An 844-2 controller with a serial number above DT209C167

An 844-2 controller with a serial number below DT209167 that includes FCO CA32618

If the controller serial number is below DT209C167, but FCO CA32618 is not available, it is necessary to obtain from Software Distribution and install on the deadstart tape (using BCS2) controlware that contains a one-card modification. The one-card modification allows the controlware to run without FCO CA32618.

Figure III-1-1 illustrates which controlware supports the 844-2 controllers. SA refers to a single-access controller; DA refers to a dual-access controller. For example, if a single-access controller with serial number DT209C166 is available and the controlware does not include FCO CA32618, then the controlware with modification must be ordered. All other combinations are supported by the released version of the controlware.

Controller Serial Number (DT209Cnnn)	Controlware with Modification (must be ordered)		Controlware without Modification (as released on deadstart tape)	
If nnn is 166 or less, and FC0 CA32618 is not included in controlware	SA	yes	SA	no
	DA	no	DA	yes
If nnn is 166 or less, and FC0 CA32618 is included in controlware	SA	yes	SA	yes
	DA	no	DA	yes
If nnn is 167 or higher, and FC0 CA32618 is included in the controlware	SA	yes	SA	yes
	DA	no	DA	yes

Figure III-1-1. 844-2 Subsystem Controlware Requirements

PROCEDURE TO IDENTIFY AVAILABLE CONTROLWARE

If it is necessary to determine which controlware card deck is available, examine either the last two data cards in the card deck 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.

Columns on Last Card	Memory Locations Contained in Columns	Contents of Controller Memory Locations
14 and 15	0FFD	Product ID (hexadecimal equivalent of 710A, for example)
12 and 13	0FFC	Right four digits of the eight-digit controlware part number
10 and 11	0FFB	Left four digits of the eight-digit controlware part number
8 and 9	0FFA	Program code only
6 and 7	0FF9	Start of FCO list; each location contains the hexadecimal equivalent of a decimal FCO number. The list ends with 00.
4 and 5	0FF8	
3 of last card and 77 of previous card	0FF7	
75 and 76 of previous card	0FF6	
•	•	
•	•	
•	•	

ALTERNATE PROCEDURE TO MODIFY BCS

It is possible to use the following procedure, instead of job BCS2, to update the 844-2 BCS controlware on the deadstart tape.

1. Type the following at the system console, pressing CR after each entry.

```
X.DIS.
COMMON(ZZZPSBF,ZZZZLDF)
COMMON(SYSTEM)
GTR(SYSTEM,BCS)PPU/BCS
ASSIGN(CR,CR)
```

2. Place the following deck in the system card reader; ready the unit.

```
844-2 controlware deck
7/8/9
*REPLACE  PPU/BCS
6/7/8/9
```

3. Type the following at the system console, pressing CR after each entry.

```
COPYBR(CR,X)
COPYBR(CR,ZZZZLDF)
RETURN(CR)
REWIND(X)
O26.
READ.BCS.
```

4. Move the insert (by repeatedly pressing the left parenthesis key) to follow the line containing the right parenthesis in column 1.

5. Type the following at the system console, pressing CR after each entry.

```
DEL.                    This deletes the remainder of the BCS record in the system
                        library (all except the 77 and 52 tables).
READN.X.
REWIND.
DIS.
COPYBR(X,ZZZPSBF)
DROP.
```

1.1.3 INSTALLATION PARAMETERS

CMRDECK, IPRDECK, LIBDECK, and other installation parameters listed in this section can all be modified using the KR1 job.

CMRDECK modifications are described in part II, section 4; IPRDECK modifications are described in part II, section 5; and LIBDECK modifications are described in part III, section 1.1.5.

The following parameters are listed alphabetically according to the deck they are in.

COMSACC PARAMETERS

List CALLSYS to obtain a listing of COMSACC.

Parameter	Released Default Value	Significance
APFN	VALIDUX	This is a micro definition that specifies the name of the file that is the KRONOS 2.1 version of the KRONOS 2.0 VALIDUS file. Refer to part IV, section 1.1, for further information on VALIDUX.
AUFN	VALINDX	This is a micro definition that specifies the name of the available user indices file. This file was part of VALIDUS under KRONOS 2.0; it is a separate file under KRONOS 2.1. Refer to part IV, section 1.1, for further information on VALINDX.

COMSEXP PARAMETERS

List CALLSYS to obtain a listing of COMSEXP.

N.PORTS	20_8	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.

Parameter	Released Default Value	Significance
T. HUNG	400_{10} milliseconds	This is the line timeout delay.

COMSPFM PARAMETERS

List CALLSYS to obtain a listing of COMSPFM.

DFPT	MD1	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		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.

COMSPRO PARAMETERS

List CALLSYS to obtain a listing of COMSPRO.

PPFN	PROFILO	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.

COMSREM PARAMETERS

List CALLSYS to obtain a listing of COMSREM.

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	20_8	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. 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 changing this parameter because lost data can occur if system activity increases.
VMPL	40_8	

COMSSSJ PARAMETER

List CALLSYS to obtain a listing of COMSSSJ.

ART	4 minutes	This value specifies the default for the NA parameter on the ATTACH control card 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.

COMSDSL PARAMETER

List CALLSYS to obtain a listing of COMSDSL.

LTKL	20_8	If a mass storage device is not initialized during deadstart (using the INITIALIZE entry described in part II, section 4.4.11), 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 KRONOS 2.1 Operator's guide.

MTR PARAMETER

List MTR to obtain a listing of this deck.

Parameter	Released Default Value	Significance
MSLC	998_{10}	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.

PPCOM PARAMETERS

List SYSTEXT to obtain a listing of PPCOM.

NMSD	20_8	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 4.4.4) and the MSAL entry in IPRDECK (part II, section 5), or DSD command (KRONOS 2.1 Operator's Guide). TEMP designates the number of mass storage devices that can contain temporary files. MSAL designates the mass storage devices to be used for rollout.
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.

RESEX PARAMETER

List RESEX to obtain a listing of RESEX.

Parameter	Released Default Value	Significance
PL	2 minutes	This is the length of time that a job waiting for an auxiliary device is kept in the timed/event queue before retrying assignment.
TL	1 minute	This is the length of time that a tape request is kept in the timed/event queue for a tape to become available before retrying assignment.

1.1.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

KR1

```
KR1
KR1,CM61000,P30,T7000.
*ACCOUNT CARD.
*GET,MOD=SYSMOD.
COMMON,OPL.
MODIFY,Z,F,N,C=0,LO=E./*READ,MOD,*
RETURN,OPL.
CATALOG,NPL,R.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL1,W,FI=KRONOS*2P1,MT,D=HY,F=I)
COPYEI,NPL,KPL1,V.
RETURN,KPL1.
RENAME,OPL=NPL.
MODIFY,Z,Q,CS=0,CB=SYST./*EDIT,SYSTEXT
MODIFY,Z,Q,CG=SYST./*EDIT,ADC.VFYLIB
MODIFY,Z,Q,CG=SYST,CB=REL./*EDIT,CPUREL.WRITEB
COMMON,SYSTEM.
GTR,SYSTEM,OLD.ULIB/SYSLIB
LIBEDIT,B=REL,I=0,L=0.
LIBGEN,F=NEW,P=SYSLIB.
MODIFY,Z,L=0./*EDIT,GENFILS.GENVAL
REWIND,SYST,LGO,SYSLIB.
LIBEDIT,P=SYSTEM,N=NSYS,B=0,L=0.
CATALOG,NSYS,R,U.
REQUEST,NDS,LB=KU,F=I.
COPYEI,NSYS,NDS,V.
*WEOR
*FILE      SYST
*FILE      LGO
*FILE      COMPILE
*FILE      SYSLIB
*WEOF
```

If adding a CMRDECK, IPRDECK, and LIBDECK, add the following cards as indicated.

```
GET(CMRDECK,IPRDECK,LIBDECK)
```

```
*FILE      CMRDECK
*FILE      IPRDECK
*FILE      LIBDECK
*WEOF
```

BCS2

```
BCS2
BCS2,CM15000,T77.
*ACCOUNT CARD
COMMENT.  THIS JOB ADDS THE 844 BUFFER CONTROLLER BINARY DECK
COMMENT.  TO THE SYSTEM COMMON FILE ZZZPSBF FROM THE
COMMENT.  DISTRIBUTED BINARY CARD DECK. (PART NUMBER 53088900).
COMMENT.  IT ALSO ADDS THE NECESSARY LIBEDIT DIRECTIVES TO THE
COMMENT.  SYSTEM COMMON FILE ZZZZLDF.
COMMON,ZZZPSBF,ZZZZLDF.
SETCORE.
RBR,INPUT,BCS.   ADD 77 AND 52 TABLE
WBR,ZZZPSBF.
COPYBR,INPUT,ZZZZLDF.
WRITEF(ZZZPSBF)
BKSP,ZZZPSBF.
*WEOR
*         844 FIRMARE DECK.
*WEOR
*REPLACE PPU/BCS
*WEOF
```

INS1

```
INS1
INS1,CM45000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB MODIFYS THE INSTALLATION DECKS RELEASE TAPE AND
*CREATES A NEW KPL2 PROGRAM LIBRARY.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL2,R,FI=INSTALLDECKS*2P1,MT,D=HY,F=I)
COPY,KPL2,OPL.
UNLOAD,KPL2.
RETURN,KPL2.
MODIFY,F,N,C=0,LO=E.
CATALOG,NPL,R.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL2,W,FI=INSTALLDECKS*2P1,MT,D=HY,F=I)
COPY,NPL,KPL2,V.
*WEOR
*/        PLACE *INSTALL DECKS* MODSET HERE.
*WEOF
```

1.1.5 LIBDECK

LIBDECK is a directive record on the deadstart tape. It is read by SYSEDIT during the system load to determine the following types of information.

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
What the default user libraries for overlay programs are	*LIBRARY
Which records are procedure files	*PROC
Which programs should use SCOPE 3.4 parameter processing	*SC

The LIBDECK released on the KRONOS 2.1 deadstart tape is:

```
LIBDECK
*CM         PP/CIO,2CA,2CB,2CC,2CD,2CE,2CF,2CG,2CH
*CM         PP/1AJ,TCS,3AE,LDR
*CM         PP/1CJ,1MA,0BF,0DF,0AV,0RP
*CM         PP/LFM
*CM         PP/1RI,3RH,1RO,3RP,3RQ
*CM         PP/PFM,3PA,3PB,3PD,3PG,3PI
*CM         PP/1TA,1TO,3TI,3TJ
*CM         PP/1MT,3MD,3MF,3ML
*CM         PP/CHK,CLL
*CM         PP/1LS
*CM         PP/1IO,1SJ,1SP,1BA
*CM         PP/1DL,9AX,9A2,9A3                    (DSD RELATED)
*LIBRARY    BASLIB/BASIC
*PROC       LIBMOD
*PROC       GENVAL
*PROC       GENHELP
*PROC       GENFILS
*PROC       GENINSD
*PROC       GENSYS
*SC         OVL/FILE,UPDATE,COMPASS
*SC         OVL/RUN23,RUN,ALGOL,COBOL,FTN,SIMI5,SIMULA,SORTMRG
```

MODIFICATION OF LIBDECK

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 TSRUN, 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 used for highly used CPU programs, such as TSRUN and PPU programs.

3. The 6638 is a high transfer rate, medium positioning time device. It could be used as a system device.

4. The 841 is a low transfer rate, fast positioning time device. It could be used for small, highly used programs, such as PPU overlays.

5. 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.

LIBDECK DIRECTIVES

The directives that can be in LIBDECK are described. Other valid input directives to SYSEDIT (*DELETE and *IGNORE, for example) are described in section 5 of the KRONOS 2.1 Reference Manual.

*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. There are some programs that will automatically reside in central memory. These are:

1. Mass storage drivers
2. Those 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.

Note that any addition to central memory allows less space for user jobs.

*MS DIRECTIVE

This directive identifies which system library routines are to 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 a *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 (for example, if the system device is a 6638, placing PPU routines on an 841).

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 some cases of error recovery. These exception are:

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 routine on an alternate system device, only that routine will subsequently be accessed from the system device.

Specify the devices that are to be used as alternate system devices with the ASR entry in the CMRDECK (part II, section 4.4.8); specify the routines that are to reside on each alternate system device in LIBDECK.

The format of the *AD LIBEDIT directive to SYSEDIT is:

$$\boxed{\text{*AD, nn, } ty_1/rec_1, ty_2/rec_2, \ldots, ty_n/rec_n.}$$

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	6000 series PPU absolute

If a record type other than ABS, OVL, or PP is specified, the message ILLEGAL CM/AD RESIDENCE, MUST BE PP, OVL, OR ABS 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 CPU routines residing there will be loaded directly by central monitor function LCEM. This causes the transfer of a program for 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.

*LIBRARY DIRECTIVE

Use the *LIBRARY directive to specify user libraries from which to satisfy externals for those product set binaries that do not contain LDSET tables (such as BASIC binaries). The LDSET control card can also be used to specify additional libraries. It is described in section 5 of the KRONOS 2.1 Reference Manual.

Any binaries generated by a common product (such as COBOL 4.0 or FORTRAN Extended 4.0) contain LDSET loader tables identifying the user libraries that satisfy externals.

The format of the directive is:

*LIBRARY, ulib/ovl_1, ovl_2, . . . , ovl_n .

ulib Name of user library for the overlays specified

ovl_i Overlay program

*PROC DIRECTIVE

Specifying a TEXT record as a procedure file allows a user to use the CALL statement to insert this file (consisting of a group of control cards) at a specific location in his job's control card 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 KRONOS 2.1 Reference Manual.

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 control card images in a certain program are to be processed in SCOPE format rather than KRONOS format. The formats are defined under control card format in section 4 of the KRONOS 2.1 Reference Manual.

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 SCOPE format

1.1.6 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 KRONOS 2.1 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.

JOB, CM60000, T100	Refer to the KRONOS 2.1 Reference Manual for control card format
ACCOUNT, usernum, passwrd.	
CALL(GENHELP)	Initiates the procedure file
7/8/9	
*BUILD HELPPL	LIBEDIT input directive (refer to KRONOS 2.1 Reference Manual)
6/7/9	

The procedure file GENHELP is as follows:

GENHELP	File name card
RETURN(HLP)	Release file HLP
MODIFY(Z, C=HLP)/*NOSEQ/*EDIT, CMDFILE	Get Modify input directives following the right parenthesis. Edit the deck CMFILE and write unsequenced compile output to file HLP.
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(CMFILE/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.

1.1.7 LIBMOD PROCEDURE FILE PROCEDURE

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.

CALL(LIBMOD(LIB=SYSBLIB,OLD=SYSTEM, LGO=CHGS)

RETURN	Make sure files Q1 and Q2 are cleared so these file names can be used later.
GTR	Get the user library record SYSLIB from SYSTEM and write it to file Q1.
LIBEDIT	Take the changes to be made to SYSLIB from the local file CHGS, combine them, and write the result to file Q2.
LIBGEN	Get the updated record from Q2, generate an updated library, and write it to file SYSLIB.
LIBEDIT	Get the updated library record from SYSLIB, replace the record SYSLIB on SYSTEM with it, and write a new system file to file NEW.

1.1.8 SYSTEM ORGANIZATION NOTES

To become more familiar with the operation of KRONOS, use Modify (refer to the Modify Reference Manual) and DOCMENT (refer to the KRONOS 2.1 Reference Manual) to obtain documentation of the programs from the system program library file (KPL1).

To use these programs, load the OPL from KPL1 into mass storage with the STAGE control card using the following sequence.

JOB card	Refer to the KRONOS 2.1 Reference Manual for format
ACCOUNT, usernum, passwrd.	
STAGE(OPL, VSN=xxxxxx)	xxxxxx is the VSN of the released KPL1 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 KRONOS 2.1 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 KRONOS. 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 card that requires no input, such as the MODIFY and STAGE commands, may be entered directly from the console and are executed 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 time-sharing executive program.

```
JOB, CM50000, T77.

ACCOUNT, usernum, passwrd.

MODIFY(LO=E, Z)/*EDIT, DIS/*EDIT, TELEX

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 KRONOS 2.1 Time-Sharing User's Guide).

```
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 KRONOS 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 card 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 common 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.)

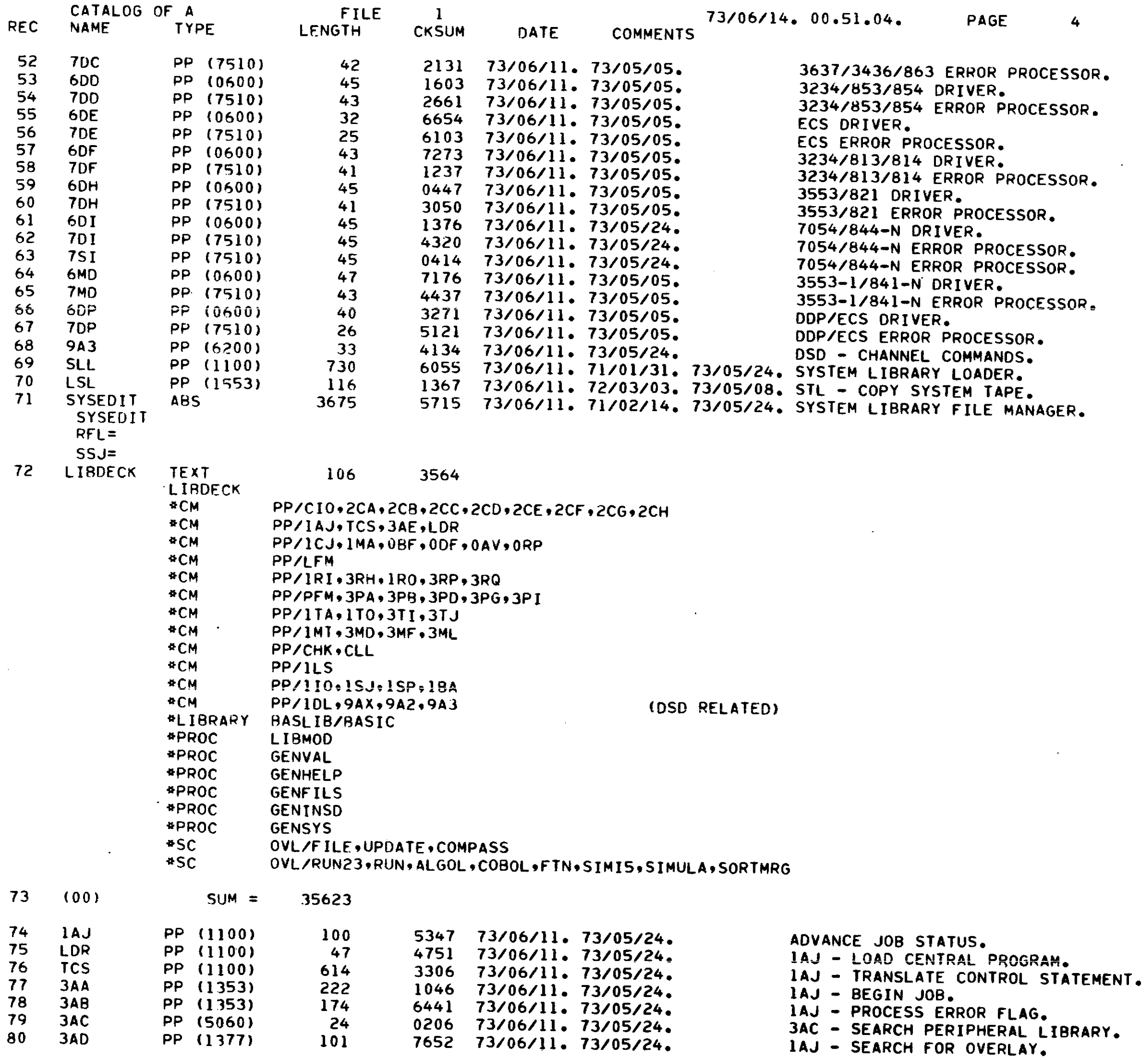
Refer to the Modify Reference Manual for further information on listing various programs. The following three parameters can be especially helpful.

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 cards, the first record on the INPUT file, are read as directive cards, 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). Note that 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.

1.1.9 CATALOG OF SYSTEM



```
        CATALOG OF A                    FILE      1                              73/06/14. 00.51.04.      PAGE       4
REC     NAME       TYPE               LENGTH    CKSUM      DATE       COMMENTS

 52     7DC        PP (7510)             42      2131   73/06/11. 73/05/05.             3637/3436/863 ERROR PROCESSOR.
 53     6DD        PP (0600)             45      1603   73/06/11. 73/05/05.             3234/853/854 DRIVER.
 54     7DD        PP (7510)             43      2661   73/06/11. 73/05/05.             3234/853/854 ERROR PROCESSOR.
 55     6DE        PP (0600)             32      6654   73/06/11. 73/05/05.             ECS DRIVER.
 56     7DE        PP (7510)             25      6103   73/06/11. 73/05/05.             ECS ERROR PROCESSOR.
 57     6DF        PP (0600)             43      7273   73/06/11. 73/05/05.             3234/813/814 DRIVER.
 58     7DF        PP (7510)             41      1237   73/06/11. 73/05/05.             3234/813/814 ERROR PROCESSOR.
 59     6DH        PP (0600)             45      0447   73/06/11. 73/05/05.             3553/821 DRIVER.
 60     7DH        PP (7510)             41      3050   73/06/11. 73/05/05.             3553/821 ERROR PROCESSOR.
 61     6DI        PP (0600)             45      1376   73/06/11. 73/05/24.             7054/844-N DRIVER.
 62     7DI        PP (7510)             45      4320   73/06/11. 73/05/24.             7054/844-N ERROR PROCESSOR.
 63     7SI        PP (7510)             45      0414   73/06/11. 73/05/24.             7054/844-N ERROR PROCESSOR.
 64     6MD        PP (0600)             47      7176   73/06/11. 73/05/05.             3553-1/841-N DRIVER.
 65     7MD        PP (7510)             43      4437   73/06/11. 73/05/05.             3553-1/841-N ERROR PROCESSOR.
 66     6DP        PP (0600)             40      3271   73/06/11. 73/05/05.             DDP/ECS DRIVER.
 67     7DP        PP (7510)             26      5121   73/06/11. 73/05/05.             DDP/ECS ERROR PROCESSOR.
 68     9A3        PP (6200)             33      4134   73/06/11. 73/05/24.             DSD - CHANNEL COMMANDS.
 69     SLL        PP (1100)            730      6055   73/06/11. 71/01/31. 73/05/24. SYSTEM LIBRARY LOADER.
 70     LSL        PP (1553)            116      1367   73/06/11. 72/03/03. 73/05/08. STL - COPY SYSTEM TAPE.
 71     SYSEDIT    ABS                 3675      5715   73/06/11. 71/02/14. 73/05/24. SYSTEM LIBRARY FILE MANAGER.
         SYSEDIT
         RFL=
         SSJ=
 72     LIBDECK    TEXT                 106      3564
                   LIBDECK
                   *CM          PP/CIO,2CA,2CB,2CC,2CD,2CE,2CF,2CG,2CH
                   *CM          PP/IAJ,TCS,3AE,LDR
                   *CM          PP/1CJ,1MA,0BF,0DF,0AV,0RP
                   *CM          PP/LFM
                   *CM          PP/1RI,3RH,1RO,3RP,3RQ
                   *CM          PP/PFM,3PA,3PB,3PD,3PG,3PI
                   *CM          PP/1TA,1TO,3TI,3TJ
                   *CM          PP/1MT,3MD,3MF,3ML
                   *CM          PP/CHK,CLL
                   *CM          PP/1LS
                   *CM          PP/1IO,1SJ,1SP,1BA
                   *CM          PP/1DL,9AX,9A2,9A3                      (DSD RELATED)
                   *LIBRARY     BASLIB/BASIC
                   *PROC        LIBMOD
                   *PROC        GENVAL
                   *PROC        GENHELP
                   *PROC        GENFILS
                   *PROC        GENINSD
                   *PROC        GENSYS
                   *SC          OVL/FILE,UPDATE,COMPASS
                   *SC          OVL/RUN23,RUN,ALGOL,COBOL,FTN,SIMIS,SIMULA,SORTMRG

 73     (00)            SUM =      35623

 74     1AJ        PP (1100)            100      5347   73/06/11. 73/05/24.             ADVANCE JOB STATUS.
 75     LDR        PP (1100)             47      4751   73/06/11. 73/05/24.             1AJ - LOAD CENTRAL PROGRAM.
 76     TCS        PP (1100)            614      3306   73/06/11. 73/05/24.             1AJ - TRANSLATE CONTROL STATEMENT.
 77     3AA        PP (1353)            222      1046   73/06/11. 73/05/24.             1AJ - BEGIN JOB.
 78     3AB        PP (1353)            174      6441   73/06/11. 73/05/24.             1AJ - PROCESS ERROR FLAG.
 79     3AC        PP (5060)             24      0206   73/06/11. 73/05/24.             3AC - SEARCH PERIPHERAL LIBRARY.
 80     3AD        PP (1377)            101      7652   73/06/11. 73/05/24.             1AJ - SEARCH FOR OVERLAY.
```

```
        CATALOG OF A            FILE     1                          73/06/14. 00.51.04.      PAGE      5
REC     NAME      TYPE        LENGTH    CKSUM     DATE      COMMENTS

 81     3AE       PP (5060)      200     5136   73/06/11. 73/05/24.            3AE - LOAD COPY ROUTINES.
 82     3AF       PP (5060)      134     5276   73/06/11. 73/05/24.            3AF - SPECIAL ENTRY POINT PROCESSING.
 83     1CJ       PP (1100)      305     3111   73/06/11. 71/02/20. 73/05/08. COMPLETE JOB.
 84     1RI       PP (1100)      262     6171   73/06/11. 73/05/24.            ROLLIN JOB.
 85     3RG       PP (5075)       32     5105   73/06/11. 73/05/24.            1RI - PROCESS SPECIAL ENTRY POINT JOBS.
 86     3RH       PP (5075)      207     2414   73/06/11. 73/05/24.            1RI - PROCESS TXOT JOBS.
 87     1RO       PP (1100)      270     3034   73/06/11. 73/05/24.            ROLLOUT JOB.
 88     3RP       PP (4712)      240     1374   73/06/11. 73/05/24.            1RO - PROCESS TXOT JOBS.
 89     3RQ       PP (4712)       33     1154   73/06/11. 73/05/24.            1RO - PROCESS MTOT JOBS.
 90     1RP       PP (1100)       32     4664   73/06/11. 73/05/24.            PPR - RELOAD PP RESIDENT.
 91     1SJ       PP (1100)      241     6271   73/06/11. 71/02/20. 73/05/24. JOB SCHEDULER.
 92     3SA       PP (3066)       52     0262   73/06/11. 71/02/20. 73/05/24. 1SJ - INITIATE SUB-SYSTEMS.
 93     1SP       PP (1100)      173     5270   73/06/11. 73/05/24.            EVALUATE PRIORITIES.
 94     (00)         SUM =      4435

 95     1DL       PP (1100)       66     5752   73/06/11. 70/12/13. 73/05/08. DISPLAY OVERLAY LOADER.
 96     1DS       PP (1100)      662     2111   73/06/11. 73/05/05.            DSD REQUEST PROCESSOR.
 97     9AA       PP (0000)       45     2217   73/06/11. 73/05/24.            DSD - DISPLAY A - DAYFILE MESSAGES.
 98     9AB       PP (0000)       46     2143   73/06/11. 73/05/24.            DSD - DISPLAY B - SYSTEM STATUS.
 99     9AC       PP (0000)       22     1021   73/06/11. 73/05/24.            DSD - DISPLAYS F, G - CENTRAL MEMORY.
100     9AD       PP (0000)       56     1606   73/06/11. 73/05/24.            DSD - DISPLAY E - EQUIPMENT STATUS TABLE.
101     9AE       PP (0000)       73     7373   73/06/11. 73/05/24.            DSD - DISPLAY E - MASS STORAGE DEVICES.
102     9AF       PP (0000)       62     6572   73/06/11. 73/05/24.            DSD - DISPLAY E - RESOURCE MOUNTING PREVIEW.
103     9AG       PP (0000)       77     6547   73/06/11. 73/05/24.            DSD - DISPLAY E - MAGNETIC TAPES.
104     9AH       PP (0000)       63     2425   73/06/11. 73/05/24.            DSD - DISPLAY H - FILE NAME TABLE.
105     9AI       PP (0000)       55     1317   73/06/11. 73/05/24.            DSD - DISPLAY I - BATCHIO STATUS.
106     9AJ       PP (0000)       47     3461   73/06/11. 73/05/24.            DSD - DISPLAY J - CONTROL POINT STATUS.
107     9AK       PP (0000)       45     3150   73/06/11. 73/05/24.            DSD - DISPLAY K - CENTRAL PROGRAM BUFFER.
108     9AL       PP (0000)       32     1722   73/06/11. 73/05/24.            DSD - DISPLAY N - FILE DISPLAY.
109     9AM       PP (0000)       66     0511   73/06/11. 73/05/24.            DSD - DISPLAY O - TRANSACTION TERMINAL STATUS.
110     9AN       PP (0000)       50     6067   73/06/11. 73/05/24.            DSD - DISPLAY O - SUB CONTROL POINT STATUS.
111     9AO       PP (0000)       77     4036   73/06/11. 73/05/24.            DSD - DISPLAY O - TASK LIBRARY DIRECTORY.
112     9AP       PP (0000)       62     2343   73/06/11. 73/05/24.            DSD - DISPLAY P - PP REGISTERS.
113     9AQ       PP (0000)       42     1316   73/06/11. 73/05/24.            DSD - DISPLAY Q - INPUT/OUTPUT/ROLLOUT QUEUES.
114     9AR       PP (0000)       77     1316   73/06/11. 73/05/24.            DSD - DISPLAY R - REMOTE BATCH STATUS.
115     9AS       PP (0000)       52     5774   73/06/11. 73/05/24.            DSD - DISPLAY S - SYSTEM CONTROL INFORMATION.
116     9AT       PP (0000)       75     1436   73/06/11. 73/05/24.            DSD - DISPLAY T - TIME SHARING STATUS.
117     9AU       PP (0000)       51     1430   73/06/11. 73/05/24.            DSD - DISPLAY Y. - MONITOR FUNCTIONS.
118     9AV       PP (0000)       72     5560   73/06/11. 73/05/24.            DSD - DISPLAY Z - DIRECTORY.
119     9AW       PP (6200)       22     6207   73/06/11. 73/05/24.            DSD - N. SYNTAX TABLE - CHARACTERS A-C, E-N.
120     9AX       PP (6200)       27     7761   73/06/11. 73/05/24.            DSD - N. SYNTAX TABLE. CHARACTERS O-*.
121     9AY       PP (6200)       13     1257   73/06/11. 73/05/24.            DSD - SYSTEM SYNTAX TABLE, *EN*.
122     9AZ       PP (6200)       25     7523   73/06/11. 73/05/24.            DSD - SYSTEM SYNTAX TABLE, *C,D*
123     9A0       PP (6200)       32     2162   73/06/11. 73/05/24.            DSD - SYSTEM SYNTAX TABLE, *I,K,L,M,O*
124     9A1       PP (6200)       37     3761   73/06/11. 73/05/24.            DSD - SYSTEM SYNTAX TABLE, *P,R,S,T,U,V,W*
125     9A2       PP (6200)       34     6206   73/06/11. 73/05/24.            DSD - CENTRAL MEMORY CHANGES.
126     9A4       PP (6200)       41     3713   73/06/11. 73/05/24.            DSD - SEND DAYFILE MESSAGES.
127     9A5       PP (6200)       40     0711   73/06/11. 73/05/24.            DSD - CONTROL POINT REQUESTS.
128     9A6       PP (6200)       26     3631   73/06/11. 73/05/24.            DSD - SUB - SYSTEMS REQUESTS.
129     9A7       PP (6200)       44     0542   73/06/11. 73/05/24.            DSD - TELEX MESSAGE REQUESTS.
130     9A8       PP (6200)       24     5570   73/06/11. 73/05/24.            DSD - BATCHIO REQUESTS.
131     9A9       PP (6200)       24     2336   73/06/11. 73/05/24.            DSD - SYSTEM REQUESTS.
132     9BA       PP (6200)       35     1432   73/06/11. 73/05/24.            DSD - SYSTEM REQUESTS.
133     9BB       PP (6200)       27     6125   73/06/11. 73/05/24.            DSD - JOB CALL REQUESTS.
134     9BC       PP (6200)       21     1264   73/06/11. 73/05/24.            DSD - SYSTEM CONTROL REQUESTS.
135     9BD       PP (6200)       32     6525   73/06/11. 73/05/24.            DSD - ENABLE SYNTAX TABLE.
```

```
      CATALOG OF A               FILE    1                          73/06/14. 00.51.04.      PAGE      6
REC   NAME       TYPE            LFNGTH   CKSUM    DATE      COMMENTS

136   9BE        PP (6200)           33    5117  73/06/11. 73/05/24.            DSD - DISABLE SYNTAX TABLE
137   9BF        PP (6200)           16    2722  73/06/11. 73/05/24.            DSD - ENABLE/DISABLE REQUESTS.
138   9BG        PP (6200)           26    7464  73/06/11. 73/05/24.            DSD - JOB CONTROL REQUESTS.
139   9BH        PP (6200)           33    2724  73/06/11. 73/05/24.            DSD - JOB CONTROL REQUESTS.
140   9BI        PP (6200)           42    6130  73/06/11. 73/05/24.            DSD - JOB CONTROL REQUESTS.
141   9BJ        PP (6200)           43    2237  73/06/11. 73/05/24.            DSD - DISPLAY CHANGE REQUESTS.
142   9BK        PP (6200)           35    1714  73/06/11. 73/05/24.            DSD - FILE CONTROL REQUESTS.
143   9BL        PP (6200)           16    6621  73/06/11. 73/05/24.            DSD - FILE CONTROL REQUESTS.
144   9BM        PP (6200)           35    3342  73/06/11. 73/05/24.            DSD - RESOURCE CONTROL COMMANDS.
145   9BN        PP (6200)           34    6551  73/06/11. 73/05/24.            DSD - ASSIGN VSN TO UNIT.
146   9BO        PP (6200)           22    5632  73/06/11. 73/05/24.            DSD - ENTER TIME.
147   9BP        PP (6200)           42    0677  73/06/11. 73/05/24.            DSD - ENTER DATE.
148   DIS        PP (1100)         1216    3156  73/06/10. 73/05/24.            JOB DISPLAY
149   9EA        PP (6605)           37    6671  73/06/10. 73/05/24.            DIS - DISPLAY A. - DAYFILE.
150   9EB        PP (6605)           16    2354  73/06/10. 73/05/24.            DIS - DISPLAY E.  CENTRAL MEMORY 4 GROUPS OF 5.
151   9EC        PP (6605)           75    5022  73/06/10. 73/05/24.            DIS - DISPLAYS F AND G.  CENTRAL MEMORY 4 GROUPS O
152   9ED        PP (6605)           24    3564  73/06/10. 73/05/24.            DIS - DISPLAY H. - FILE NAME TABLE.
153   9EE        PP (6605)           43    3616  73/06/10. 73/05/24.            DIS - DISPLAY J. - SYSTEM STATUS.
154   9EF        PP (6605)           41    1115  73/06/10. 73/05/24.            DIS - DISPLAY K. - EQUIPMENT STATUS TABLE.
155   9EG        PP (6605)           52    0651  73/06/10. 73/05/24.            DIS - DISPLAY L. - FILE NAME TABLE.
156   9EH        PP (6605)           56    4276  73/06/10. 73/05/24.            DIS - DISPLAY P. - PP REGISTERS.
157   9EI        PP (6605)           32    1514  73/06/10. 73/05/24.            DIS - DISPLAY Q. - INPUT/OUTPUT/ROLLOUT QUEUES.
158   9EJ        PP (6605)           26    3672  73/06/10. 73/05/24.            DIS - DISPLAY V.  CENTRAL MEMORY BUFFER.
159   9EK        PP (6605)           46    5223  73/06/10. 73/05/24.            DIS - DISPLAY Y. - MONITOR FUNCTIONS.
160   9EL        PP (6605)           66    6501  73/06/10. 73/05/24.            DIS - DISPLAY Z. - DIRECTORY.
161   9EM        PP (7266)           24    2440  73/06/10. 73/05/24.            DIS - CPU COMMANDS.
162   9EN        PP (7266)           60    7546  73/06/10. 73/05/24.            DIS - STATEMENT ENTRY.
163   9EO        PP (7266)           30    4650  73/06/10. 73/05/24.            DIS - ENTER REGISTER.
164   9EP        PP (7266)           30    6472  73/06/10. 73/05/24.            DIS - ENTER X REGISTER.
165   9EQ        PP (7266)           45    3201  73/06/10. 73/05/24.            DIS - ENTER MEMORY.
166   9ER        PP (7266)           47    1354  73/06/10. 73/05/24.            DIS - ENTER INSTRUCTION.
167   9ES        PP (7266)           23    0664  73/06/10. 73/05/24.            DIS - CPU PROGRAM INTERFACE COMMANDS.
168   9ET        PP (7266)           44    5014  73/06/10. 73/05/24.            DIS - FIELD LENGTH CHANGES.
169   9EU        PP (7266)           21    4330  73/06/10. 73/05/24.            DIS - MISCELLANEOUS COMMANDS.
170   9EV        PP (7266)           37    1350  73/06/10. 73/05/24.            DIS - MISCELLANEOUS COMMANDS.
171   9EW        PP (7266)           43    7351  73/06/10. 73/05/24.            DIS - INTERPRET KEYSET MESSAGE.
172   9EX        PP (7266)           24    3147  73/06/10. 73/05/24.            DIS - INTERPRET MORE MESSAGES.
173   9EY        PP (7266)           24    4100  73/06/10. 73/05/24.            DIS - CALL PPU PROGRAM.
174   026        PP (1100)         1243    3736  73/06/11. 73/05/24.            CONSOLE TEXT EDITOR.
175   9GA        PP (6201)           70    0314  73/06/11. 73/05/24.            026 - FILE COMMANDS.
176   9GB        PP (6201)           66    2016  73/06/11. 73/05/24.            026 - LINE ENTRY AND DATA MOVE.
177   9GC        PP (6201)           62    0415  73/06/11. 73/05/24.            026 - DISPLAY, TAB, DUP AND SCAN CONTROL.
178   9GD        PP (6201)           50    1444  73/06/11. 73/05/24.            026 - LINE SEARCH COMMANDS.
179   9GE        PP (6201)           76    1175  73/06/11. 73/05/24.            026 - RECORD SEARCH COMMANDS.
180   9GF        PP (6201)           67    5417  73/06/11. 73/05/24.            026 - REPLACE COMMANDS.
181   9GG        PP (6201)           55    3504  73/06/11. 73/05/24.            026 - MISC. COMMANDS.
182   ADC        PP (1100)         1137    0516  73/06/10. 71/01/09. 72/05/19. ANDY CAPP DISPLAY.
183   BAT        PP (1100)         1170    4076  73/06/10. 71/03/02. 73/05/08. BASEBALL GAME.
184   CHD        PP (1100)          751    3142  73/06/10. 71/01/09. 73/05/08. CHESS DISPLAY DRIVER.
185   CHESS      OVL 00,00        27076    2520   02/10/70.
186   DOG        PP (1100)          337    3774  73/06/11. 73/05/05.            SNOOPY WWI FLYING ACE.
187   9ZA        PP (3105)          750    4017  73/06/11. 73/05/05.            ACE - DISPLAY DATA.
188   DS1        PP (1100)          635    3375  73/06/11. 71/01/09. 73/05/08. 6612/DD60 DISPLAY ALIGNMENT TEST.
189   GYR        COS               5145    1712
190   WRM        PP (1100)          463    2443  73/06/11. 71/01/09. 72/05/19. WORM(S) DISPLAY.
191   (00)             SUM =      53543
```

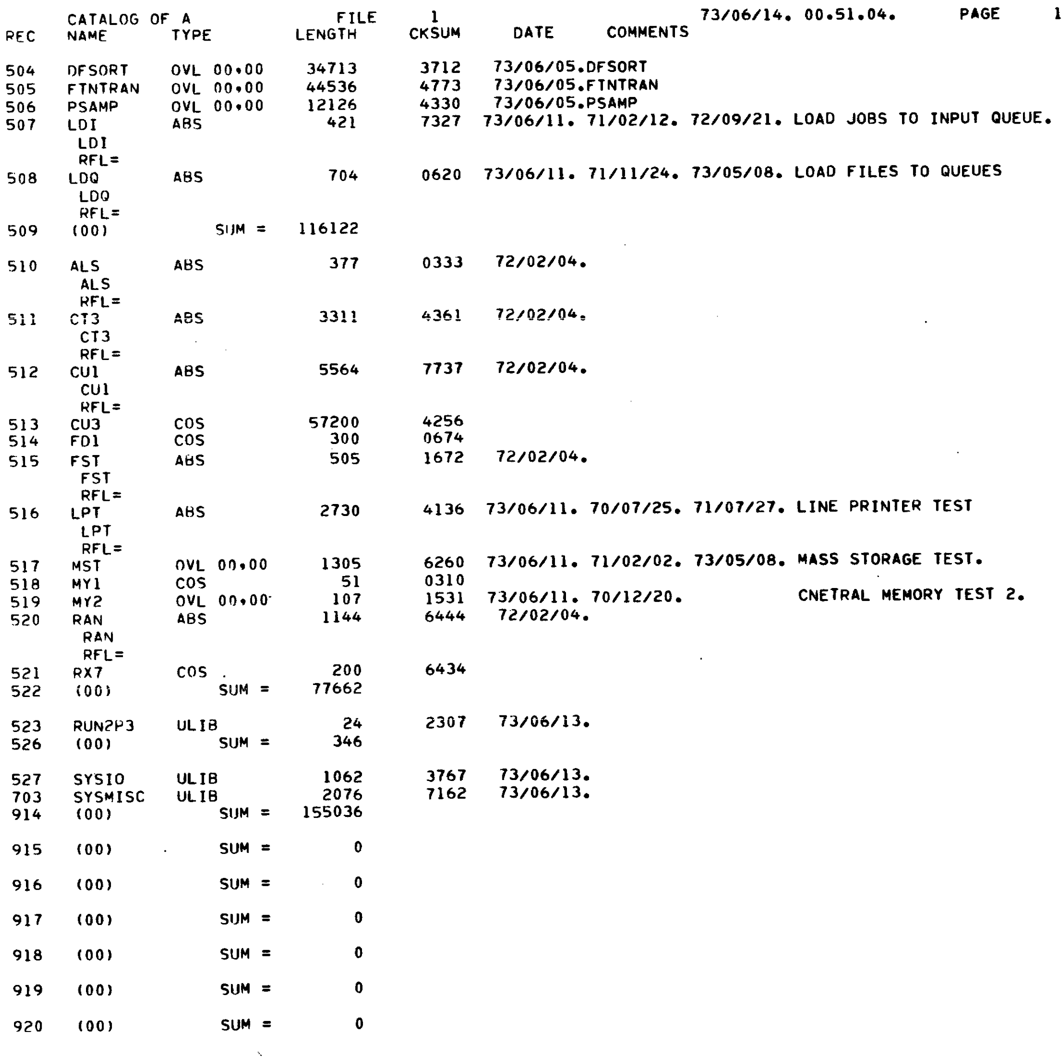
```
      CATALOG OF A                 FILE    1                      73/06/14. 00.51.04.      PAGE      7
REC   NAME      TYPE         LENGTH   CKSUM      DATE      COMMENTS

192   PFILES    ABS            1024    5526  73/06/11. 73/05/24.           PERMANENT FILE MANIPULATOR.
       APPEND
       ATTACH
       CHANGE
       DEFINE
       GET
       PACKNAM
       PERMIT
       PURGE
       REPLACE
       SAVE
       RFL=
       SDM=
193   PFATC1    OVL 01,00      2023    3161  73/06/11. 73/05/24.           PFATC1 - CATALOG PF ARCHIVE TAPE.
194   PFCAT1    OVL 01,00      3651    5126  73/06/11. 73/05/24.           PFCAT1 - CATALOG PERMANENT FILE DEVICE.
195   PFCOPY1   OVL 01,00      2066    2552  73/06/11. 73/05/24.           PFCOPY1 - COPY ARCHIVE FILE UTILITY.
196   PFDUMP1   OVL 01,00      4656    0404  73/06/11. 73/05/24.           PFDUMP1 - PERMANENT FILE DUMP.
197   PFLOAD1   OVL 01,00      6723    2604  73/06/11. 73/05/24.           PFLOAD1 - PERMANENT FILE LOAD.
198   PFS       ABS            2350    0545  73/06/11. 73/05/24.           PERMANENT FILE SUPERVISOR.
       PFS
       PFLOAD
       PFDUMP
       PFCAT
       PFATC
       PFCOPY
       RFL=
       SSJ=
199   PURGALL   ABS             620    6722  73/06/11. 73/05/05.           PURGALL - PURGE ALL PERMANENT FILES.
       PURGALL
       RFL=
200   CMS       PP (1100)       335    4601  73/06/10. 73/05/05.           MASS STORAGE MANAGER
201   IMS       PP (1100)       417    1637  73/06/11. 73/05/05.           INITIALIZE MASS STORAGE.
202   MSI       ABS            3140    7023  73/06/11. 73/05/24.           MASS STORAGE DEVICE INITIALIZATION.
       MSI
       FLAW
       RFL=
       SSJ=
203   PFM       PP (1100)       437    2040  73/06/11. 73/05/24.           PFM - PERMANENT FILE MANAGER.
204   3PA       PP (1372)       774    7572  73/06/11. 73/05/24.           PFM - COMMAND PROCESSING
205   3PB       PP (4705)       134    6471  73/06/11. 73/05/24.           PFM - SAVE,REPLACE,APPEND COMMAND PROCESSING.
206   3PC       PP (2046)        23    4046  73/06/11. 73/05/24.           PFM - APPEND PROCESSING.
207   3PD       PP (4705)       170    2356  73/06/11. 73/05/24.           PFM - ATTACH PROCESSING.
208   3PE       PP (1372)       345    5200  73/06/11. 73/05/24.           PFM - CATALOG LIST ROUINTES.
209   3PF       PP (4705)       167    7533  73/06/11. 73/05/24.           PFM - DEFINE PROCESSING.
210   3PG       PP (4705)        74    3777  73/06/11. 73/05/24.           PFM - PERMIT/PURGE PROCESSING.
211   3PH       PP (1372)       236    3042  73/06/11. 73/05/24.           PFM - ERROR PROCESSING.
212   3PI       PP (7000)        54    1467  73/06/11. 73/05/24.           PFM - AUXILIARY ROUTINES.
213   3PJ       PP (4705)        64    0531  73/06/11. 73/05/24.           PFM - CHANGE PROCESSING.
214   PFU       PP (1100)       736    3304  73/06/11. 73/05/24.           PF UTILITY PROCESSOR.
215   3UA       PP (3655)       357    7462  73/06/11. 73/05/24.           PFU - PFLOAD ROUTINES
216   1CK       PP (1100)       232    4734  73/06/11. 73/05/05.           SYSTEM CHECK POINT.
217   (00)         SUM =      41112

218   CHK       PP (1100)        20    3346  73/06/10. 70/09/03.           CHECK STATUS OF OUTPUT FILE.
219   CLL       PP (1100)       103    0063  73/06/10. 70/12/20. 73/05/08. *COS* FORMAT LIBRARY LOADER.
```

```
      CATALOG OF A              FILE     1                         73/06/14. 00.51.04.     PAGE      8
REC   NAME      TYPE        LENGTH     CKSUM     DATE     COMMENTS

220   CPM       PP (1100)      320      3126 73/06/10. 73/05/24.           CONTROL POINT MANAGER.
221   EXU       PP (1100)      107      4064 73/06/11. 70/12/20. 73/05/08. EXECUTE *COS* FORMAT PROGRAM.
222   LFM       PP (1100)      204      2270 73/06/11. 73/05/24.           LOCAL FILE MANAGER.
223   3LA       PP (1534)       44      6216 73/06/11. 73/05/24.           LFM - ERROR PROCESSOR.
224   3LB       PP (1534)       64      6423 73/06/11. 73/05/24.           LFM - LOCAL FILE FUNCTIONS.
225   3LC       PP (1534)      277      0040 73/06/11. 73/05/24.           LFM - EQUIPMENT REQUESTS.
226   3LD       PP (1534)      106      7103 73/06/11. 73/05/24.           LFM - COMMON FILE FUNCTIONS.
227   3LE       PP (1534)      107      6241 73/06/11. 73/05/24.           LFM - FILE DISPOSAL FUNCTIONS.
228   3LF       PP (1534)      117      6400 73/06/11. 73/05/24.           LFM - CONTROL STATEMENT FILE FUNCTIONS.
229   SFM       PP (1100)      636      4114 73/06/11. 73/05/24.           SYSTEM FILE MANAGER.
230   SFP       PP (1100)       47      2502 73/06/11. 72/09/25. SCOPE FUNCTION PROCESSOR.
231   2SA       PP (1141)      144      2676 73/06/11. 73/05/24.           72/09/25. SFP - STS - STATUS PROCESSOR
232   2SB       PP (1141)      737      7540 73/06/11. 73/05/24.           72/09/25. SFP - EMG - SDA/SIS MESSAGE GENERATOR
233   2SC       PP (1141)      206      6355 73/06/11. 73/05/24.           72/09/25. SFP - RPV - REPRIEVE CENTRAL PROGRAM
234   2SD       PP (1141)       36      1475 73/06/11. 73/05/24.           72/09/25. SFP - PFE - EXTEND/ALTER FUNCTION
235   2SE       PP (1141)      142      5213 73/06/11. 73/05/24.           72/09/25. SFP - ACE - ADVANCE CONTROL CARD
236   2SF       PP (1141)       25      0414 73/06/11. 73/05/24.           72/09/25. SFP - PRM - PERMISSION CHECKING FUNCTION
237   2SG       PP (1141)       15      1254 73/06/11. 73/05/24.           72/09/25. SFP - SRP - SPECIAL REQUEST PROCESSING
238   2SH       PP (1141)       77      0031 73/06/11. 73/05/24.           72/09/25. SFP - ERP - ERROR PROCESSOR
239   (00)          SUM =     4520

240   ACCFAM    ABS            435      2062 73/06/11. 73/05/24.           ACCOUNT AND FAMILY CARD PROCESSOR.       COPYRIGHT
       ACCOUNT
       FAMILY
       VAL=
       SDM=
       RFL=
       SSJ=
241   CHARGE    ABS            766      5275 73/06/11. 73/05/05.           JOB PROFILE VALIDATION PROGRAM.
       CHARGE
       ARG=
       SDM=
       VAL=
       RFL=
       SSJ=
242   ISF       ABS            625      4416 73/06/11. 73/05/05.           INITIALIZE SYSTEM FILES.
       ISF
       RFL=
       SSJ=
243   MODVAL    ABS          11335      5553 73/06/11. 73/05/24.           ACCOUNT FILE MANAGER.
       MODVAL
       LIMITS
       PASSWOR
       SDM=
       RFL=
       SSJ=
244   PROFILE   ABS           4364      3171 73/06/11. 73/05/24.           PROJECT PROFILE MANAGER.
       PROFILE
       ARG=
       RFL=
       SSJ=
245   SFS       OVL 01.00     1577      2447 73/06/11. 73/05/24.           SPECIAL SYSTEM FILE SUPERVISOR.
246   0AV       PP (0000)      142      0324 73/06/11. 73/05/05.           VERIFY USER ACCOUNT NUMBER.
247   2TJ       PP (2000)      122      5041 73/06/11. 70/12/13. 73/05/24. TRANSLATE JOB CARD.
248   (00)          SUM =    22254
```

```
      CATALOG OF A                  FILE      1                           73/06/14. 00.51.04.      PAGE       9
REC   NAME      TYPE             LENGTH    CKSUM     DATE       COMMENTS

249   1BA       PP (1100)            64     5361  73/06/11. 73/05/05.            BATCHIO AUXILLARY PROCESSOR.
250   3BA       PP (2000)            33     6407  73/06/11. 73/05/05.            1BA - ACCOUNTING.
251   3BB       PP (2000)           313     7501  73/06/11. 73/05/05.            1BA - INITIAL PRINT DATA.
252   3BC       PP (2000)           133     3214  73/06/11. 73/05/05.            1BA - INITIAL PUNCH DATA.
253   1CD       PP (1301)          1060     0642  73/06/11. 70/09/12. 73/05/24. BATCHIO COMBINED DRIVER.
254   1IO       PP (1100)           147     4231  73/06/11. 73/05/05.            BATCHIO MANAGER.
255   3IA       PP (2104)           175     6277  73/06/11. 73/05/05.            1IO - SUBROUTINES.
256   3IB       PP (3214)           234     1404  73/06/11. 72/12/28.            1IO - LOAD 3555/512 IMAGE MEMORY.
257   2LP       PP (2000)           253     2510  73/06/11. 70/12/13. 73/05/08. 3256/501 - 3555/512 LINE PRINTER DRIVER.
258   2PC       PP (2000)           267     0722  73/06/11. 70/12/13. 73/05/08. 3446/415 CARD PUNCH DRIVER.
259   2RC       PP (2000)           350     1055  73/06/11. 70/12/13. 73/05/08. 3447/405 CARD READER DRIVER.
260   (00)           SUM =         3557

261   E200CP    OVL 00,00           604     6233  73/06/11. 70/11/11. 73/05/08. EXPORT CPU SECTION.
262   XSP       PP (1100)           174     4677  73/06/11. 71/01/31. 73/05/08. EI-200 SERVICE PROCESSOR
263   1ED       PP (1100)          1004     4034  73/06/11. 70/10/09. 73/05/24. EI-200 6671 DRIVER
264   1LS       PP (1100)           120     7135  73/06/11. 73/05/24.            EI-200 EXPORT EXECUTIVE.
265   9IA       PP (2000)           555     6615  73/06/11. 73/05/24.            1LS - FUNCTION PROCESSOR
266   9IB       PP (6000)           112     7607  73/06/11. 73/05/24.            1LS - SEARCH OUTPUT QUEUE, INITIATE PRINT
267   9IC       PP (6000)           111     7142  73/06/11. 73/05/24.            1LS - PROCESS NEXT JOB
268   9ID       PP (2000)            25     6133  73/06/11. 73/05/24.            1LS - CENTRAL MEMORY MANAGER
269   9IE       PP (2000)            64     4367  73/06/11. 73/05/24.            1LS - INITIALIZE EXPORT
270   9IF       PP (2401)           121     5412  73/06/11. 73/05/24.            1LS - ABORT EXPORT
271   9IG       PP (1100)            35     1723  73/06/11. 73/05/24.            1LS - INITIALIZE LOCAL RPL
272   9IH       PP (7000)            77     1417  73/06/11. 73/05/24.            1LS - RESIDENT SUBROUTINES
273   (00)           SUM =         3512

274   (00)           SUM =            0

275   STIMULA   ABS               15666     7176  73/06/05. 73/05/05. 73/05/09. LOW SPEED INTERACTIVE TERMINAL STIMULATOR.
       STIMULA
       RFL=
276   TELEX     OVL 00,00          4061     0353  73/06/11. 71/03/02. 73/05/24. TTY EXECUTIVE INITILIZATION VERSION ≠VERSI
277   TELEX1    OVL 01,00          5651     3202  73/06/11. 71/03/02. 73/05/24. TTY EXECUTIVE PROCESSOR.
278   TELEX2    OVL 00,00           653     4005  73/06/11. 71/03/02. 73/05/24. TTY EXECUTIVE TERMINATION.
279   TLX       PP (1100)            34     3331  73/06/11. 73/05/05.            TERMINAL ACTION PROCESSOR.
280   1TA       PP (1100)           263     7002  73/06/11. 73/05/24.            TELEX AUXILIARY FUNCTION PROCESSOR.
281   3TA       PP (2571)            17     6677  73/06/11. 73/05/24.            1TA - ADJUST *TELEX* FIELD LENGTH.
282   3TB       PP (2571)            53     1563  73/06/11. 73/05/24.            1TA - RETURN TERMINAL JOB.
283   3TC       PP (2571)           135     6320  73/06/11. 73/05/24.            1TA - CREATE ROLLOUT FILE.
284   3TD       PP (2571)           173     7506  73/06/11. 73/05/24.            1TA - TERMINAL LOGOUT PROCESSOR.
285   3TE       PP (2571)            74     3114  73/06/11. 73/05/24.            1TA - TERMINAL RECOVERY PROCESSOR
286   3TF       PP (2571)           151     7453  73/06/11. 73/05/24.            1TA - SECONDARY FILE DESCRIPTIONS.
287   3TG       PP (2571)            36     7307  73/06/11. 73/05/24.            1TA - CPU TIME STATUS COMMAND.
288   3TH       PP (2571)           207     1130  73/06/11. 73/05/24.            1TA - RECOVERY FILE PROCESSOR.
289   3TI       PP (2571)           151     3737  73/06/11. 73/05/24.            1TA - JOB SCHEDULING.
290   3TJ       PP (2571)           107     0206  73/06/11. 73/05/24.            1TA - GATHER STATISTICS.
291   3TK       PP (2571)            63     2273  73/06/11. 73/05/24.            1TA - CLEAN UP SALVARE FILE.
292   3TL       PP (2571)            16     0314  73/06/11. 73/05/24.            1TA - INITIATE PRIMARY FILE.
293   3TM       PP (2571)            44     3745  73/06/11. 73/05/24.            1TA - CHECK JOB STATUS.
294   1TD       PP (4705)           476     3714  73/06/11. 73/05/11. 73/06/07. 1TD - INITIALIZATION AND COMPLETION.
295   2TD       PP (1100)          1105     3602  73/06/11. 73/05/11. 73/06/07. LOW SPEED INTERACTIVE TERMINAL DRIVER.
296   9JA       PP (0000)            70     0254  73/06/11. 73/05/11. 73/06/07. 1TD - *ASCII* TERMINAL.
297   9JB       PP (0000)            70     2131  73/06/11. 73/05/11. 73/06/07. 1TD - CORRESPONDENCE/TEXT.
298   9JC       PP (0000)            70     1775  73/06/11. 73/05/11. 73/06/07. 1TD - CORRESPONDENCE/APL.
299   9JD       PP (0000)            70     0735  73/06/11. 73/05/11. 73/06/07. 1TD - MEMOREX 1240/APL.
```

```
      CATALOG OF A            FILE      1                          73/06/14. 00.51.04.      PAGE     10
REC   NAME       TYPE        LENGTH    CKSUM     DATE      COMMENTS

300   1TO        PP (1100)      310     3557  73/06/11. 71/01/31. 73/05/24. TERMINAL INPUT/OUTPUT.
301   1TS        PP (1100)      311     2114  73/06/05. 73/05/05.           INTERACTIVE TERMINAL STIMULATION DRIVER.
302   (00)          SUM =     35557

303   BLANK      ABS           1327     3315  73/06/11. 73/05/24.           BLANK TAPE LABELING PROGRAM.   COPYRIGHT CONTROL D
       BLANK
       ARG=
       RFL=
       SSJ=
304   MAGNET     OVL 00,00      755     2047  73/06/11. 73/05/24.           MAGNETIC TAPE EXECUTIVE.
305   MAGNET1    OVL 00,00      275     6033  73/06/11. 73/05/24.           MAGNETIC TAPE EXECUTIVE TERMINATION.
306   RESEX      ABS           5016     0355  73/06/11. 73/06/01.           RESOURCE EXECUTIVE.  COPYRIGHT CONTROL DATA CORP.
       ASSIGN
       LABEL
       REQUEST
       RESOURC
       VSN
       LFM
       PFM
       REQ
       ARG=
       DMP=
       RFL=
       SSJ=
307   1LT        PP (1100)        2     4615  73/06/11. 73/05/24.           1LT - LONG RECORD PROCCESOR.
308   1MT        PP (1100)      216     5726  73/06/11. 73/05/24.           PPU MAGNETIC TAPE EXECUTIVE.
309   3MA        PP (2377)      316     1455  73/06/11. 73/05/24.           1MT - INITIALIZE TAPE EXECUTIVE.
310   3MB        PP (6603)      170     6120  73/06/11. 73/05/24.           1MT - ERRLOG MESSAGE PROCESSOR.
311   3MC        PP (2377)      405     5150  73/06/11. 73/05/24.           1MT - LOAD CONVERSION MEMORY.
312   3MD        PP (6603)       72     6211  73/06/11. 73/05/24.           1MT - CONTROL POINT/CODED PRESET.
313   3ME        PP (2161)      250     1372  73/06/11. 73/05/24.           1MT - USER JOB OPERATIONS.
314   3MF        PP (2161)      216     4100  73/06/11. 73/05/24.           1MT - READ FUNCTION PROCESSOR.
315   3MG        PP (2660)        3     1420  73/06/11. 73/05/24.           1MT - READ LONG BLOCK PROCESSOR.
316   3MH        PP (7301)       66     0173  73/06/11. 73/05/24.           1MT - READ - CODED PROCESSOR. 73/05/24.
317   3MI        PP (3047)      465     7120  73/06/11. 73/05/24.           1MT - READ LABEL PROCESSOR.
318   3MJ        PP (2411)      105     3042  73/06/11. 73/05/24.           1MT - TAPE POSITIONING OPERATIONS.
319   3MK        PP (3777)       65     5766  73/06/11. 73/05/24.           1MT - READ ERROR PROCESSOR.
320   3ML        PP (2161)      225     5360  73/06/11. 73/05/24.           1MT - WRITE FUNCTION PROCESSOR.
321   3MM        PP (2656)        3     3433  73/06/11. 73/05/24.           1MT - WRITE LONG BLOCK PROCESSOR.
322   3MN        PP (7301)       75     7120  73/06/11. 73/05/24.           1MT - WRITE - CODED PROCESSOR.73/05/24.
323   3MO        PP (3052)      227     1160  73/06/11. 73/05/24.           1MT - WRITE LABEL PROCESSOR.
324   3MP        PP (4002)       64     2743  73/06/11. 73/05/24.           1MT - WRITE ERROR PROCESSOR.
325   (00)          SUM =     13542

326   CATALOG    ABS           1344     3232  73/06/11. 71/02/12. 73/05/24. CATALOG FILE.
       CATALOG
       RFL=
327   CATLIST    ABS           1744     0007  73/06/11. 71/03/02. 73/05/08. CATALOG PERMANENT FILES.
       CATLIST
       RFL=
328   CONTROL    ABS            214     5402  73/06/11. 73/05/24.           JOB CONTROL PROCESSOR.
       COMMENT
       MODE
       NOEXIT
       ONEXIT
       ONSW
       CATALOG OF A              FILE    1                          73/06/14. 00.51.04.      PAGE     11
REC    NAME      TYPE           LENGTH     CKSUM     DATE      COMMENTS
        OFFSW
        RFL
        ROLLOUT
        SETPR
        SETTL
        SUI
        SWITCH
        USECPU
        RFL=
329    CTL2      ABS             1064      4777  73/06/11. 73/05/24.           CONTROL STATEMENT BRANCH PROCESSOR.
        CALL
        EXIT
        GOTO
        RFL=
330    CTL3      ABS              763      2465  73/06/11. 73/05/24.           CONTROL REGISTER AND CONDITIONAL TESTING.
        DISPLAY
        IF
        SET
        RFL=
        ARG=
331    COPYB     ABS             1173      3017  73/06/11. 73/05/24.           BINARY FILE COPIES.
        COPY
        COPYBF
        COPYBR
        COPYEI
        COPYX
        RFL=
332    COPYC     ABS              602      5257  73/06/11. 71/03/02.           CODED FILE COPIES.
        COPYSBF
        COPYCF
        COPYCR
        RFL=
333    COPY67    ABS              410      4302  73/06/11. 70/06/06. 71/02/14. COPY 6600 TAPES TO 7600 FORMAT.
        COPY67
        RFL=
334    COPY76    ABS              400      7661  73/06/11. 70/06/06. 71/02/14. COPY 7600 TAPES TO 6600 FORMAT.
        COPY76
        RFL=
335    CPMEM     ABS             4630      2317  73/06/11. 72/12/10.           CONTROL POINT MEMORY UTILITIES.
        DMD
        DMP
        LBC
        LOC
        PBC
        RBR
        WBR
        DMP=
        MFL=
336    DAYFILE   ABS              556      5755  73/06/11. 71/03/01. 73/05/07. DUMP DAYFILES.
        DFD
        AFD
        ELD
        DAYFILE
        RFL=
337    DOCMENT   OVL 00,00       3301      6452  73/06/11. 70/08/25. 71/12/05. INTERNAL/EXTERNAL DOCUMENTATION PROGRAM.
```

```
      CATALOG OF A              FILE     1                         73/06/14. 00.51.04.      PAGE      12
REC   NAME      TYPE        LENGTH     CKSUM    DATE     COMMENTS

338   EDIT      ABS           3213      4010  73/06/11. 73/05/05.           TIME SHARING TEXT EDITOR
       EDIT
       RFL=
339   FILES     ABS            574      0557  73/06/11. 73/05/24.           LOCAL FILE MANIPULATOR.
       BKSP
       COMMON
       DISPOSE
       EVICT
       LOCK
       OUT
       RELEASE
       RENAME
       REWIND
       RETURN
       SKIPEI
       SKIPF
       SKIPFB
       SKIPR
       SETID
       UNLOAD
       UNLOCK
       WRITEF
       WRITER
       RFL=
340   HELP      ABS           2337      5235  73/06/11. 71/03/02. 73/05/23. PROCESS HELP FOR TS USER.
       HELP
       RFL=
341   GTR       ABS           1513      1216  73/06/11.
       GTR
       COPYRF
342   LIBEDIT   OVL 00,00     4117      1603  73/06/11. 70/06/06. 73/05/24. LIBRARY EDITING PROGRAM.
343   LIST80    ABS           1222      5553  73/06/11. 01/20/70. 71/02/14. COMPRESS COMPASS LISTINGS.
       LIST80
       RFL=
344   L072      ABS           3140      2317  73/06/11. 70/08/01.           COMPRESS OUTPUT FILES.
       L072
       RFL=
345   MSORT     OVL 00,00      255      5502  73/06/11. 71/03/01. 71/12/05. MULTI-TERMINAL SORT ROUTINE.
346   PACK      ABS            367      3611  73/06/11. 71/01/06. 73/05/23. PACK - PACK FILE TO ONE RECORD.
       PACK
       RFL=
347   RESEQ     ABS           1405      6614  73/06/11. 71/02/28. 73/05/08. TIME SHARING RESEQUENCE ROUTINE.
       RESEQ
       RFL=
348   SORT      OVL 00,00      720      5525  73/06/11. 71/03/01. 72/03/06. FILE SORT ROUTINE
349   STAGE     ABS            770      5511  73/06/11. 71/01/06. 73/05/07. STAGE TAPE FILES TO DISK.
       STAGE
       RFL=
350   SUBMIT    ABS           1436      0367  73/06/11. 71/02/12. 73/05/08. ENTER JOB IN INPUT QUEUE.
       SUBMIT
       RFL=
351   TDUMP     OVL 00,00     1057      0160  73/06/11. 73/05/05.           TDUMP - FILE DUMP.
352   VALNET    OVL 00,00     5753      7721  73/06/11. 72/06/14.           VALIDATE TERMINAL DESCRIPTION FILE.
353   VERIFY    ABS           1452      0207  73/06/11. 73/05/05.           VERIFY FILES.
       VERIFY
       RFL=
       CATALOG OF A             FILE     1                      73/06/14. 00.51.04.      PAGE      13
REC    NAME        TYPE           LENGTH      CKSUM     DATE       COMMENTS

354    VFYLIB      OVL 00,00       1545        5722  73/06/11. 73/05/24.            69/05/07. VERIFY LIBRARY FILES.
355    OUT         PP (1100)         27        4610  73/06/11. 70/09/12. 72/05/19. RELEASE OUTPUT FILES.
356    SMP         PP (1100)        156        1277  73/06/11. 71/07/27. 73/05/08. SAMPLE CPU P-REGISTER.
357    (00)             SUM =     62566

358    COMPASS     OVL 00,00        664        6310  73/05/10. CYBER 70/ MODEL 76  COMPREHENSIVE ASSEMBLER PROGRAM VERSION 3.73130.
359    COMP2$      OVL 01,00      22326        4215  73/05/10. CYBER 70/ MODEL 76  COMPREHENSIVE ASSEMBLER PROGRAM VERSION 3.73130.
360    KRONREF     OVL 00,00       1646        5646      5332  73/06/11. 70/10/26. 72/09/21. KRONOS SYSTEM CROSS REFERENCE PROGRAM.
361    OPLEDIT     OVL 00,00       3017        2063  73/06/11. 07/20/69. 73/05/07. OPL EDITING PROGRAM.
362    MODIFY      OVL 00,00       6475        6423  73/06/11. 70/06/06. 73/05/08. SOURCE LIBRARY EDITOR - VER 2.0.


363    UPDATE      OVL 00,00      13445        0423  73/05/10.
364    UPMOD       OVL 00,00       1720        5601  73/06/11. 70/06/06. 73/05/07. UPDATE TO MODIFY CONVERSION PROGRAM.
365    (00)             SUM =     54161

366    SYSTEXT     OVL 01,01       5127        3733  73/06/10. 70/12/13.            SYSTEM COMMUNICATION TEXT.
367    (00)             SUM =      5127

368    CPCTEXT     OVL 01,01       4423        4224  73/05/05. SYSTEM TEXT FOR CPU PROGRAMS USING *CPC*.
369    CPUTEXT     OVL 01,01       3714        3465  73/05/10. SYSTEM TEXT FOR CPU PROGRAMS USING MACE I/O.
370    IOTEXT      OVL 01,01       7235        3653  73/06/08.
371    IPTEXT      OVL 01,01        335        5605  73/05/05. INSTALLATION PARAMETER SYSTEM MACROS.
372    LDRTEXT     OVL 01,01        676        5557  73/05/05. LOADER REQUEST SYSTEM MACROS.
373    PFMTEXT     OVL 01,01       1060        1200  73/05/05. PERMANENT FILE SYSTEM MACROS.
374    PPTEXT      OVL 01,01       5130        6341  73/05/05. PPU PROGRAM SYSTEM TEXT.
375    SCPTEXT     OVL 01,01      10124        5623  73/05/05. SCOPE 3 SYSTEM TEXT.
376    TXT6RM      OVL 01,01      12047        7533  73/06/08.
377    (00)             SUM =     51432

378    FILE        OVL 00,00       1557        1050   73/06/08.FILE
379    LINK        OVL 01,00        210        5252  73/06/11. 73/05/24.            LINK - CONTROL CARD LOAD.
380    LDR=        OVL 00,00       3060        1730  73/06/11. 73/05/24.            CPU RELOCATABLE PROGRAM LOADER.
381    LINKCTL     ABS              674        0732  73/06/11. 73/05/24.            LINK - CONTROL CARD PROCESSING.
         LDSET
         LOAD
         MAP
         NOMAP
         LIBRARY
         REDUCE
         SETCORE
382    MAP=        OVL 01,00       1137        2356  73/06/11. 73/05/24.            LINK - WRITE LOAD MAP.
383    OVG=        OVL 01,00       2473        5766  73/06/11. 73/05/24.            LINK - OVERLAY GENERATION.
384    LIBGEN      OVL 01,00       3067        3561  73/06/11. 73/05/24.            LINK - GENERATE USER LIBRARY.
385    SYSLIB      ULIB            1420        1124   73/06/11.
495    (00)             SUM =     36250

496    LIBMOD      TEXT              17        6625
497    GENFILS     TEXT              25        5267
498    GENINSD     TEXT              17        0475
499    GENSYS      TEXT             106        6625
500    GENVAL      TEXT              11        3262
501    GENHELP     TEXT              37        5635
502    (00)             SUM =       241

503    DMQ         ABS              776        3253  73/06/11. 71/11/24. 73/05/08. DUMP QUEUES
         DMQ
         RFL=
```



```
        CATALOG OF A            FILE       1                            73/06/14. 00.51.04.       PAGE      14
REC     NAME       TYPE         LENGTH     CKSUM      DATE      COMMENTS

504     DFSORT     OVL 00,00     34713     3712    73/06/05.DFSORT
505     FTNTRAN    OVL 00,00     44536     4773    73/06/05.FTNTRAN
506     PSAMP      OVL 00,00     12126     4330    73/06/05.PSAMP
507     LDI        ABS             421     7327   73/06/11. 71/02/12. 72/09/21. LOAD JOBS TO INPUT QUEUE.
          LDI
          RFL=
508     LDQ        ABS             704     0620   73/06/11. 71/11/24. 73/05/08. LOAD FILES TO QUEUES
          LDQ
          RFL=
509     (00)           SUM =    116122

510     ALS        ABS             377     0333    72/02/04.
          ALS
          RFL=
511     CT3        ABS            3311     4361    72/02/04.
          CT3
          RFL=
512     CU1        ABS            5564     7737    72/02/04.
          CU1
          RFL=
513     CU3        COS           57200     4256
514     FD1        COS             300     0674
515     FST        ABS             505     1672    72/02/04.
          FST
          RFL=
516     LPT        ABS            2730     4136   73/06/11. 70/07/25. 71/07/27. LINE PRINTER TEST
          LPT
          RFL=
517     MST        OVL 00,00      1305     6260   73/06/11. 71/02/02. 73/05/08. MASS STORAGE TEST.
518     MY1        COS              51     0310
519     MY2        OVL 00,00       107     1531   73/06/11. 70/12/20.           CNETRAL MEMORY TEST 2.
520     RAN        ABS            1144     6444    72/02/04.
          RAN
          RFL=
521     RX7        COS .           200     6434
522     (00)           SUM =     77662

523     RUN2P3     ULIB             24     2307    73/06/13.
526     (00)           SUM =       346

527     SYSIO      ULIB           1062     3767    73/06/13.
703     SYSMISC    ULIB           2076     7162    73/06/13.
914     (00)           SUM =    155036

915     (00)           SUM =         0

916     (00)           SUM =         0

917     (00)           SUM =         0

918     (00)           SUM =         0

919     (00)           SUM =         0

920     (00)           SUM =         0
```

```
      CATALOG OF A            FILE    1                73/06/14. 00.51.04.        PAGE     15
REC   NAME     TYPE       LENGTH    CKSUM    DATE     COMMENTS

921   (00)        SUM =        0

922   (00)        SUM =        0

923   (00)        SUM =        0

924   (00)        SUM =        0

925   (00)        SUM =        0

926   SYSTEM   OPLD         1433    4762  73/06/14.

927   * EOF *     SUM =  1140247
```

1.2 COMPASS 3.0

1.2.1 RELEASE DESCRIPTION

COMPASS 3.0 consists of two overlays. The level (0,0) overlay, COMPASS, is the main control program. The level (1,0) overlay, COMP2$, contains the assembler which can be called by compilers to process embedded COMPASS source programs.

HARDWARE REQUIREMENTS

COMPASS 3.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length requirement to run COMPASS 3.0 is 42,000$_8$.

CORRECTIONS

COMPASS 3.0 includes all eligible PSR corrective code published through PSR Summary 348.

DEFICIENCIES

Modification CPS020 from PSR Summary 351 must be available in COMPASS 3.0 if it is to operate under KRONOS 2.1. This modification is included in the binary code on KPL5 and on the deadstart tape, but it is not included in the source code on PL2. Therefore, add this modification the first time that the COMPASS 3.0 source code is modified and/or updated.

```
*/ $$CPS020         9    COMPASS
*IDENT    CPS020      RHG.    73/03/20.
*/        PSR                 COMPASS 3.0  SCOPE 3.4 KRONOS 2.1.
*/        PREVENT *I/O SEQUENCE ERROR* ON SCRATCH FILE.
*/
*DELETE COMPASS.4822
 PRT3     RECALL R
          SX6    CBUF         WRITE COMPASS TO DISK
*COMPILE COMPASS
*/    THERE ARE    9 CORRECTION CARDS INCLUDING THIS COMMENT.
```

1.2.2 INSTALLATION PROCEDURE

The deadstart tape contains the KRONOS 2.1 version of COMPASS 3.0. The procedure to modify and update COMPASS 3.0 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for COMPASS 3.0, the following materials are necessary.

KPL2 contains the jobs to modify and update; the jobs required for COMPASS 3.0 are CMP1 and CMP2.

CMP1 — Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.

CMP2 — Use to collect the COMPASS binary code and generate directive files for later use as input to the procedure file GENSYS.

KPL5 contains COMPASS 3.0 binary code.

PL2 contains COMPASS 3.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.1.2.

1.2.3 INSTALLATION PARAMETERS

The installation parameter definitions are in alphabetic order on pages 5 through 8 of the COMPASS 3.0 assembly listing. The following are parameters that an installation may change.

Program	Released Default Value	Significance

INTMUL	0	Specifies the integer multiply instruction usage
		0 Integer multiply OPDEF feature
		1 Integer multiply hardware feature
		To change the released value, use the following modification.
		`*D CMP30.114`
		`INTMUL EQU 1`

MODEL	74	Specifies the CDC CYBER 70 Series model on which COMPASS runs. To run COMPASS 3.0 under KRONOS 2.1, change this parameter to 76.
		To change the released value, use the following modification.
		`*D CMP30.152`
		`MODEL MICRO 1,,*76*`
		Make this change when using the CMP1 job to generate a new program library from PL2.

TIMEMSG	1	Indicates whether or not COMPASS 3.0 issues a dayfile message giving the total CPU time at the end of a batch of assemblies. 0 COMPASS does not issue message 1 COMPASS issues message

1.2.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

CMP1

```
CMP1
CMP1,CM51000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF COMPASS 3.0 ON KPL5.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=COMPASS3P0*3P4,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
RETURN,OLDPL.
COMPASS,I,S=IPTEXT,B=COMTEXT,L=0.
COMPASS,I,S=IPTEXT,G=COMTEXT,B=NEWPL,L=0.
WRITEF,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL5,W,FI=COMPASS3P0*3P4,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL5,V.
*WEOR
*/          PLACE *COMPASS* MODSET HERE.
*WEOF
```

CMP2

```
CMP2
CMP2,CM15000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL5,R,FI=COMPASS3P0*3P4,MT,D=HY,F=SI)
SKIPF,KPL5.
COPYBF,KPL5,COMPASS.
UNLOAD,KPL5.
RETURN,KPL5.
COMMON,ZZZZLDF,ZZZPSBF.
CATALOG,COMPASS,R.
COPYBF,COMPASS,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
*WEOR
*ADD        LIB13,OVL/COMPASS-COMP2$
*WEOF
```

1.3 RECORD MANAGER 1.0

1.3.1 RELEASE DESCRIPTION

Record Manager 1.0 modules are in two parts, basic access modules and sequential and word addressable I/O modules. Basic access modules control selective loading based upon file organization. They contain jump vectors directing a user call to the I/O code appropriate to the file organization selected. Their program names have an RM suffix. The suffix for the sequential and word addressable I/O modules is the file type, such as SQ, WA, or DA.

The I/O macro text included with the Record Manager program library is IOTEXT. It consists of, but is not limited to, the macros included in the following table. (Some auxiliary macros exist that are not supported at the user level.)

Macro Name	System	Reference	COMDECK
FILE	6RM	Record Manager Ref. Man.	6RMCOM
FETCH	↓		
STORE		↓	↓
OPENM			
CLOSEM			
GET			
GETP			
GETN	IS/DA		
PUT	6RM		
PUTP	↓		
REPLACE			
DELETE			
ENDFILE			
SKIPdu			
d=F/B, u=L/P/F			
SEEK	IS/DA		
REWINDM	6RM		
WEOR	↓		
WTMK			
GETL			
PUTL			
CLOSEL			
CHECK			

HARDWARE REQUIREMENTS

Record Manager 1.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The Record Manager field length will vary from 5000_8 to $16,000_8$ if it is used.

CORRECTIONS

Record Manager 1.0 includes all eligible PSR corrective code published through PSR Summary 348.

DEFICIENCIES

EO = TD/DD/AD, does not work.

Checksumming of I type blocks is not available.

Calls to the 7000 Record Manager are not available.

Modification RM40089 from PSR Summary 351 must be available in Record Manager 1.0 if it is to operate under KRONOS 2.1. This modification is included in the binary code on KPL6 and on the deadstart tape, but it is not in the source code on PL3. Therefore, add this modification the first time the source code is modified and/or updated.

```
*IDENT RM40089
*/   THIS CODE IS FOR SIS 2.0 UNDER SCOPE 3.4
*/ *******************
*/
*/ KRONOS MODS TO ENABLE SIS VER 2.0 TO RUN UNDER KRONOS 2.1
*/
*/ *******************
*D,6RMRMIO.201
           SX1    X1+4          LIMIT = LWA + 4
*D,6RMRMIO.204
           SX7    X7-4          LWA
*D,6RMRMIO.228
           SX7    X7+3          END OF DATA + 4
*I,6RMRMIO.257
           ZR     X7,*+1
           SX7    X7-2
*/ TWJ - 03/22/73
*/ FIX KRONOS 2.1 3 WORD LOOK AHEAD
*/ APPLIES TO 3.4 6RM
*INSERT C6RMMGRM.33
 ≡FG≡      EQU    4
*DELETE C6RMMGRM.230
           SX4    X4+≡FG≡
*DELETE C6RMMGRM.250
           SX4    X4+≡FG≡
*DELETE C6RMMGRM.276
           SX4    X4+≡FG≡
*DELETE C6RMMGRM.296
           SX4    X4+≡FG≡
*DELETE    IMS0128.334,IMS0128.335
*0   2.    FIND AMOUNT OF SPACE REQUESTED (NSP) - ADD 4 TO IT. THE USERS
*          REQUEST IS INCREASED BY FOUR WORDS TO AVOID A BUFFER ARGUMENT
*DELETE IMS0128.337
*          AINS NSP+4. SAVE NEXT AVAILABLE ID (NID) IN B2. NEXT STEP
*DELETE C6RMMGRM.410
           SB4    X4+≡FG≡
*DELETE RM9704.10
           SX6    B4-≡FG≡
*C 6RMMGRM
*COMPILE 6RMRMIO,C6RMMGRM
```

1.3.2 INSTALLATION PROCEDURE

The deadstart tape contains the released version of Record Manager 1.0. The procedure to modify and update Record Manager is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for Record Manager, the following materials are necessary.

KPL2 contains the jobs to modify and update, and verify installations; the jobs required for Record Manager 1.0 are SIXRM1, SIXRM2, and SIXRM3.

SIXRM1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
SIXRM2	Use to collect the Record Manager binary code and generate directive files for later use as input to the procedure file GENSYS.
SIXRM3	Use to verify that the product is correctly installed.

KPL4 contains SCOPE 3.4 texts' binary and source code.

KPL6 contains Record Manager 1.0 binary code.

PL3 contains Record Manager 1.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.1.3.

1.3.3 INSTALLATION PARAMETERS

The installation parameters described below permit a certain amount of tailoring. To facilitate writing the Update cards, 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.

```
*DELETE           <mnemonic>.1

=<parameter>=     EQU  <user-value>
```

Use the following Update IDENTS 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.

Record Manager routines that reference SCOPE 3.4 system text IPARAMS are listed in part III, section 1.7.3.

1.3.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

SIXRM1

```
SIXRM1
SIXRM1,CM60000,T2000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB UPDATES AND CREATES THE BINARY OF 6RM
COMMENT.  AND COPIES THE OLDPL AND BINARY OF IS AND DA TO THE
COMMENT.  NEWPL.  THE NEWPL WILL CONSIST OF NINE FILES
COMMENT.  THE FIRST FILE WILL BE THE 6RM NEWPL
COMMENT.  THE SECOND FILE WILL BE THE BINARIES OF THE TEXTS
COMMENT.  THE THIRD FILE WILL BE THE BINARY OF THE 6RM I/O MODULES
COMMENT.  THE FOURTH FILF WILL BE THE ABS. BINARY OF THE FILE PROGRAM
COMMENT.  THE FIFTH WILL BE THE RELOCATABLE BINARY OF THE FILE PROGRAM
COMMENT.  THE LAST  FOUR FILES WILL BE THE PLS AND BINARIES
COMMENT.  OF IS AND DA.
*VSN CARD.
LABEL(OLDPL,R,FI=SCOPE3P4TEXTS*2P1,MT,D=HY,F=SI)
REWIND,OLDPL.
UPDATE(Q,P=OLDPL,N=RANPL,C=0)
UNLOAD(OLDPL)
RETURN(OLDPL)
*VSN CARD.
LABEL(OLDPL,R,FI=RM*3P4,F=SI,MT,D=HY)
REWIND,OLDPL.
UPDATE(F,P=OLDPL,N=NEWPL,W,X,R=C)
SKIPF,NEWPL.
REWIND,OLDPL.
COMPASS,I,L=0,B=TXTS,S=IPTEXT,X=RANPL.
GTR(TXTS,TEXTS)OVL/IOTEXT-TXT6RM
COMPASS,I,L=0,G=TEXTS/TXT6RM,S=PFMTEXT,B=IOMODS.
COMPASS,I,L=0,G=TEXTS/TXT6RM,B=FILEC.
RETURN,COMPILE.
LDSET(LIB=SYSIO,REWIND)
LOAD(FILEC)
REWIND,IOMODS,FILE,FILEC,TEXTS.
COPYBF,TEXTS,NEWPL.
COPYBF,IOMODS,NEWPL.
COPYBF,FILE,NEWPL.
COPYBF,FILEC,NEWPL.
SKIPF,OLDPL,5.
COPYBF,OLDPL,NEWPL,4.
UNLOAD,OLDPL.
RETURN,OLDPL.
REWIND,NEWPL.
*VSN CARD.
LABEL(KPL6,W,FI=RM*3P4,D=HY,MT,F=SI)
REWIND,KPL6.
COPYEI,NEWPL,KPL6,V.
CATALOG,KPL6,R,N=0.
UNLOAD,KPL6.
*WEOR
*C CPCTEXT
*WEOR
*/  PLACE 6RM MODIFICATIONS HERE
*WEOF
```

SIXRM2

```
SIXRM2
SIXRM2,CM45000,T2000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB ADDS THE 6RM V1.0 BINARIES TO THE SYSTEM COMMON FILE
COMMENT.  ZZZPSBF FROM THE KPL6 RELEASE TAPE.  THE FILE PROGRAM AND
COMMENT.  THE RECORD MANAGER TEXTS ARE ADDED AS OVERLAYS.  THE 6RM
COMMENT.  I/O MODULES ARE ADDED TO THE USER LIBRARY -SYSIO-.
COMMENT.  EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT.  JOB SIXRM1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT.  TO COMMON FILE ZZZZLDF FOR GENERARING A NEW SYSTEM DEADSTART.
COMMENT.
*VSN CARD.
LABEL(KPL6,R,FI=RM*3P4,MT,D=HY,F=SI)
SKIPF,KPL6.
COPYBF,KPL6,RM.         IOTEXT,TXT6RM
COPYBF,KPL6,IOMODS.      6RM I/O MODULES
BKSP,RM.
COPYBF,KPL6,RM.          FILE PROGRAM
UNLOAD,KPL6.
RETURN,KPL6.
CATALOG,RM,R.
COMMON,ZZZPSBF,ZZZZLDF.
COPYBF,RM,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSIO
IF(EF=0)GOTO,10.
SET(R3=1)
COMMON,SYSTEM.
GTR(SYSTEM,OLD)ULIB/SYSIO
10,LIBEDIT,N=LGO,B=IOMODS,L=0.
LIBGEN,F,P=SYSIO.
CATALOG,SYSIO,U,R.
RETURN,NEW.
LIBEDIT,P=ZZZPSBF,B=SYSIO,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
*WEOR
*BEFORE *,REL/*
*WEOR
*BEFORE *,ULIB/SYSIO
*WEOR
*ADD      LIB15,OVL/IOTEXT,TXT6RM
*ADD      LIB16,OVL/FILE
*WEOR
*ADD      LIB21,ULIB/SYSIO
*WEOF
```

SIXRM3

```
SIXRM3
SIXRM3,CM60000,T30.
*ACCOUNT CARD.
COMMENT. *****************
COMMENT.
COMMENT.  6RM INSTALLATION VERIFICATION TEST
COMMENT.
COMMENT. **********************
LIBRARY(SYSIO)
COMPASS(S=IOTEXT)
LGO.
COMMENT.  ****** 6RM TEST PASSED ******
EXIT.
COMMENT.  ****** 6RM TEST FAILED ******
*WEOR
          IDENT    P3BIN
          ENTRY   P3
          SST
*
*         BINARY FILE ******
*         1 RECORD,  512 WORDS,  WORDS ARE NUMBERED IN ASCENDING ORDER
*
          LIST     G
FIT       FILE     LFN=TAPE,BFS=513,FWB=BFS,MRL=512,FET=FET,WSA=WSA,CM=N
,O,BT=I,RT=W,MNR=10,MBL=5120,LT=UL
BFS       BSSZ     517
FET       BSSZ     13
WSA       BSSZ     514
RL        BSSZ     1
P3        BSS      0
          OPENM    TAPE,I-O
          SB1      1                      INCREMENT
          SB2      513
          SB3      B0                     COUNTER
          SB4      WSA
          SB4      B4-B1
DO        SB3      B3+B1
          SX7      B3                     SET WORD TO NUMBER IN SEQUENCE
          SA7      B4+B3                  STORE IN BUFFER
          LE       B3,B2,DO
          STORE    TAPE,MRL=5120          MAXIMUM LENGTH OF RECORD IN CHA
PPPP      PUT      TAPE,WSA,5120,,,       WRITE 5120 CHARACTERS
          SX7      B0
          SA7      WSA
          SA7      WSA+511                LAST WORD
          REWINDM  TAPE
GGGG      GET      TAPE,WSA,RL,,,
*                                         NOW, CHECK THE FIRST AND LAST W
*                                         WORD OF THE FILE
```

```
          SB2       512
          SB1       1
          SA1       WSA
          SB5       X1
          NE        B5,B1,ERROR           IS THE FIRST WORD = 1
          SA1       WSA+511                LAST WORD
          SB5       X1
          NE        B5,B2,ERROR            IS LAST WORD = 512
          CLOSEM    TAPE,R,FILE
          ENDRUN
ERROR     BSS     0
SS     0
          MESSAGE   MSG0,,R
          ABORT     01
MSG0      DIS       ,*DATA NOT WRITTEN CORRECTLY - TEST FAILED*
          END       P3
*WEOR
```

Dayfile information from running the installation verification program should appear similar to the following.

```
01.20.12.SIXRM3,CM60000,T30.
01.20.12.*ACCOUNT CARD.
01.20.12.ACCOUNT,USER1.
01.20.13.COMMENT. *****************
01.20.13.COMMENT.
01.20.13.COMMENT.  6RM INSTALLATION VERIFICATION
01.20.13.TEST
01.20.13.COMMENT.
01.20.13.COMMENT. **********************
01.20.14.LIBRARY(SYSIO)
01.20.14.COMPASS(S=IOTEXT)
01.20.23.    3.989  CPU SECONDS  ASSEMBLY TIME.
01.20.23. ASSEMBLY COMPLETE.   50200B  CM USED.
01.20.23.LGO.
01.20.24.COMMENT.  ****** 6RM TEST PASSED ******
```

1.4 DIRECT ACCESS 1.0

1.4.1 RELEASE DESCRIPTION

Direct Access (DA) operates as a functional module of Record Manager 1.0. It provides all the routines necessary to create, update, and access random files on mass storage. It consists of a related set of central processor routines that are loaded, as required, in the user's field length.

A key analysis utility routine is available to aid in the selection of a hashing routine.

A direct access file created by DA can only be processed by DA. (Do not confuse a file created by DA 1.0 with the KRONOS direct access permanent file concept.)

HARDWARE REQUIREMENTS

DA 1.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. Refer to the field length requirements for Record Manager in part III, section 1.3.1.

CORRECTIONS

DA 1.0 includes all eligible PSR corrective code published through PSR Summary 348.

DEFICIENCIES

If DA is used in a program with overlays, it should be included in the main overlay. The DA routines must not be overlayed.

The create utility requires that Sort/Merge 4.0 be installed. If Sort/Merge is not available, comparable DA files can be created through explicit Record Manager 1.0 calls at the expense of appreciably greater creation time.

1.4.2 INSTALLATION PROCEDURE

The deadstart tape contains the released version of Direct Access 1.0. The procedure to modify and update DA is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for DA, the following materials are necessary.

KRONOS 2.1 deadstart tape

KPL2 contains the jobs to modify and update, and verify installation; the jobs required for DA 1.0 are DA1, DA2, and DA3.

DA1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.

DA2	Use to collect the DA binary code and generate directive files for later use as input to the procedure file GENSYS.
DA3	Use to verify that the product is correctly installed.

KPL6 contains DA 1.0 binary code.

PL3 contains DA 1.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.1.3.

1.4.3 INSTALLATION PARAMETERS

DA contains a single parameter value that is effective when the user does not supply his own setting.

Parameter	Released Default Value	Significance
BLKHDL EQU	1	Specifies block header length (CM words); possible range of values 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.

1.4.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

DA1

```
DA1
DA1,CM60000,T1000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB UPDATES AND CREATES THE BINARY OF DA
COMMENT.  AND COPIES 6RM AND IS TO THE NEW KPL6.
COMMENT.  THE NEW KPL6 WILL CONSIST OF NINE FILES.
COMMENT.  THE FIRST SEVEN FILES WILL BE THE OLDPLS AND BINARIES
COMMENT.  OF 6RM AND IS
COMMENT.  FILE EIGHT WILL BE THE DA NEWPL
COMMENT.  FILE NINE WILL BE THE BINARY OF DA
*VSN CARD.
LABEL(OLDPL,R,FI=RM*3P4,MT,D=HY,F=SI)
REWIND,OLDPL.
COPYBF,OLDPL,NEWPL,7.         COPY 6RM AND IS TO NEW KPL7.
UPDATE(F,P=OLDPL,N=NEWPL,W,X,R=C)
UNLOAD,OLDPL.
RETURN,OLDPL.
COMPASS,I,L=0,S=TXT6RM.
COPYBR,COMPILE,NIL.           SKIP MSD PP PROGRAM
COMPASS,I,L=0,S=IOTEXT.
COMPASS,I,L=0,S=IOTEXT,S=CPCTEXT,S=LDRTEXT,S=SMTEXT.
REWIND,LGO.
SKIPF,NEWPL.
COPYBF,LGO,NEWPL.
*VSN CARD.
LABEL(KPL6,W,FI=RM*3P4,D=HY,MT,F=SI)
REWIND,KPL6,NEWPL.
COPYEI,NEWPL,KPL6,V.
CATALOG,KPL6,R,N=0.
UNLOAD,KPL6.
*WEOR
*/ PLACE DA MODIFICATIONS HERE
*WEOF
```

DA2

```
DA2
DA2,CM55000,T2000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB ADDS THE  DA V1.0 BINARIES TO THE SYSTEM COMMON FILE
COMMENT.  ZZZPSBF FROM THE KPL6 RELEASE TAPE.  THE DA I/O MODULES ARE
COMMENT.  ADDED TO THE USER LIBRARY -SYSIO-.
COMMENT.  EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT.  JOB  DA1  MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT.  TO COMMON FILE ZZZZLDF FOR GENERARING A NEW SYSTEM DEADSTART.
COMMENT.
*VSN CARD.
LABEL(KPL6,R,FI=RM*3P4,MT,D=HY,F=SI)
SKIPF,KPL6,8.
COPYBF,KPL6,SDA.
UNLOAD,KPL6.
RETURN,KPL6.
COMMON,ZZZPSBF,ZZZZLDF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSIO
IF(EF=0)GOTO,10.
SET(R3=1)
COMMON,SYSTEM.
GTR(SYSTEM,OLD)ULIB/SYSIO
10,LIBEDIT,N=LGO,B=SDA,L=0.
LIBGEN,F,P=SYSIO.
CATALOG,SYSIO,U,R.
RETURN,NEW.
LIBEDIT,P=ZZZPSBF,B=SYSIO,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
*WEOR
*BEFORE *,REL/*
*WEOR
*BEFORE    *,ULIB/SYSIO
*WEOR
*ADD       LIB21,ULIB/SYSIO
*WEOF
```

DA3

```
DA3
DA3,CM60000,T20
*ACCOUNT CARD.
LIBRARY(SYSIO)
COMPASS(S=IOTEXT)
COMMENT. *** INSTALLATION TEST ***
COMMENT. LANGUAGE- COMPASS
COMMENT. PRODUCT- SDA
SWITCH(1)
LGO.
COMMENT. ***********TEST PASSED ***************
REWIND,ZZZZZEF.
REWIND(OUTPUT)
EXIT.
COMMENT. ***********TEST FAILED ************
REWIND,ZZZZZEF.
COPYBF(ZZZZZEF,OUTPUT)
*WEOR
          IDENT   INSTALL
          ENTRY   START
          SST
*
*
          FILE    LFN=CREATE,BFS=000,FET=XFET,HMB=10,FO=DA,FWB=000000,FLM
,=70,KA=XKA,KL=10,KP=0,MBL=64,MNR=30,MRL=52,PM=R,RKP=0,RKW=0,WSA=XWSA
 START    BSS     0
          OPENM   CREATE,NEW,R
          SX6     B0
 LOOP     SB1     1
          SA6     XWSA                STORE RECORD IN WSA
          SA6     A6+B1
          SA6     A6+B1
          SA6     A6+B1
          PUT     CREATE,,40
          SA2     COUNT
          SX6     X2+1                INCREMENT COUNTER
          SA6     A2
          SX3     60
          IX3     X3-X6
          NZ      X3,LOOP
3,LOOP
          CLOSEM  CREATE
          OPENM   CREATE,I-O,R
 LOOP2    GETN    CREATE
          SA1     EXKEY               LOAD EXPECTED KEY
          SA2     XWSA                LOAD RECORD FROM WSA
          SA3     A2+1
          SA4     A3+1
          SA5     A4+1
          IX0     X1-X2
          NZ      X0,ERR              COMPARE AGAINST EXPECTED VALUE
          IX0     X1-X3
          NZ      X0,ERR
          IX0     X1-X4
          NZ      X0,ERR
```

```
         IX0     X1-X5
         NZ      X0,ERR
         SX6     X1+10                   INCREMENT EXPECTED KEY
         SA6     A1
         SA2     COUNT2
         SA5     A4+1
         IX0     X1-X2
         NZ      X0,ERR
         SX2     X1+10
         IX0     X2-X3
         NZ      X0,ERR
         IX0     X2-X4
         NZ      X0,ERR
         IX0     X2-X5
         NZ      X0,ERR
         SX6     X1+1
         SA6     A1
         SX0     X6-60
         NZ      X0,LOOP5
         CLOSEM  CREATE
 BB      BSS     0
         ENDRUN
 ERR     MESSAGE ER,,RECALL
         ENDRUN
 ER      DATA    10LTEST FAILE
         DATA    1CD
 XWSA    BSSZ    4
SSZ   4
         DATA    -0
 XKA     BSSZ    1
 FLAG    BSSZ       1SSZ      1
 COUNT   BSSZ    1
 COUNT2  BSSZ    1
 COUNT3  BSSZ    1
 EXKEY   BSSZ    1
 BUFFER  BSSZ    200
XFET     BSSZ    10
         END     START
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
01.22.27.DA3,CM60000,T20
01.22.27.*ACCOUNT CARD.
01.22.28.ACCOUNT,USER1.
01.22.31.LIBRARY(SYSIO)
01.22.32.COMPASS(S=IOTEXT)
01.22.44.    9.445  CPU SECONDS  ASSEMBLY TIME.
01.22.44. ASSEMBLY COMPLETE.   50200B  CM USED.
01.22.44.COMMENT. *** INSTALLATION TEST ***
01.22.44.COMMENT. LANGUAGE- COMPASS
01.22.44.COMMENT. PRODUCT- SDA
01.22.45.SWITCH(1)
01.22.45.LGO.
01.23.17.COMMENT. ***********TEST PASSED ********
01.23.17.*******
01.23.17.REWIND,ZZZZZEF.
01.23.17.REWIND(OUTPUT)
```

1.5 INDEXED SEQUENTIAL 2.0

1.5.1 RELEASE DESCRIPTION

Indexed Sequential 2.0 (IS) operates as a functional module of Record Manager 1.0. It provides for the creation and maintenance of a random storage file with records that can be retrieved by key as well as sequentially. An indexed sequential file created by IS can be processed only by IS.

There are two utility routines called by control cards for indexed sequential files. SISTAT prints the statistics for an existing IS file; ESTMATE produces estimates of block and buffer sizes from input cards containing IS file descriptions.

Interface routines allow the execution of IS 1.0 programs that exist in either source language or binary format.

HARDWARE REQUIREMENTS

IS 2.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. Refer to the field length requirements for Record Manager in part III, section 1.3.1.

CORRECTIONS

IS 2.0 includes all eligible PSR corrective code published through PSR Summary 348.

DEFICIENCIES

If IS is used in a program with overlays, it should be included in the main overlay. The IS routines must be overlayed.

1.5.2 INSTALLATION PROCEDURE

The deadstart tape contains the released version of IS 2.0. The procedure to modify and update IS is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for IS, the following materials are necessary.

KPL contains the jobs to modify and update, and verify installation; the jobs required for IS 2.0 are IS1, IS2, and IS3.

IS1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
IS2	Use to collect the IS binary code and generate directive files for later use as input to the procedure file GENSYS.
IS3	Use to verify that the product is correctly installed.

KPL6 contains IS 2.0 binary code.

PL3 contains IS 2.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.1.3.

1.5.3 INSTALLATION PARAMETERS

This system contains parameter values that are effective when the user does not supply settings. The default parameters are defined on the program library tape PL3 in the common deck SISCOMM. Their definitions, default values, ranges of acceptable decimal values, and Update sequence locations are as follows:

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

1.5.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

IS1

```
IS1
IS1,CM65000,T2000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB UPDATES AND CREATES THE BINARY OF IS
COMMENT.  THE NEW KPL6 WILL CONSIST OF NINE FILES.
COMMENT.  THE FIRST FIVE FILES WILL BE 6RM
COMMENT.  FILE SIX WILL BE THE IS NEWPL
COMMENT.  FILE SEVEN WILL BE THE IS BINARY
COMMENT.  THE LAST TWO FILES WILL BE THE PL AND BINARY OF DA.
*VSN CARD.
LABEL(OLDPL,R,FI=RM*3P4,MT,D=HY,F=SI)
REWIND,OLDPL.
COPYBF,OLDPL,NEWPL,5.          COPY 6RM TO NEW KPL7.
UPDATE(F,P=OLDPL,N=NEWPL,W,R=C)
SKIPF,NEWPL.
COMPASS,I,L=0,B=NEWBIN,S=TXT6RM,S=IPTEXT.
COMPASS,I,L=0,B=NEWBIN,S=TXT6RM.

RFL,64000.
COMPASS,I,L=0,B=NEWBIN,S=TXT6RM,S=IOTEXT.
RFL,20000.
REWIND,NEWBIN.
COPYBF,NEWBIN,NEWPL.
SKIPF,OLDPL,2.
COPYBF,OLDPL,NEWPL,2.
UNLOAD,OLDPL.
RETURN,OLDPL.
*VSN CARD.
LABEL(KPL6,W,FI=RM*3P4,MT,D=HY,F=SI)
REWIND,KPL6,NEWPL.
COPYEI,NEWPL,KPL6,V.
CATALOG,KPL6,R,N=0.
UNLOAD,KPL6.
*WEOR
*/  ADD IS MODIFICATIONS HERE
*WEOF
```

IS2

```
IS2
IS2,CM55000,T2000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB ADDS THE  IS V2.0 BINARIES TO THE SYSTEM COMMON FILE
COMMENT.  ZZZPSBF FROM THE KPL6 RELEASE TAPE.   THE ESTMATE AND SISTAT
COMMENT.  PROGRAMS ARE ADDED AS OVERLAYS.  THE -IS-
COMMENT.  I/O MODULES ARE ADDED TO THE USER LIBRARY -SYSIO-.  THE
COMMENT.  PROGRAM -RUNCALL- IS ADDED TO THE USER LIBRARY -RUN2P3-.
COMMENT.  EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT.  JOB  IS1  MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT.  TO COMMON FILE ZZZZLDF FOR GENERARING A NEW SYSTEM DEADSTART.
COMMENT.
COMMENT.  W A R N I N G -- THE S I S UTILITIES  -E S T M A T E -
COMMENT. AND - S I S T A T - WILL NOT BE GENERATED PROPERLY
COMMENT. IF JOB -F4LIB2- WAS NOT RUN PRIOR TO THIS JOB AND
COMMENT. THE USER LIBRARY - F O R T R A N - IS NOT ON THE
COMMENT. SYSTEM.
COMMENT.
*VSN CARD.
LABEL(KPL6,R,FI=RM*3P4,MT,D=HY,F=SI)
SKIPF,KPL6,6.
COPYBF,KPL6,SIS.
UNLOAD,KPL6.
RETURN,KPL6.
GTR(SIS,RUN)REL/RUNCALL
GTR(SIS,ESTM)REL/ESTMATE
GTR(SIS,SIST)REL/SISTAT
COMMON,ZZZPSBF,ZZZZLDF.
COMMON,SYSTEM.
NOEXIT.
GTR(ZZZPSBF,FORLIB)ULIB/FORTRAN
IF(EF=0)GOTO,1.
SET(EF=0)
SET(R3=1)
GOTO,2.
1,LIBGEN(F=FORLIB,P=FORTRAN)
2,SET(EF=0)
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSIO
IF(EF=0)GOTO,10.
SET(R1=1)
GTR(SYSTEM,OLD)ULIB/SYSIO
10,LIBEDIT(B=SIS,L=0,N=LGO)
LIBGEN,F,P=SYSIO.
CATALOG,SYSIO,U,R.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
SET(R3=0)
SET(EF=0)
BKSP,ZZZPSBF.
REWIND,ESTM,SIST.
LIBRARY(FORTRAN)
LINK,F=ESTM,B=UTIL,P=SYSIO.
LINK,F=SIST,B=UTIL,P=SYSIO.
CATALOG,UTIL,R.
COPYBF,UTIL,ZZZPSBF.
BKSP,ZZZPSBF.
RETURN,OLD,LGO.
GTR(ZZZPSBF,OLD,,,S)ULIB/RUN2P3
IF(EF=0)GOTO,20.
COMMON,SYSTEM.
```

```
SET(R2=1)
GTR(SYSTEM,OLD)ULIB/RUN2P3
20,LIBEDIT,N=LGO,B=RUN,L=0.
LIBGEN,F,P=RUN2P3.
CATALOG,RUN2P3,U,R.
RETURN,NEW,LGO.
REWIND,SYSIO,RUN2P3.
COPYEI,SYSIO,LGO.
BKSP,LGO.
COPYEI,RUN2P3,LGO.
REWIND,LGO.
LIBEDIT,P=ZZZPSBF,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
IF(R1≠0)COPYBR,,ZZZZLDF.
IF(R1=0)SKIPR,.
IF(R2≠0)COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
*WEOR
*IGNORE   REL/RUNCALL,ESTMATE,SISTAT
*BEFORE *,REL/*
*WEOR
*ADD       LIB29,OVL/ESTMATE,SISTAT
*WEOR
*BEFORE    *,REL/RUNCALL
*WEOR
*BEFORE    *,ULIB/SYSIO
*BEFORE    *,ULIB/RUN2P3
*WEOR
*ADD       LIB21,ULIB/SYSIO
*WEOR
*ADD       LIB20,ULIB/RUN2P3
*WEOF
```

IS3

```
IS3
IS3,CM60000,T20.
*ACCOUNT CARD.
COMMENT.   ******** S I S   V 1     ************
COMMENT.   ******** VALIDATION      ************
***********
COMPASS(S=CPCTEXT)
LIBRARY(SYSIO)
LGO.
COMMENT.   ***** T E S T  P A S S E D  *******
EXIT.
REWIND,SISMESS.
COPYBF(SISMESS,OUTPUT)
COMMENT.  ***** TEST FAILED *******
*WEOR
          IDENT   V1INSTL
          ENTRY   START

 SISFILE  SETFET DATREC,20,20,CMBUF,500

 START    SB1     1
          SETKEY SISFILE,KEYLOC,0,S,10
          SETBLKI  SISFILE,0,63
          SETBLKD  SISFILE,0,0,127
          SETCOLL  SISFILE,DC,CD
          SETERR SISFILE,ERRCODE
          OPENNEW  SISFILE
          SX4     20
          PX6     B0,X4
          SA6     RECSIZE               INITIALIZE RECORD SIZE
 LL0      INSERT  SISFILE,RECSIZE
          SA1     COUNT
          SX6     X1-1                  DECREMENT RECORD COUNT
          SA6     A1
          ZR      X6,LL1
6,LL1
          SA1     KEYLOC
          SX6     X1+1
          SA6     KEYLOC                INCREMENT KEY
          SA1     DATREC
          SX6     X1+1
          SA6     A1                    INCREMENT DATA RECORD
          SA6     DATREC+1
          EQ      LL0
 LL1      TERMNAT  SISFILE

          SA1     KEY1
          BX6     X1
          SA6     KEYLOC                RESET KEY
          OPENOLD  SISFILE,R
          SEEKS   SISFILE
          SEEKL  SISFILE
          ACCESSK  SISFILE,RECSIZE
```

```
  SX1     1
  PX6     B0,X1
  SA6     P1                   INITIALIZE REPOS PARAMETER
  REPOS   SISFILE,P1
  ACCESSN   SISFILE,RECSIZE
  REPLACE   SISFILE,RECSIZE
  DELETE  SISFILE
          FORCEW  SISFILE
          TERMNAT  SISFILE
          ENDRUN
 DATREC   DATA    1
          DATA    1
 KEYLOC   DATA    10LKAAAAAAAAA
 CMBUF    BSSZ    500
 ERRCODE  BSSZ    1
 DC       DATA    77303132333435360000B
          DATA    37404243444546470000B
          DATA    50515254555657600000B
          DATA    61626365666770710000B
          DATA    72737475761621200000B
          DATA    22241417250023130000B
          DATA    04025364260341050000B
          DATA    06072710011112150000B
 CD       DATA    55746165606770710000B
          DATA    73757657527745530000B
          DATA    47465056515464720000B
          DATA    01020304050607100000B
          DATA    11661213141516170000B
          DATA    20212262232425260000B
          DATA    27303132633334350000B
          DATA    36374041424544000000B
 RECSIZE  BSSZ    1
 COUNT    DATA    10
 KEY1     DATA    10LKAAAAAAAAA
 P1       BSS     1
          END     START
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
01.05.14.IS3,CM60000,T20.
01.05.14.*ACCOUNT CARD.
01.05.15.ACCOUNT,USER1.
01.05.16.COMMENT.   ********* S I S   V 1     *****
01.05.16.*******
01.05.16.COMMENT.   ********* VALIDATION      *****
01.05.16.*******
01.05.19.COMPASS(S=CPCTEXT)
01.05.29.    5.170  CPU SECONDS  ASSEMBLY TIME.
01.05.29. ASSEMBLY COMPLETE.   43600B  CM USED.
01.05.29.LIBRARY(SYSIO)
01.05.29.LGO.
01.05.39.COMMENT.   ***** T E S T  P A S S E D  *
01.05.39.******
```

1.6 UPDATE

1.6.1 RELEASE DESCRIPTION

HARDWARE REQUIREMENTS

Update requires the same minimum hardware requirements as does KRONOS 2.1.

CORRECTIONS

Update can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length requirement to run Update is $35,000_8$.

DEFICIENCIES

None.

1.6.2 INSTALLATION PROCEDURE

The deadstart tape contains the released version of Update. The procedure to modify and update Update is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for Update, the following materials are necessary.

KPL2 contains the jobs to modify and update; the jobs required for Update are UPD1 and UPD2.

UPD1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
UPD2	Use to collect the Update binary code and generate directive files for later use as input to the procedure file GENSYS.

KPL7 contains the Update binary code.

PL1C contains the Update source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.1.4.

1.6.3 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. Specify these changes in the installation deck UPD1.

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 card is not recognized as a legal directive.

Parameter	Released Default Value	Significance
DECLKEY	Enabled	Enables DECLARE directive
PMODKEY	Enabled	Enables PULLMOD card 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 KRONOS 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.

1.6.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

UPD1

```
UPD1
UPD1,CM54000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF UPDATE 1.2 ON KPL7.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=UPDATE1P2,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
RETURN,OLDPL.
COMPASS,I,S=IPTEXT,B=NEWPL,L=0.
WRITEF,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL7,W,FI=UPDATE1P2,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL7,V.
*WEOR
*/         PLACE *UPDATE* MODSET HERE.
*WEOF
```

UPD2

```
UPD2
UPD2,CM15000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL7,R,FI=UPDATE1P2,MT,D=HY,F=SI)
SKIPF,KPL7.
COPYBF,KPL7,UPDATE.
UNLOAD,KPL7.
RETURN,KPL7.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBR,,ZZZZLDF.
CATALOG,UPDATE,R.
COPYBF,UPDATE,ZZZPSBF.
BKSP,ZZZPSBF.
*WEOR
*ADD       LIB13,OVL/UPDATE
*WEOF
```

1.7 SCOPE 3.4 SYSTEM TEXTS

1.7.1 RELEASE DESCRIPTION

KPL4 consists of the SCOPE 3.4 COMDECKs that comprise the eight text decks and additional programs required to install and maintain the common product set.

HARDWARE REQUIREMENTS

Maintenance of KPL4 requires the same minimum hardware requirements as does KRONOS 2.1.

CORRECTIONS

KPL4 includes all eligible PSR corrective code published through PSR Summary 348.

DEFICIENCIES

None.

1.7.2 INSTALLATION PROCEDURE

The deadstart tape contains the released version of SCOPE 3.4 Texts. The procedure to modify or update SCOPE 3.4 Texts is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for SCOPE 3.4 Texts, the following materials are necessary.

KPL2 contains the jobs to modify and update; the jobs required for SCOPE 3.4 Texts are SCP1 and SCP2.

SCP1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
SCP2	Use to collect the SCOPE 3.4 Texts binary code and generate directive files for later use as input to the procedure file GENSYS.

KPL4 contains the SCOPE 3.4 Texts binary and source code.

The tapes are described in detail in part I, section 2.1.1.

CAUTION

The KRONOS modifications contained in job SCP1 is in the binary code on KPL4 and on the KRONOS 2.1 deadstart tape. It is not in the source code on KPL4, in order to prevent any future modification conflicts with SCOPE 3.4 PSR summaries. These modifications contain the changes to SCOPE 3.4 COMDECK COMCSYS; it resolves incompatibilities between SCOPE 3.4 and KRONOS 2.1.

1.7.3 INSTALLATION PARAMETERS

General installation parameters related to the common product set are defined within the COMDECK IPARAMS.

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.

The following list constitutes the extent of installation changeable symbols in IPARAMS. Those parameters preceded by an asterisk (*) have been changed for KRONOS 2.1 and are reflected in IPARAMS.

Parameter	Released Default Value	Significance
IP.CMU	0	If nonzero, compare/move unit hardware is present.

The following common product routines reference IP.CMU.

Record Manager 1.0: TXTERM

COBOL 4.0:

```
DCSLEMV  DCCPWA   CCCOBIO DCSBLIT DDUFCNT  CCADVAN
DCSLESC  DDSESC2  DDSBSC6 DDSBSC7 DDDSPLY  CCCADD
CCNSTAN  CDTRUBL  DDCVBD  DBN1SA  DDBN     DBN6SA
DENESS   BN1CSA   DBN1CS  BNEDSA  DBN6DS   DB1DCA
DBN1CD   DB6DDA   DBN6DD  CBN1SB  BN1SBA   DDEXAMO
DDBCDCM  CDSCM    CONV1   CONTROL SAVEREG  SNAP
SCAN2    ITEMCOP  SAD     LEXDATA PASS1B   01B
CCNCRDI  LEXXY    PASS1B1 01B1    PASS1C   PASS1D
LEXFFCC  FASS1E   DIG     C1E     PEF      PASS1F
ART      MNEMON   PASS1G  TROUT1G DISCIO   PASS1H
GOTCGEN  GENPLIM  PRFOPS  TCLIMB  GENLOC   GENSTO
PASS2    CHECK    GENARTH GENMOVE GENIF    LIT02
GENCISF  SUBSCR   GREPORT
```

Sort/Merge 4.0:

```
KYCFL    KEYCCD  BUFALL TRNSRT  TSC    TMC
FMC      FMIP    FMOP   FSRTGET MRGCON MRGDSN
SRTCCN   SRTPUT  SRTGET
```

Parameter	Released Default Value	Significance
*IP.CSET	IP.C63	Defines character set to be used throughout the system. The following common product routines reference IP.CSET. FORTRAN Extended 4.0 Compiler: SISCFEN FORTRAN Extended 4.0 Library: KODER= KRAKER COBOL 4.0: DDSLBMV CONV1 GENMOVE GENIF Sort/Merge 4.0: ENDPRO ALGOL 3.0: ALG1 SIMSCRIPT 3.0: SIMLLA SIM1 SIMURUN SIMLB01 FORTRAN 2.3: INPUTS READMS BACKSP BUFFEI BUFFEO ENDFIL IOCHEC IFENDF INPUTB INPUTC IOCHEK KODER KRAKER OUTPTB OUTPTC REWINM SIC$
IP.IMUL	0	If nonzero, integer multiply hardware is present. The following common product routines reference IP.IMUL. COBOL 4.0: DDSLBMV DDOPWA CCCOBIO DDSBLIT DDUPCNT DDADVAN DDSLESC DDSESC2 DDSBSC6 DDSBSC7 DDDSPLY DDDADD CONSTAN DDTRUBL DDCVBC DBN1SA DDEN DEN6SA DBNESS BN1DSA DBN1DS BN6DSA DBN6DS DE1DDA DBN1DD DB6DDA DBN6DD DBN1SB BN1SBA DDEXAMO DDBCCCM DDSCM CONV1 CONTROL SAVEREG SNAP SCAN2 ITEMCOP SAD LEXDATA PASS1B 01B DDNCRDI LEXXY PASS1B1 01B1 PASS1C PASS1D LEXPROC PASS1E DIG 01E REF PASS1F ART MNEMON PASS1G TROUT1G DISCIO PASS1H GCTCGEN GENFLIM PRFOPS TCLIMB GENLOC GENSTO PASS2 CHECK GENARTH GENMOVE GENIF LIT02 GENDISP SUBSCR GREPORT

Parameter	Released Default Value	Significance
IP.TYPE	6600	Determines the type of central processor to be used by the system for generation of optimal code. Acceptable values are 6400 (CDC CYBER Model 72) or 6600 (CDC CYBER Model 74) The following common product routines reference IP.TYPE. COBOL 4.0:

```
DDSUEMV  DDOPWA    CCCOBIO   CDSBLIT   DDUPCNT   DCADVAN


CCSLESC  CDSESC2   DDSBSC6   CDSBSC7   DDCSPLY   DCCADD
CCNSTAN  DDTRUBL   DDCVBD    CBN1SA    DDEN      DEN6SA
CENESS   BN1CSA    DBN1DS    EN6DSA    DBNEDS    DE1DDA
DEN1CD   CB6DDA    DEN6DD    CBN1SB    BN1SBA    DCEXAMO
DCBCCCM  DDSCM     CONV1     CONTRCL   SAVEREG   SNAP
SCAN2    ITEMCOP   SAD       LEXDATA   PASS1B    01B
CCNCRDI  LEXXY     PASS1E1   C1E1      PASS1C    PASS1D
LEXPROC  PASS1E    DIG       01E       REF       PASS1F
ART      MNEMCN    PASS1G    TRCUT1G   CISCIC    PASS1H
GCTCGEN  GENFLIM   PRFOPS    TCLIMB    GENLOD    GENSTO
PASS2    CHECK     GENARTH   GENMOVE   GENIF     LIT02
GENCISF  SUBSCR    GREPORT
```

*OS.NAME	KRONOS	The common product routine that references OS.NAME is FTNXAS$ from the FORTRAN Extended 4.0 compiler.
*OS.VER	2.1	The common product routine that references OS.VER is FTNXAS$ from the FORTRAN Extended 4.0 compiler.

1.7.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

SCP1

```
SCP1
SCP1,CM55000,P30,T1000.
*ACCOUNT CARD
*THIS JOB CREATES NEW PROGRAM LIBRARIES
*AND BINARIES OF SCOPE 3.4 SYSTEMS TEXTS
*AND ASSORTED LIBRARY ROUTINES ON KPL4.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=SCOPE3P4TEXTS*2P1,MT,D=HY,F=SI)
UPDATE,F,N,W,R,X.
SKIPF,OLDPL.
UPDATE,F,N=PL1B,W,R=N,C=0.
UPDATE,F,P=PL1B,R=P,X,L=0.
UNLOAD,OLDPL.
REWIND,COMPILE.
RETURN,OLDPL.
SKIPEI,NEWPL.
COMPASS,I,S=0,L=0.
GTR(LGO,IP)OVL/IPTEXT
COPYBF,PL1B,NEWPL.
GTR(LGO,CPU)OVL/CPUTEXT
COPYBF,LGO,NEWPL.
BKSP,NEWPL.
COMPASS,I,S=0,B=NEWPL,L=0.
COMPASS,I,G=IP,G=CPU,B=NEWPL,L=0.
WRITEF,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL4,W,FI=SCOPE3P4TEXTS*2P1,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL4,V.
*WEOR
*/       PLACE *SCOPE 3.4 PL1A* MODSET HERE.
*WEOR
*/       PLACE *SCOPE 3.4 PL1B* MODSET HERE.
*WEOR
*IDENT   KRONOS
*D,COMCSYS.135,COMCSYS.146
         BX6   X6+X1
         SA6   MSGA
         MX1   0
         AX6   18
         SX6   X6+
         ZR    X6,MSG2
         SX1   3
         AX6   1
         ZR    X6,MSG2
         AX1   1
 MSG2    BX6   X1
```

```
X6    X1
          SA1    MSGA
          SA6    A1
          MX6    42
          BX6    -X6*X1
          SA1    A6
          LX6    -18
          BX1    X1+X6
          LX1    37
          EQ     MSG1
*I,SYSEQ.24
          ENTRY  CIO=
*I,SYSEQ.27
*CALL     COMCCIO
OMCCIO
*WEOF
```

SCP 2

```
SCP2
SCP2,CM45000,P30,T1000.
*ACCOUNT CARD
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL4,R,FI=SCOPE3P4TEXTS*2P1,MT,D=HY,F=SI)
SKIPF,KPL4,2.
COPYBF,KPL4,BIN.
RETURN,KPL4.
GTR(BIN,TEXTS)OVL/*
GTR(BIN,LIB)REL/*
CATALOG,TEXTS,R.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBF,TEXTS,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSIO
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R3=1)
GTR(SYSTEM,OLD)ULIB/SYSIO
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=SYSIO,NX=1.
CATALOG,SYSIO,R,U.
LIBEDIT,P=ZZZPSBF,B=SYSIO,C,L=0.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
*WEOR
*BEFORE   *,REL/*
*WEOR
*BEFORE   *,ULIB/SYSIO
*WEOR
*ADD      LIB15,OVL/CPCTEXT-PFMTEXT
*WEOR
*ADD      LIB21,ULIB/SYSIO
*WEOF
```

1.8 MAINTENANCE TOOLS

1.8.1 RELEASE DESCRIPTION

The maintenance tools include the Time-Sharing Stimulator (STIMULA and 1TS), the dayfile sort program (DFSORT), the FORTRAN Translator (FTNTRAN), the P register analyzer (PSAMP), and the object libraries for COBOL 3.0, FORTRAN Extended 3.0, and Sort/Merge 3.0.

The procedure to use the Time-Sharing Stimulator is in part IV, section 5.

DFSORT is used to sort the dayfile for the purpose of running accounting programs that expect DAYFILE to be sorted by specific keys.

FTNTRAN translates or converts source programs written in G.E. FORTRAN to source programs acceptable to Time-Sharing FORTRAN.

PSAMP analyzes the P register.

The object libraries allow relocatable binary code created under KRONOS 2.0 to satisfy external symbols from the version 3.0 libraries. These libraries do not use Record Manager for input/output; they retain calls to CIO. These routines are only included to allow the use of 2.0 binary decks during the 2.0 to 2.1 transition phase.

Use of the version 3.0 object libraries under KRONOS 2.1 requires an LDSET (LIB=SYSMISC/SYSIO) control card before the job's EXECUTE control card.

HARDWARE REQUIREMENTS

Maintenance tools can be maintained on the same minimum hardware configuration as KRONOS 2.1.

CORRECTIONS

The version 3.0 object libraries include all eligible PSR corrective code published through PSR Summary 348.

DEFICIENCIES

None.

1.8.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update the maintenance tools information is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for the maintenance tools, the following materials are necessary.

KPL1 contains the KRONOS 2.1 system source code; it is required for job MTN1 only.

KPL2 contains the jobs to modify, update, and install; the jobs required for the maintenance tools are MTN1, MTN2, F3L1, F3L2, C3L1, C3L2, S3L1, and S3L2.

MTN1 Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code for STIMULA, 1TS, DFSORT, and FTNTRAN.

MTN2 Use to collect the STIMULA, 1TS, DFSORT, FTNTRAN, and PSAMP binary code and generate directive files for later use as input to the procedure file GENSYS.

F3L1 Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code for FORTRAN Extended 3.0 object library. This job must be run on a KRONOS 2.0 system with FORTRAN Extended 3.0.

F3L2 Use to collect the FORTRAN Extended 3.0 object library binary code and generate directive files for later use as input to the procedure file GENSYS.

C3L1 Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code for COBOL 3.0 object library.

C3L2 Use to collect the COBOL 3.0 object library binary code and generate directive files for later use as input to the procedure file GENSYS.

S3L1 Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code for Sort/Merge 3.0 object library.

S3L2 Use to collect the Sort/Merge 3.0 object library binary code and generate directive files for later use as input to the procedure file GENSYS.

KPL3 contains the binary and source code for STIMULA, 1TS, DFSORT, FTNTRAN, PSAMP, FORTRAN Extended 3.0 object library, COBOL 3.0 object library, and Sort/Merge 3.0 object library.

The tapes are described in detail in part I, sections 2.1.1 and 2.1.5.

1.8.3 INSTALLATION PARAMETERS

None.

1.8.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

MTN1

```
MTN1
MTN1,T1000,CM64000.
*ACCOUNT CARD.
COMMENT.  THIS JOB MODIFYS AND CREATES THE BINARIES OF THE
COMMENT.  FIVE MAINTENANCE PROGRAMS -- 1TS, STIMULA, FTNTRAN,
COMMENT.  DFSORT, AND PSAMP.
COMMENT.  THE NEW KPL3 WILL CONTAIN 8 FILES.
COMMENT.  FILES 1 AND 2 --- PL AND BINARIES FOR MAINTENANCE TOOLS.
COMMENT.  FILES 3 AND 4 --- V3.0 FTN PL AND BINARIES FOR OBJECT LIBRARY.
COMMENT.  FILES 5 AND 6 --- V3.0 COBOL PL AND BINARIES FOR OBJECT LIBARY
COMMENT.  FILES 7 AND 8 --- V3.0 SORT PL AND BINARY FOR OBJECT LIBRARY.
COMMENT.  THE FIRST FILE OF THE NEW KPL3 WILL BE THE NEWPL
COMMENT.  OF THE FIVE MAINTENANCE TOOLS PROGRAMS ONLY.
COMMENT.  THE SECOND FILE WILL BE THE ABSOLUTE BINARIES OF
COMMENT.  THE FIVE MAINTENANCE TOOLS PROGRAMS ONLY.
COMMENT.  FILES 3 - 8 WILL BE COPIED FROM THE OLD KPL3.
COMMENT.
REQUEST(OLDPL,LB=KU,F=X,D=HY)
REWIND,OLDPL.
COPYBF,OLDPL,MTNPL.
SKIPF,OLDPL.
COPYBF,OLDPL,NKPL3,6.
UNLOAD,OLDPL.
RETURN,OLDPL.
*GET(MOD=MTNMOD)
COMMON,OPL.
MODIFY,Z,N,LO=E./*OPLFILE MTNPL/*EDIT 1TS.PSAMP
COMPASS,I,L=0.
RUN23,S,,,COMPILE,,BIN.
RETURN,OUTPUT.
LINK,F=BIN,B.
REWIND,LGOB.
COPYBF,LGOB,LGO.
REWIND,LGO,NPL,NKPL3.
REQUEST(KPL3,LB=KU,F=X,D=HY)
REWIND,KPL3.
COPYBF,NPL,KPL3.
COPYBF,LGO,KPL3.
COPYBF,NKPL3,KPL3,6.
REWIND,KPL3,NKPL3,NPL,LGO.
VERIFY,NPL,KPL3.
VERIFY,LGO,KPL3.
VERIFY,NKPL3,KPL3,N=6.
CATALOG,KPL3,R,N=0.
UNLOAD,KPL3.
*WEOR
*READ MOD,*
*WEOF
```

MTN2

```
MTN2
MTN2,T1000,CM45000.
*ACCOUNT,CARD.
COMMENT.  THIS JOB ADDS THE BINARIES OF THE FIVE MAINTENANCE TOOLS
COMMENT.  PROGRAMS -(1TS, STIMULA, FTNTRAN, DFSORT AND PSAMP) TO
COMMENT.  THE SYSTEM COMMON FILE ZZZPSBF FROM THE LATEST
COMMENT.  KPL3 TAPE.  EITHER THE RELEASED VERSION OR THE
COMMENT.  LATEST VERSION CREATED BY JOB MTN1 MAY BE USED.
COMMENT.  IT ALSO ADDS THE NECESSARY DIRECTIVES TO
COMMENT.  TO COMMON FILE ZZZZLDF FOR GENERARING A NEW SYSTEM DEADSTART.
REQUEST(KPL3,LB=KU,F=X,D=HY)
SKIPF,KPL3.
COPYBF,KPL3,MAINT.
REWIND,MAINT.
UNLOAD,KPL3.
RETURN,KPL3.
COMMON,ZZZPSBF,ZZZZLDF.
COPYBF,MAINT,ZZZPSBF.
CATALOG,MAINT,R.
BKSP,ZZZPSBF.
COPYBR,,ZZZZLDF.
*WEOR
*ADD      LIB10,ABS/STIMULA,PP/1TS
*ADD      LIB18,OVL/DFSORT,FTNTRAN,PSAMP
*WEOF
```

F3L1

```
F3L1
F3L1,CM60000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB CAN ONLY RUN WITH FTN V3.0
COMMENT. AND WILL PRODUCE INCORRECT
COMMENT. RESULTS IF RUN WITH FTN V4.0.
COMMENT. THIS JOB UPDATES AND ASSEMBLES THE FTN EXTENDED V3.0 OBJECT
COMMENT. LIBRARY ROUTINES AND CREATES A NEW KPL3.
COMMENT. THE NEW KPL3 WILL CONTAIN 8 FILES.
COMMENT. FILES 1 AND 2 --- PL AND BINARIES FOR THE MAINTENANCE PROGRAMS
COMMENT.                   -1TS, STIMULA, FTNTRAN, DFSORT, AND PSAMP.
COMMENT. FILES 3 AND 4 --- V3.0 FTN PL AND BINARIES FOR OBJECT LIBRARY.
COMMENT. FILES 5 AND 6 --- V3.0 COBOL PL AND BINARIES FOR OBJECT LIB.
COMMENT. FILES 7 AND 8 --- V3.0 SORT PL AND BINARY FOR OBJECT LIBRARY.
REQUEST(OLDPL)
COPYBF,OLDPL,NKPL3,2.
UPDATE,F,W,N=NKPL3,R=C.
SKIPF,OLDPL,2.
FTN(SYSEDIT=IDENT,I=COMPILE,S=IPTEXT,L=0)
REWIND,LGO.
WRITEF,NKPL3.
COPYBF,LGO,NKPL3.
COPYBF,OLDPL,NKPL3,4.
UNLOAD,OLDPL.
RETURN,OLDPL.
REQUEST(KPL3)
REWIND,NKPL3.
COPYBF,NKPL3,KPL3,8.
CATALOG,KPL3,R,N=8.
UNLOAD,KPL3.
*WEOR
*/          PLACE * FTN V3.0 LIBARY* MODSET HERE.
*WEOF
```

F3L2

```
F3L2
F3L2,CM45000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB ADDS THE FTN V3.0 OBJECT LIBRARY TO THE SYSTEM
COMMENT. COMMON FILE ZZZPSBF FROM THE KPL3 RELEASE TAPE.
COMMENT. EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT. JOB F3L1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT. TO COMMON FILE ZZZZLDF FOR GENERATING A NEW SYSTEM DEADSTART.
REQUEST(KPL3,LB=KU,F=X,D=HY)
SKIPF,KPL3,3.
COPYBF,KPL3,LIB.
UNLOAD,KPL3.
RETURN,KPL3.
COMMON,ZZZPSBF,ZZZZLDF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSMISC
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R1=1)
GTR(SYSTEM,OLD)ULIB/SYSMISC
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=SYSMISC.
CATALOG,SYSMISC,R,U.
RETURN,NEW.
LIBEDIT,P=ZZZPSBF,B=SYSMISC,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
COMMON,ZZZZLDF.
IF(R1=1)COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
*WEOR
*BEFORE   *,REL/*
*WEOR
*BEFORE   *,ULIB/SYSMISC
*WEOR
*ADD      LIB21,ULIB/SYSMISC
*WEOF
```

C3L1

```
C3L1
C3L1,CM60000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB UPDATES AND ASSEMBLES THE COBOL V3.0 OBJECT
COMMENT. LIBRARY ROUTINES AND CREATES A NEW KPL3.
COMMENT. THE NEW KPL3 WILL CONTAIN 8 FILES.
COMMENT. FILES 1 AND 2 --- PL AND BINARIES FOR THE MAINTENANCE PROGRAMS
COMMENT.                   -1TS, STIMULA, FTNTRAN, DFSORT, AND PSAMP.
COMMENT. FILES 3 AND 4 --- V3.0 FTN PL AND BINARIES FOR OBJECT LIBRARY.
COMMENT. FILES 5 AND 6 --- V3.0 COBOL PL AND BINARIES FOR OBJECT LIB.
COMMENT. FILES 7 AND 8 --- V3.0 SORT PL AND BINARY FOR OBJECT LIBRARY.
REQUEST(OLDPL)
COPYBF,OLDPL.NKPL3,4.
UPDATE,F,W,N=NKPL3,R=C.
SKIPF,OLDPL,2.
COMPASS,I,S=CPCTEXT,S=IPTEXT,L=0.
REWIND,LGO.
WRITEF,NKPL3.
COPYBF,LGO,NKPL3.
COPYBF,OLDPL,NKPL3,2.
UNLOAD,OLDPL.
RETURN,OLDPL.
REQUEST(KPL3)
REWIND,NKPL3.
COPYBF,NKPL3,KPL3,8.
CATALOG,KPL3,R,N=8.
UNLOAD,KPL3.
*WEOR
*/        PLACE *COBOL V3.0 LIBRARY* MODSET HERE.
*WEOF
```

C3L2

```
C3L2
C3L2,CM45000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB ADDS THE COBOL V3.0 OBJECT LIBRARY TO THE SYSTEM
COMMENT. COMMON FILE ZZZPSBF FROM THE KPL3 RELEASE TAPE.
COMMENT. EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT. JOB C3L1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT. TO COMMON FILE ZZZZLDF.
REQUEST(KPL3,LB=KU,F=X,D=HY)
SKIPF,KPL3,5.
COPYBF,KPL3,LIB.
UNLOAD,KPL3.
RETURN,KPL3.
COMMON,ZZZPSBF,ZZZZLDF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSMISC
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R1=1)
GTR(SYSTEM,OLD)ULIB/SYSMISC
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=SYSMISC.
CATALOG,SYSMISC,R,U.
RETURN,NEW.
LIBEDIT,P=ZZZPSBF,B=SYSMISC,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
IF(R1=1)COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,ULIB/SYSMISC
*WEOR
*ADD       LIB21,ULIB/SYSMISC
*WEOF
```

S3L1

```
S3L1
S3L1,CM60000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB UPDATES AND ASSEMBLES THE SORT/MERGE V3.0 OBJECT
COMMENT. LIBRARY ROUTINES AND CREATES A NEW KPL3.
COMMENT. THE NEW KPL3 WILL CONTAIN 8 FILES.
COMMENT. FILES 1 AND 2 --- PL AND BINARIES FOR THE MAINTENANCE PROGRAMS
COMMENT.                   -1TS, STIMULA, FTNTRAN, DFSORT, AND PSAMP.
COMMENT. FILES 3 AND 4 --- V3.0 FTN PL AND BINARIES FOR OBJECT LIBRARY.
COMMENT. FILES 5 AND 6 --- V3.0 COBOL PL AND BINARIES FOR OBJECT LIB.
COMMENT. FILES 7 AND 8 --- V3.0 SORT PL AND BINARY FOR OBJECT LIBRARY.
REQUEST(OLDPL)
COPYBF,OLDPL,NKPL3,6.
UPDATE,F,W,N=NKPL3,R=C.
SKIPF,OLDPL,2.
COMPASS,I,S=CPCTEXT,S=IPTEXT,L=0.
REWIND,LGO.
WRITEF,NKPL3.
COPYBF,LGO,NKPL3.
UNLOAD,OLDPL.
RETURN,OLDPL.
REQUEST(KPL3)
REWIND,NKPL3.
COPYBF,NKPL3,KPL3,8.
CATALOG,KPL3,R,N=8.
UNLOAD,KPL3.
*WEOR
*/        PLACE *SORT V3.0 LIBRARY* MODSET HERE.
*WEOF
```

S3L2

```
S3L2
S3L2,CM45000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB ADDS THE SORT V3.0 OBJECT LIBRAY TO THE SYSTEM
COMMENT. COMMON FILE ZZZPSBF FROM THE KPL3 RELEASE TAPE.
COMMENT. EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT. JOB S3L1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT. TO COMMON FILE ZZZZLDF.
REQUEST(KPL3,LB=KU,F=X,D=HY)
SKIPF,KPL3,7.
COPYBF,KPL3,LIB.
UNLOAD,KPL3.
RETURN,KPL3.
COMMON,ZZZZLDF,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSMISC
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R1=1)
GTR(SYSTEM,OLD)ULIB/SYSMISC
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=SYSMISC.
CATALOG,SYSMISC,R,U.
RETURN,NEW.
LIBEDIT,P=ZZZPSBF,B=SYSMISC,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
IF(R1=1)COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,ULIB/SYSMISC
*WEOR
*ADD       LIB21,ULIB/SYSMISC
*WEOF
```

ALGOL 3.0 2

2.1 RELEASE DESCRIPTION

2.1.1 HARDWARE REQUIREMENTS

ALGOL 3.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length to run ALGOL 3.0 is $27,000_8$.

2.1.2 CORRECTIONS

ALGOL 3.0 includes all eligible PSR corrective code published through PSR Summary 348.

2.1.3 DEFICIENCIES

The KRONOS control card REDUCE cannot be used when ALGOL programs are executed because they use the space following the program as the stack area for all variables and for input-output buffers.

Segment mode loading (ALGOL control card options S, U, R, and G) is deactivated; attempts to use these options result in job termination.

2.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update (ALGOL 3.0) is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape).

To use these procedures to install, modify, and/or update ALGOL 3.0, the following release materials are necessary.

KPL2 contains the job to modify, update, install, and verify installation; the jobs required for ALGOL 3.0 are ALG1, ALG2, and ALG3.

ALG1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
ALG2	Use to collect the ALGOL 3.0 binary code and generate directive files for later use as input to the procedure file GENSYS.
ALG3	Use to verify that the product is correctly installed.

KPL8 contains the ALGOL 3.0 binary code.

PL18 contains the ALGOL 3.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.2.

2.3 INSTALLATION PARAMETERS

Installation options are available in the following areas.

Default compiler options

Default execution-time options

Parameter	Released Default Value	Significance
ALGOL	L,X,F	The default compiler options are handled by the macro ALGOL. The parameters of the ALGOL macro are the compiler control card options required to be on by default (without specification). The release tape contains the following ALGOL call. ALGOL L,X,F To change these compiler defaults, modify the macro parameters as follows: *DELETE,V3CCARD.115 ALGOL default options *COMPILE,ALGOL,ALGO
OPTIONS	S=0,D=0,C=61,E=V	The default execution time options are handled by the macro OPTIONS. The parameters are execution time options required to be on by default (without specification). The parameters are provided in the same format as on an OPTIONS card. The release tape contains the following OPTIONS call. OPTIONS S=0,D=0,C=61,E=V To change these defaults, modify the parameters as follows: *DELETE,V3DEFB0.156 OPTIONS default options *COMPILE,ALGLB00

ALGOL 3.0 routines that reference SCOPE 3.4 COMDECK IPARAMS are listed in part III, section 1.7.3.

2.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

2.4.1 ALG1

```
ALG1
ALG1,CM56000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF ALGOL 3.0 ON KPL8.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=ALGOL3P0*3P4,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
RETURN,OLDPL.
COMPASS,I,S=CPCTEXT,S=IPTEXT,L=0.
COMPASS,I,S=SCPTEXT,S=CPCTEXT,S=IPTEXT,B=LIB,L=0.
UPDATE,Q,P=NEWPL,L=0.
MAP.
SKIPEI,NEWPL.
COMPASS,I,S=0,B=NEWPL,L=0.
LINK,F,B=NEWPL.
SKIPEI,NEWPL.
REWIND,LIB,LGO.
WRITEF,NEWPL.
COPYBF,LIB,NEWPL.
COPYBF,LGO,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL8,W,FI=ALGOL3P0*3P4,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL8,V.
*WEOR
*/         PLACE *ALGOL* MODSET HERE.
*WEOR
*IDENT     KALG
*B         AL30053.1
           IDENT  ALGTEXT
           STEXT
*D         ALGSYS2.1,ALGSYS2.3
*C         ALGTEXT
*WEOF
```

2.4.2 ALG2

```
ALG2
ALG2,CM45000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL8,R,FI=ALGOL3P0*3P4,MT,D=HY,F=SI)
SKIPF,KPL8.
COPYBF,KPL8,ALGOL.
COPYBF,KPL8,LIB.
UNLOAD,KPL8.
RETURN,KPL8.
CATALOG,ALGOL,R.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBF,ALGOL,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSMISC
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R3=1)
GTR(SYSTEM,OLD)ULIB/SYSMISC
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=SYSMISC.
CATALOG,SYSMISC,R,U.
LIBEDIT,P=ZZZPSBF,B=SYSMISC,C,L=0.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,ULIB/SYSMISC
*WEOR
*ADD       LIB15,OVL/ALGTEXT
*ADD       LIB24,OVL/ALGOL-ALG5
*WEOR
*ADD       LIB21,ULIB/SYSMISC
*WEOF
```

2.4.3 ALG3

```
ALG3
ALG3,CM56000,P30,T100.
*ACCOUNT CARD.
ALGOL(L,X)
LGO.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
VERIFY
≠BEGIN≠
≠PROCEDURE≠ TELLEM(STRING).,  ≠STRING≠ STRING.,
      OUTPUT(61,≠(≠50S≠)≠,STRING).,
TELLEM(≠(≠ALGOL IS BEST≠)≠).,
≠END≠
         ≠EOP≠
*WEOR
CHANNEL,END
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.32.02.ALG3,CM56000,P30,T100.
00.32.02.ACCOUNT,USER1.
00.32.03.*ACCOUNT CARD.
00.32.03.ALGOL(L,X)
00.32.05.
00.32.05.LGO.
00.32.07.*END OF JOB.
```

3.1 RELEASE DESCRIPTION

APL 1.0 consists of three modules: APLSYS, APLB, and APLT. APLSYS is the overlay loader and is resident in the running system. APLB and APLT are overlay files used to process batch and terminal input; they are direct access permanent files accessed under user number LIBRARY.

3.1.1 HARDWARE REQUIREMENTS

APL 1.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length requirement to run APL 1.0 is $20,000_8$.

3.1.2 CORRECTIONS

APL 1.0 is a new product; therefore, there are no corrections.

3.1.3 DEFICIENCIES

None.

3.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update APL 1.0 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for APL 1.0 are APL1, APL2, and APL3.

APL1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
APL2	Use to collect the APL binary code and generate directive files for later use as input to the procedure file GENSYS.
APL3	Use to verify that the product is correctly installed.

KPL9 contains the APL 1.0 binary and source code.

These materials are described in detail in part I, sections 2.1.1 and 2.3.

3.3 INSTALLATION PARAMETERS

None.

3.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

3.4.1 APL1

```
APL1
APL1,CM54000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF APL 1.0 ON KPL9.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=APL1P0*2P1,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
RETURN,OLDPL.
COMPASS,I,S=0,B=APLTEXT,L=0.
SKIPEI,NEWPL.
COMPASS,I,G=APLTEXT,B=NEWPL,L=0.
SKIPR,COMPILE.
WRITEF,NEWPL.
COMPASS,I,G=APLTEXT,B=OLD,L=0.
LIBEDIT,B=0,L=0.
MAP.
LINK,F=NEW,B=NEWPL.
SKIPEI,NEWPL.
WRITEF,NEWPL.
LIBEDIT,B=0,L=0.
LINK,F=NEW,B=NEWPL.
SKIPEI,NEWPL.
REWIND,OLD.
WRITEF,NEWPL.
COPYBF,OLD,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL9,W,FI=APL1P0*2P1,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL9,V.
*WEOR
*/          PLACE *APL* MODSET HERE.
*WEOR
*D          REL/BAT
*WEOR
*D          REL/TTY
*WEOF
```

3.4.2 APL2

```
APL2
APL2,CM15000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL9,R,FI=APL1P0*2P1,MT,D=HY,F=SI)
SKIPF,KPL9.
COPYBF,KPL9,APL.
COPYBF,KPL9,ZZZADAF,2.
UNLOAD,KPL9.
RETURN,KPL9.
COMMON,ZZZZCCF,ZZZZLDF,ZZZPSBF.
COPYBR,,ZZZZCCF.
COPYBR,,ZZZZLDF.
COMMON,ZZZADAF.
CATALOG,APL,R.
CATALOG,ZZZADAF,R,N=2.
COPYBF,APL,ZZZPSBF.
BKSP,ZZZPSBF.
*WEOR
COMMON,ZZZADAF.
SUI,377776.
PURGE,APLB,APLT/NA.
DEFINE,APLT,APLB/M=R,CT=S.
REWIND,ZZZADAF.
COPYBF,ZZZADAF,APLT.
COPYBF,ZZZADAF,APLB.
RELEASE,ZZZADAF.
*WEOR
*ADD      LIB22,ABS/APLSYS
*WEOF
```

3.4.3 APL3

```
APL3
APL3.
*ACCOUNT CARD.
APL.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
)LOAD EQ
$LP TRIG
$QD $IS C $IS .1 $ML $NG21+$I041
$LP
$LP $NG7  ARCTANH  DOMAIN IS  1$GT$MDB
$TP2 20$RO B,R$IS $NG7$CIB$IS(1$GT$MDC)/C
$LP
$LP         ARCTANH(-Z)=-ARCTANH(Z)
(L$IS$NG7$CI-B)=R$IS -$NG7$CIB
$TP3 20$ROB,L,R
$LP
$LP $NG6  ARCCOSH  DOMAIN IS  1$GE$MDB
$LP $NG6  ARCCOSH  DOMAIN IS  1$GE$MDB
$TP 2 21 $RO B,R $IS $NG6$CI B $IS (1$GE$MDC)/C
```

```
$LP
$LP        ARCCOSH(-Z) = ARCCOSH(Z)
(L $IS $NG6$CI-B)=R $IS $NG6$CIB
$TP 3 21 $RO B,L,R
$LP
$LP  ARCSINH  $NG5
$TP 2 41 $RO B,R $IS $NG5$CI B $IS C
$LP
$LP  ARCSINH(-Z)= -ARCSINH(Z)
(L $IS $NG5$CI-B) =  R $IS-$NG5$CI B
$TP 3 41 $RO B,L,R
$LP
$LP  $NG3 ARCTAN(-Z) = -ARCTAN(Z)
$LP
(L $IS $NG3$CI -B) = R $IS -$NG3$CI B $IS C
$TP 3 41 $RO  B,L,R
$LP
$LP  $NG2 ARCCOS(-Z)= PI-ARCCOS(Z)  DOMAIN IS 1$GE$MD B
(L $IS $NG2$CI-B)=R $IS ($CI1)-$NG2$CI B $IS (1$GE$MDC)/C
$TP 3 21 $RO B,L,R
$LP
$LP  $NG1 ARCSIN(-Z)= -ARCSIN(Z)  DOMAIN IS 1$GE$MDB
(L $IS $NG1$CI -B) = R $IS -$NG1$CI B $IS (1$GE$MDC)/C
$TP 3 21 $RO B,L,R
$LP  1  SIN(-Z) = -SIN(Z)
(L $IS 1$CI -B) = R $IS -1$CI B $IS C
$TP 3 41 $RO B,L,R
$LP
$LP  2  COS(-Z)= COS(Z)
(L $IS 2$CI-B) = R $IS 2$CI B $IS C
$TP 3 41 $RO B,L,R
$LP
$LP  3  TAN(-Z) = -TAN(Z)
(L $IS 3$CI -B) = R $IS 3$CI B $IS C
$LP
$LP  5  SINH(-Z)= -SINH(Z)
(L $IS 5$CI-B) = R $IS-5$CI B $IS C
$TP 3 41 $RO B,L,R
$LP  6  COSH(-Z)= COSH(Z)
(L $IS 6$CI -B) = R $IS 6$CI B $IS C
$LP
$LP  7  TANH(-Z) = -TANH(Z)
(L $IS 7$CI -B) = R $IS - 7$CI B $IS C
)SYSTEM
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.26.29.APL3.
00.26.29.ACCOUNT, USER1.
00.26.29.*ACCOUNT CARD.
00.26.30.APL.
00.26.33.*END OF JOB.
```

APT 4

Available with a later release.

5.1 RELEASE DESCRIPTION

5.1.1 HARDWARE REQUIREMENTS

BASIC 2.1 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length required to run BASIC 2.1 is $30,000_8$.

5.1.2 CORRECTIONS

BASIC 2.1 includes all eligible PSR corrective code for BASIC 2.0 published through KRONOS release level 12.

5.1.3 DEFICIENCIES

None.

5.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update BASIC 2.1 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for BASIC 2.1, the following materials are necessary.

KPL1 contains the KRONOS 2.1 source code.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for BASIC 2.1 are BAS1, BAS2, and BAS3.

BAS1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
BAS2	Use to collect the BASIC binary code and generate directive files for later use as input to the procedure file GENSYS.
BAS3	Use to verify that the product is correctly installed.

KPL11 contains the BASIC 2.1 binary and source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.5.

5.3 INSTALLATION PARAMETERS

The default base for arrays is set to 1. It can be changed to 0 with the following modification set.

```
*IDENT,BSDFLT
*/CHANGE ARRAY BASE DEFAULT
*DELETE BASEST.2
*BDFLT    DATE 0.0    DEFAULT ARRAY BASE
```

In the released version of BASIC 2.1, the unary minus operation is performed following exponentiation. Thus, -2**2=-4. To perform unary minus first, remove the corrective identifier BA30003 with the *YANK directive.

In the released version of BASIC 2.1, a maximum of 15_{10} user files including INPUT and OUTPUT, are allowed. It can be changed with the following modification set.

```
*IDENT NUMFIL
*/CHANGE DEFAULT MAX. NO. OF FILES
*D,BFILMT.1
NUMFILES   EQU   x        NUMBER OF FILES
```

5.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1. The KRONOS 2.1 system program library must be staged (KPL1).

5.4.1 BAS1

```
BAS1
BAS1,CM55000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF BASIC 2.1 ON KPL11.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=BASIC2P1*2P1,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
COMMON,OPL.
RETURN,OLDPL.
COMPASS,I,L=0,X=OPL.
GTR(LGO,LIB)REL/BASOGEN-BASMOPR
LIBGEN,F=LIB,P=BASLIB.
REWIND,BASLIB.
MAP.
LINK,F,B=NEWPL.
SKIPEI,NEWPL.
COPYBF,BASLIB,NEWPL.
REWIND,LIB,LGO.
COPYBF,LIB,NEWPL.
COPYBF,LGO,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL11,W,FI=BASIC2P1*2P1,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL11,V.
*WEOR
*/        PLACE *BASIC* MODSET HERE.
*WEOF
```

5.4.2 BAS2

```
BAS2
BAS2,CM15000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL11,R,FI=BASIC2P1*2P1,MT,D=HY,F=SI)
SKIPF,KPL11.
COPYBF,KPL11,BASIC.
UNLOAD,KPL11.
RETURN,KPL11.
COMMON,ZZZPSBF,ZZZZLDF,ZZZZSDF.
COPYBR,,ZZZZLDF.
COPYBR,,ZZZZSDF.
CATALOG,BASIC,R,U.
COPYBF,BASIC,ZZZPSBF.
BKSP,ZZZPSBF.
*WEOR
*ADD      LIB23,OVL/BASIC-ULIB/BASLIB
*WEOR
*LIBRARY  BASLIB/BASIC
*WEOF
```

5.4.3 BAS3

```
BAS3
BAS3,CM40000,P30,T100.
*ACCOUNT CARD.
*THIS SIMPLE PROGRAM VERIFIES CORRECT
*INSTALLATION OF BASIC 2.1.
BASIC,L.
BASIC,L,B,N.
LGO.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
20  PRINT≡BASIC INSTALLED CORRECTLY≡
30  PRINT
40  PRINT≡THIS DECK USED COMPILE AND EXECUTE MODE≡
100 END
*WEOR
20  PRINT≡BASIC LIBRARY INSTALLED CORRECTLY≡
30  PRINT
40  PRINT≡THIS TEST DECK USED COMPILE TO LGO MODE≡
50  PRINT
60  PRINT≡THANK YOU FOR INSTALLING BASIC 2.1≡
100 END
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.28.17.BAS3,CM40000,P30,T100.
00.28.17.ACCOUNT,USER1.
00.28.17.*ACCOUNT CARD.
00.28.18.*THIS SIMPLE PROGRAM VERIFIES CORRECT
00.28.18.*INSTALLATION OF BASIC 2.1.
00.28.20.BASIC,L.
00.28.20.INPUT    --006000
00.28.22.BASIC,L,B,N.
00.28.22.INPUT    --006000
00.28.23.LGO.
00.28.24.*END OF JOB.
```

COBOL 4.0 6

6.1 RELEASE DESCRIPTION

6.1.1 HARDWARE REQUIREMENTS

COBOL 4.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length necessary to run COBOL 4.0 is 52,000$_8$.

6.1.2 CORRECTIONS

COBOL 4.0 includes all eligible PSR corrective code published through PSR Summary 348.

6.1.3 DEFICIENCIES

Modification CL40304 from PSR Summary 355 must be available in COBOL 4.0 if it is to operate under KRONOS 2.1. This modification is included in the binary code on KPL12, but it is not in the source code on PL9. Therefore, add this modification the first time that the source code is modified and/or updated.

```
*IDENT   CL40304
*/       DFN   APPLIES TO COBOL 4.0
*/             DO NOT CALL D00 FOR KRONOS
*INSERT FEAT33R.254   AT END OF DDTRUBL
         IFNE  OPSYSV,SCOPE34
         SA4   =6LR NBR
         SB2   4
         LX6   48
         MX7   0
         MX1   57
 CNVLPAA BSS   0
         LX6   3          CONVERT MESSAGE NUMBER IN X6 TO DISP CODE
         BX2   -X1*X6
         SX2   X2+33B
         LX7   6
         IX7   X7+X2
         SB2   B2-B1
         NZ    B2,CNVLPAA
         IX7   X7+X4      PUT IN MESSAGE PART
         SA7   D.ETD
         SA2   D.ETD+2    GET FIRST PARAMETER
         SA3   =00055555555555555555
         SB3   54
         BX6   X2
         NZ    X2,SHFLP
         SA3   =XD.BLANK
         BX6   X3
         EQ    STX6
 SHFLP    BSS     0              CHANGE BIN ZERO TO BLANKS
          BX5     -X1*X2
          NZ      X5,DONEXX
          AX2     6
          SB3     B3-6
          EQ      SHFLP
 DONEXX   BSS     0
          AX3     X3,B3
          BX6     X6+X3
 STX6     BSS     0
          SA6     A2
          MESSAGE  ERRMSG,,R
          MX6     0
          SA6     D.ETD+2        CLEAR INSERT
          ELSE
*INSERT FEAT33R.260
          ENDIF
*INSERT FEAT33R.261
          IFNE    OPSYSV,SCOPE34,1
 ERRMSG   DATA    10HCOBOL ERRO
*INSERT FEAT33R.262
          IFNE    OPSYSV,SCOPE34,1
          DATA    10H. PARAM =
          IFEQ    OPSYSV,SCOPE34,1
*C DDTRUBL
*INSERT FEAT33Q.20  CONTROL.1204
          IFEQ    OPSYS,KRONOS,1
 OVLFLG   DATA    1              FORCE SORTL NOT TO BE USED
          IFNE    OPSYS,KRONOS,1
*COMPILE CONTROL
*DELETE ART.1979
          VFD     12/4300B,12/2,36/0
*COMPILE ART
*/    THERE ARE    62 CORRECTION CARDS INCLUDING THIS COMMENT.
```

6.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update COBOL is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for COBOL 4.0, the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for COBOL are CBL1, CBL2, and CBL3.

CBL1 Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.

CBL2 Use to collect the COBOL binary code and generate directive files for later use as input to the procedure file GENSYS.

CBL3 Use to verify that the product is correctly installed.

KPL12 contains the COBOL 4.0 binary code.

PL9 contains the COBOL 4.0 source code.

These materials are described in detail in part I, sections 2.1.1 and 2.6.

6.3 INSTALLATION PARAMETERS

The COBOL compiler uses symbol definitions from IPTEXT for IP.CMU, IP.IMUL, and IP.TYPE. Refer to part III, section 1.7.3 for a list of routines that reference these IPARAMS. To override these installation parameter values, make the following changes in the COMDECK ASSEMOP when COBOL is assembled.

Feature	Required Change
Generate code optimized for a 6600	*D ASSEMOP.3, 4
Generate code optimized for a 6400	*D ASSEMOP.3
Generate integer multiply instruction code	*D ASSEMOP.6, 7
Generate noninteger multiply instruction code	*D ASSEMOP.6
Generate CMU instructions	*D ASSEMOP.9, 10
Generate non-CMU instructions	*D ASSEMOP.9

6.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

6.4.1 CBL1

```
CBL1
CBL1,CM60000,P30,T3500.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF COBOL 4.0 ON KPL12.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=COBOL4P0*3P4,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
RETURN,OLDPL.
COMPASS,I,S=0,B=CETXT,L=0.
COMPASS,I,S=IOTEXT,S=IPTEXT,G=CETXT,B=LIB,L=0.
COMPASS,I,S=IOTEXT,S=IPTEXT,G=CETXT,B=REL,L=0.
SKIPR,COMPILE.
COMPASS,I,S=CPCTEXT,L=0.
REWIND,CETXT,LIB.
COPYBR,CETXT,NEWPL.
MAP.
LINK,F=REL,P=SYSIO,B=NEWPL.
SKIPEI,NEWPL.
LINK,F,P=SYSIO,B=NEWPL.
SKIPEI,NEWPL.
WRITEF,NEWPL.
COPYBF,LIB,NEWPL.
COPYBF,REL,NEWPL.
COPYBF,LGO,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL12,W,FI=COBOL4P0*3P4,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL12,V.
*WEOR
*/       PLACE *COBOL* MODSET HERE.
*WEOF
```

6.4.2 CBL2

```
CBL2
CBL2,CM45000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL12,R,FI=COBOL4P0*3P4,MT,D=HY,F=SI)
SKIPF,KPL12.
COPYBF,KPL12,COBOL.
COPYBF,KPL12,LIB.
UNLOAD,KPL12.
RETURN,KPL12.
CATALOG,COBOL,R.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBF,COBOL,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/COBOL
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R3=1)
GTR(SYSTEM,OLD)ULIB/COBOL
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=COBLIB,N=COBOL.
CATALOG,COBLIB,R,U.
LIBEDIT,P=ZZZPSBF,B=COBLIB,C,L=0.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
*WEOR
*BEFORE   *,REL/*
*WEOR
*BEFORE   *,ULIB/COBOL
*WEOR
*ADD      LIB15,OVL/COBERTX
*ADD      LIB25,OVL/COBOL-COPYCL
*WEOR
*ADD      LIB25,ULIB/COBOL
*WEOF
```

6.4.3 CBL3

```
CBL3
CBL3,CM60000,P30,T100.
*ACCOUNT CARD.
*THIS SIMPLE PROGRAM VERIFIES CORRECT
*INSTALLATION OF COBOL 4.0.
COBOL(LX)
LGO.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. COBOL-VERIFICATION-PROGRAM.
000030 ENVIRONMENT DIVISION.
000040 CONFIGURATION SECTION.
000050 SOURCE-COMPUTER. 6400.
000060 OBJECT-COMPUTER. 6400.
       SPECIAL-NAMES.
           CONSOLE IS SCOP.
000110 DATA DIVISION.
000900 PROCEDURE DIVISION.
000910 START.
           DISPLAY ≠  COBOL 4.0 EXISTS UNDER KRONOS 2.1  ≠ UPON SCOP.
           STOP RUN.
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.33.15.CBL3,CM60000,P30,T100.
00.33.15.ACCOUNT,USER1.
00.33.15.*ACCOUNT CARD.
00.33.16.*THIS SIMPLE PROGRAM VERIFIES CORRECT
00.33.16.*INSTALLATION OF COBOL 4.0.
00.33.16.COBOL(LX)
00.33.17.COMPILING COBOL-V
00.33.20. 000 E AND 000 T/U  DIAGNOSTICS ISSUED
00.33.20.    053000B SCM USED
00.33.20.      .223 CP SECONDS COMPILATION TIME
00.33.20.END COBOL
00.33.20.LGO.
00.33.22.  COBOL 4.0 EXISTS UNDER KRONOS 2.1
00.33.22.*END OF JOB.
```

CYBERLINK INTERCHANGE 1.0 7

7.1 RELEASE DESCRIPTION

Cyberlink Interchange is a subsystem of the KRONOS 2.1 operating system. Cyberlink adds remote computer capability to KRONOS 2.1 and a link to CDC CYBERNET Data Center services. The CYBERNET computers use the SCOPE 3.4 operating system. Both operating systems run on CDC CYBER 70/Models 72, 73, and 74 computers. The remote link provides flexibility both in workload leveling and sharing of hardware, software, and data base resources.

Cyberlink consists of program modules written in COMPASS. It appears as a remote terminal to MARC IV EXPORT when running at a control point within the KRONOS 2.1 operating system.

KRONOS Cyberlink can simulate up to three remote terminals concurrently. For each terminal (or line) supported, two simultaneous data streams (as well as an operator message data path) are possible between the KRONOS and SCOPE operating systems.

7.1.1 HARDWARE REQUIREMENTS

The minimum configuration for Cyberlink is the same as the minimum configuration for KRONOS 2.1 with the addition of one 6673 or 6674 multiplexer coupled on a dedicated I/O channel. The 6673 multiplexer can support at most two communication lines, whereas the 6674 multiplexer contains hardware capable of supporting up to four such lines. QSE 11241 is required for the 6673 or 6674 multiplexer in order to support voice grade (2-9.6 kilohertz) communications.

KRONOS 2.1 connects to the Data Center services SCOPE system through a communication link that consists of two DATAPHONE 303 data sets and one TELEPAK A communication line or their logical and physical equivalent. Voice grade communications consists of two DATAPHONE 203 data sets and one four-wire communications line for their logical and physical equivalent.

7.1.2 CORRECTIONS

None.

7.1.3 DEFICIENCIES

If Cyberlink is connected to the same remote site on more than one line, the line will not be allowed to login because of current job identification and recovery techniques.

It is not known how three active lines may affect operation because Cyberlink has not been actively tested supporting more than two active lines.

Unless absolutely necessary, it is not suggested that the line driver 1CY be operated in synchronous mode when more than one line is active. All lines must wait for the slowest line to complete every communications cycle. This adversely affects efficient operation, with the result that lines may be unnecessarily shut down.

Synchronous communications simultaneous with Cyberlink rollin/rollout activation is not a supported feature.

When KRONOS 2.1 is operating with more than one VALIDUX file (more than one family of permanent files is being concurrently supported), then Cyberlink is unable to properly distinguish jobs submitted by users from different families having the same user index. Therefore, it is possible for users to gain access to jobs that do not actually belong to them.

It is recommended not to run SCOPE 3.4 dependent jobs through Cyberlink because of problems in correctly recovering the job.

7.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update Cyberlink is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for Cyberlink, the following materials are necessary.

KPL2 contains the jobs to modify and update, install, and verify installation; the jobs required for Cyberlink are CYB1, CYB2, and CYB3.

CYB1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
CYB2	Use to collect the Cyberlink binary code and generate directive files for later use as input to the procedure file GENSYS.
CYB3	Use to verify that the product is correctly installed.

KPL13 contains the Cyberlink binary and source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.7.

7.3 INSTALLATION PARAMETERS

The Cyberlink subsystem is designed to allow variation from installation to installation in order to meet the needs of different installations. This variation is possible by changing Cyberlink installation parameters. All installation parameters are in COMSCYB (the common deck that may be listed with the nonexecutable deck CYBDEF). The parameters are catagorized as follows.

	Type of Installation Parameter	Description
1.	Installation parameters that must be changed by all installations	Section 7.3.1
2.	Installation parameters that will probably need to be changed in order to configure Cyberlink to a particular installation's hardware or system operating environment	Section 7.3.2
3.	Installation parameters that affect Cyberlink's external characteristics regarding KRONOS 2.1 (such as timing relays, rollout/rollin characteristics, and B display characteristics)	Section 7.3.3
4.	Installation parameters that affect the internal characteristics of Cyberlink regarding the line driver (such as timeout delays and retransmission counts)	Section 7.3.4

Type 3 and 4 installation parameters allow fine tuning of Cyberlink's total operational characteristics and should be changed only by system analysts who are fully aware of the specific ramifications of any given change. The most important type 3 and 4 parameters are listed and discussed in sections 7.3.2 and 7.3.4; all type 1 and 2 parameters are listed and described in sections 7.3.1 and 7.3.2. A listing of Cyberlink installation parameters is in section 7.3.5.

7.3.1 TYPE 1: INSTALLATION PARAMETERS REQUIRING CHANGE

Every installation using Cyberlink is assigned a unique station identification code. This code consists of four alphanumeric characters preceded by a check character that is used during Cyberlink login to validate the station identification code. The three parameters CHECK, REGION, and UNIQUE define the valid station identification code; they are in COMSCYB. The default vales in COMSCYB for these parameters do not allow Cyberlink to be used until they are modified.

Parameter	Released Default Value	Significance
CHECK	1Rc	One right-justified character (c) that is the valid check character for the station identification code
REGION	2Rxx	Two characters (xx) that define the first two characters of the four-character station identification code
UNIQUE	2Ryy	Two characters (yy) that define the last two characters of the four-character station identification code

7.3.2 TYPE 2: INSTALLATION PARAMETERS THAT MAY BE CHANGED

The following parameters are ones that may be changed to properly configure Cyberlink for a specific installation. They are in the order in which they appear in the common deck COMSCYB. This order does not imply any hierarchy of importance or effect. Many of these parameters are micros that are correctly defined by using the following format.

```
parameter    MICRO    1,,*default value string*
```

The asterisks are delimiters for the default value string and must be included in the micro definition. All parameters of this nature are indicated by the word MICRO preceding the default value.

Parameter	Released Default Value	Significance

IP. LINES	MICRO 0H,1H	IP. LINES defines the ports available on each multiplexer as well as the relative speed of each port. H implies high speed; L implies low speed. The numeric digit for each line indicator can have the value 0, 1, 2, or 3; this designates logically one of four ports for the 6674 multiplexer. Values 2 and 3 are illegal as logical line designators for a 6673 multiplexer because the 6673 multiplexer has only two possible ports. Cyberlink is capable of supporting up to three active lines on a multiplexer, so that up to three numeric digit/line speed character designators may be specified in IP. LINES. Even though three lines, at the most, can be supported, these three lines may be any three ports of a 6674 multiplexer.
		The following example defines port 0 for high speed, port 1 for low speed; it might be used for either a 6673 or 6674 multiplexer.
		`IP. LINES     MICRO     1,,*0H, 1L*`
		The following example defines port 0 for low speed, ports 2 and 3 for high speed; it implies the existence of a 6674 multiplexer.
		`IP. LINES     MICRO     1,,*0L, 2H, 3H*`
		The following example defines port 2 for high speed; it implies the existence of a 6674 multiplexer because it is the third of four logical ports that is defined, even though only one line is indicated.
		`IP. LINES     MICRO     1,,*2H*`

IP. DMUX	MICRO 0	When this parameter is defined, it specifies the logical multiplexer number that is assumed during initialization if the operator uses the default initialization procedure.
IP. DPORT	MICRO 0	If IP. DMUX is defined, IP. DPORT defines the default port (0-3) that is used at initialization if the operator uses default initialization; it must be one of the numeric values contained in IP. LINES.

Parameter	Released Default Value	Significance
IP.DFLT	0	If IP.DFLT is nonzero, Cyberlink bypasses operator requests for the active configuration and assumes the values IP.DMUX and IP.DPORT for defining the port or ports to activate. If zero, the default values are used only if the operator enters END in response to the initialization configuration request, thus indicating that the defaults are acceptable.
IP.DDCM	0	If both IP.DFLT and IP.SYNC are nonzero, then IP.DDCM nonzero causes the creation of a central memory buffer area for the residence of the peripheral processor line driver code while the line driver is bouncing. If IP.DDCM is equal to zero, then this central memory buffer is not provided and the line driver must be reloaded from its library residence, whether in central memory or on disk, during each bounce. It is desirable to provide this CM buffer for the line driver if it is necessary for the line driver to bounce, because a degradation in overall performance will occur otherwise.

7.3.3 TYPE 3: INSTALLATION PARAMETERS THAT AFFECT EXTERNAL CYBERLINK CHARACTERISTICS

The type 3 parameters that are listed below are in the order in which they appear in the common deck COMSCYB; no specific relative order is implied.

Parameter	Released Default Value	Significance
IP.RJBS	25D	IP.RJBS defines the maximum number of jobs allowed at any remote site at any one time. This parameter affects the static field length of Cyberlink while running, as well as the job recovery file format.
IP.CLEV	20D	This parameter defines the maximum number of minutes that a lost job is held for resubmission and possible recovery.
IP.LCEV	39D*60D	IP.LCEV defines the maximum number of minutes that a returned job is held in the user access lock queue (that is, file type LCFT) before being evicted. The default value is 39 hours.

Parameter	Released Default Value	Significance
IP.NOACT	1*60D+30D	IP.NOACT defines the number of seconds Cyberlink must remain idle before it is rolled out. If IP.NOACT=0, Cyberlink rollout is inhibited. The default value is 1.5 minutes.
IP.RWAIT	3*60D	IP.RWAIT defines the number of seconds Cyberlink will remain rolled out when jobs are currently at the remote station. Jobs are prevented from returning while Cyberlink is rolled out. The default value is 3 minutes.
IP.RSTAY	5*60D	IP.RSTAY defines the number of seconds Cyberlink will remain rolled out when no jobs are currently at a remote station. The default value is 5 minutes.
IP.ISTAY	4*60D	IP.ISTAY defines the number of seconds Cyberlink will remain active after login before rollout may be resumed. The default value is 4 minutes.
IP.MHOLD	3	IP.MHOLD defines the maximum number of seconds to allow a line message to flash on the B display before automatically clearing it (AUTOCLEAR function). If IP.MHOLD=0, the AUTOCLEAR function is disabled. The value of IP.MHOLD may be effectively dynamically changed by means of a console command during Cyberlink operation. The default value is 3 seconds.
IP.SYNC	0	If IP.SYNC is nonzero, then Cyberlink's line driver (program deck 1CY) is assembled so that all communication lines are synchronized with regard to their logical cyclical operation. This implies that the basic functional cycle TRANSMIT-RECEIVE-PROCESS must be actually completed by all lines currently active before the next similar cycle may begin for any line. This, in fact, causes all lines to always wait for the the slowest line to complete its basic functional cycle before continuing. In most cases, synchronous operation of the line driver is not suggested. The one advantage that synchronous operations does allow is that the peripheral processor in which 1CY is executing may be released between each cycle for a brief

Parameter	Released Default Value	Significance
		period of time to allow other system routines to execute. However, considerable degradation of throughput on all lines results from the line driver bouncing in this fashion. If IP.SYNC is equal to zero, the line driver is assembled so that all communication lines operate asynchronously. This effectively allows all lines to communicate at maximum speed, because no line need wait on any other as described for synchronous operation. However, the peripheral processor in which 1CY is executing must be dedicated, and hence, is unavailable for any other system task. Cyberlink defaults to asynchronous operation in order to maximize communication effectiveness.

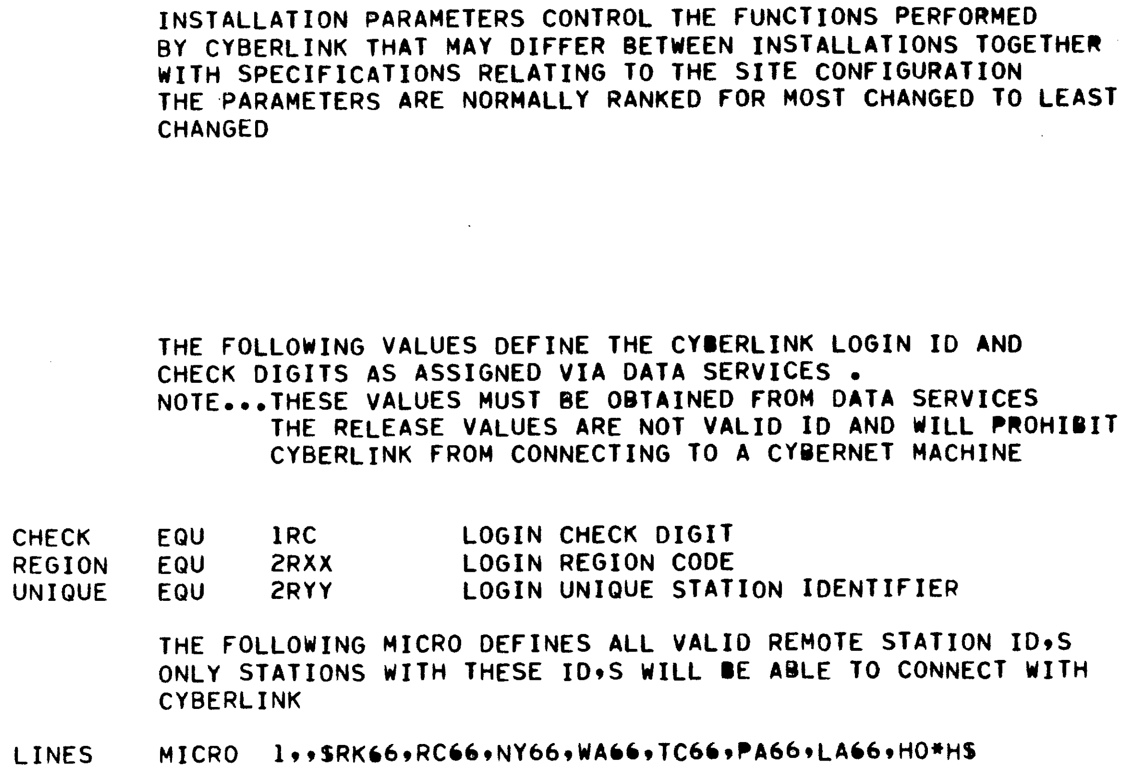
7.3.4 TYPE 4: INSTALLATION PARAMETERS THAT AFFECT INTERNAL CYBERLINK CHARACTERISTICS

The type 4 installation parameters discussed below affect the internal timing of the Cyberlink line driver only. They determine the amount of wait time and corresponding number of retransmission retrys before a communication link is considered inoperable. Conditions that cause a timeout and subsequent shutdown of a given line may arise from many different causes. These range from hardware failures at the local site (for example, in the 6673 or 6674 multiplexer) to transmission errors introduced by the leased communication lines to system failures at a remote installation. Whatever the cause, Cyberlink attempts a graceful disconnection and cleanup in addition to informative messages to the B display and system dayfile. Recovery is generally possible by means of a reactivation of the communication link which has just been broken. All of the following internal timing parameters are reinitialized after any successful communication so that momentary, but recoverable, errors do not cause unnecessary abnormal shutdown at a later time (that is, errors that are not fatal are not allowed to propagate through time).

Parameter	Released Default Value	Significance
TIMLIMIT	10D	TIMLIMIT defines the effective wait time while Cyberlink is expecting to receive from the remote site before communications with that site are considered disrupted. The actual period of time involved depends upon the number of lines being supported by Cyberlink; because each line is serviced in round-robin fashion, more active lines imply a longer wait period. The receive-first-byte code must be entered unsuccessfully 4096 times for each decrement of TIMLIMIT. If the value of TIMLIMIT reaches zero, communications are considered to be disrupted and Cyberlink leaves its receive code (for the line in question) and goes into a transmission cycle in an attempt to recover synchronization with the remote site.

Parameter	Released Default Value	Significance
FRTCONT	2	FRTCONT defines the number of times that retransmission is attempted (following a previous timeout on TIMLIMIT) before communication with a remote site is considered lost.
SEQERCNT	60D	If a sequence error occurs (that is, the sequence bit as sent by the remote site is not as expected), a transmission error is indicated and Cyberlink attempts to recover by retransmitting its previous data. SEQERCNT defines the number of attempted retransmits before communication is considered lost.
ERRLIMIT	20D	ERRLIMIT defines the number of attempted retransmissions following any error other than receive timeout or sequence bit errors before communication with a remote site is considered lost.

7.3.5 INSTALLATION PARAMETER LISTING



```
                  INSTALLATION PARAMETERS CONTROL THE FUNCTIONS PERFORMED
                  BY CYBERLINK THAT MAY DIFFER BETWEEN INSTALLATIONS TOGETHER
                  WITH SPECIFICATIONS RELATING TO THE SITE CONFIGURATION
                  THE PARAMETERS ARE NORMALLY RANKED FOR MOST CHANGED TO LEAST
                  CHANGED




                  THE FOLLOWING VALUES DEFINE THE CYBERLINK LOGIN ID AND
                  CHECK DIGITS AS ASSIGNED VIA DATA SERVICES .
                  NOTE...THESE VALUES MUST BE OBTAINED FROM DATA SERVICES
                         THE RELEASE VALUES ARE NOT VALID ID AND WILL PROHIBIT
                         CYBERLINK FROM CONNECTING TO A CYBERNET MACHINE

         CHECK    EQU    1RC         LOGIN CHECK DIGIT
         REGION   EQU    2RXX        LOGIN REGION CODE
         UNIQUE   EQU    2RYY        LOGIN UNIQUE STATION IDENTIFIER

                  THE FOLLOWING MICRO DEFINES ALL VALID REMOTE STATION ID,S
                  ONLY STATIONS WITH THESE ID,S WILL BE ABLE TO CONNECT WITH
                  CYBERLINK

         LINES    MICRO  1,,$RK66,RC66,NY66,WA66,TC66,PA66,LA66,HO*H$
         THE FOLLOWING DEFINES THE CONFIGURATION OF PORTS ON THE
         6673/4 MUX TOGETHER WITH THE DEFAULTS (IF ANY) FOR PORT
         CONFIGURATION

         IP.MUXES DEFINES THE LOGICAL MUX NUMBER(S) FOR THE MUX(ES)
                PRESENT IN THIS INSTALLATION
         IP.LINES DEFINES THE PORTS AVAILABLE AS WELL AS THE SPEED OF
                EACH (I.E., HIGH SPEED OR LOWER SPEED).  EACH PORT
                IS DESIGNATED BY ITS LOGICAL NUMBER (0-3) FOLLOWED BY
                THE CHARACTER -H-, IF HIGH SPEED, OR -L-, IF LOWER
                SPEED.  THE NUMBER OF PORTS DEFINED AFFECTS THE FIELD
                LENGTH REQUIRED.
                NOTE.  CURRENTLY, -1CY- ALLOWS AT MOST THREE LINES
         IP.DMUX-IF DEFINED-AT ALL INDICATES THE MUX NUMBER (CURRENTLY
                0 ONLY) THAT WILL BECOME THE DEFAULT MUX IF NO OPERATOR
                CHANGE IS MADE I.E-END-ENTERED FOR CONFIGURATION
                DISCRIPTION AT INITIALIZATION
         IP.DPORT DEFINES THE PORTS TO BE USED ON MUX IP.DMUX SHOULD
                THE OPERATOR CHOOSE DEFAULTS AT INITIALIZATION.
                IP.PORT CAN BE ANY VALUE OR VALUES DEFINED IN IP.LINES
         IP.DFLT-IF DEFINED ≠ 0 WILL BYPASS OPERATOR REQUEST FOR
                CONFIGURATION AND WILL ASSUME THE DEFAULTS IP.DMUX AND
                IP.DPORT - IF DEFINED = 0  THE DEFAULTS IP.DMUX AND
                IP.DPORT ARE USED ONLY IF THE OPERATOR ENTERS THE INPUT
                -END- TO THE CONFIGURATION REQUEST AT INITIALIZATION
         IP.DDCM DEFINES THE DEFAULT DRIVER RESIDENCE TO BE USED ONLY
                IF IP.DFLT ≠ 0.  IP.DDCM = 0 INDICATES THAT THE
                DEFAULT RESIDENCE OF THE LINE DRIVER IS NOT CM BUFFERED
                IP.DDCM ≠ 0 INDICATES TO CREATE A BUFFER FOR THE
                LINE DRIVER RESIDENCE
                NOTE -- THIS PARAMETER ONLY HAS MEANING IF THE DRIVER
                IS ASSEMBLED IN SYNC MODE AND IP.DFLT ≠ 0

IP.MUXES MICRO  1,,*0*        DEFINE LOGICAL MUX 0 PRESENT

IP.LINES MICRO  1,,*0H,1H*

IP.DMUX  MICRO  1,,*0*        DEFINE DEFAULT TO MUX 0

IP.DPORT MICRO  1,,*0*        DEFINE DEFAULT TO PORT 0

IP.DFLT  EQU    0             ASK OPERATOR FOR CONFIGURATION CHANGES

IP.DDCM  EQU    0             DEFAULT DRIVER RESIDENCE TO NOT CM BUFFERED


         IP.RJBS DEFINE THE MAXIMUM NUMBER OF JOBS ALLOWED AT ANY
                REMOTE SITE AT ANY ONE TIME - THIS PARAMETER CONTROLS
                THE SIZE OF THE JOB TABLE AND RECOVERY BLOCK -IF
                CHANGED THE CYBERLINK FIELDLENGTH WILL ALSO CHANGE
                AND ANY RECOVERY FILE MUST BE REINITIALIZED
```

```
IP.RJBS  EQU    25D           ALLOW 25 JOBS AT REMOTE STATION

          THE FOLLOWING VALUES SET THE AGE TIMES FOR CYBERLINK FILES
          IP.AGECL DEFINES THE TIME BETWEEN AGEING OF CYBERLINK FILES
                 (MINUTES)
          IP.CLEV DEFINES THE NUMBER OF MINUTES A FILE WILL WAIT FOR
                 OPERATOR RE-SUBMISSION AFTER A LOST IN CYBERNET IS
                 INDICATED TO THE OPERATOR
          IP.LCEV DEFINES THE NUMBER OF MINUTES A FILE RETURNED FROM
                 CYBERNET WILL WAIT IN QUEUE BEFORE BEING EVICTED FROM
                 THE SYSTEM.  THE USER MUST RETRIEVE THE FILE BEFORE THE
                 TIME HAS ELAPSED
          ALL VALUES MUST BE LESS THAN 4096

IP.AGECL EQU    2             AGE EVERY 2 MINUTES

IP.CLEV  EQU    20D           HOLD JOB LOST 20 MINUTES FOR RE-SUBMITS

IP.LCEV  EQU    39D*60D       HOLD USER RETURNED FILES FOR 39 HOURS ONLY

          THE FOLLOWING VALUES SET THE EXISTENCE OF CYBERLINK ROLLOUT
          AND THE TIMES FOR ROLLIN/ROLLOUT
          IP.NOACT IF ≠ 0 DEFINES THAT ROLLOUT EXISTS AND SETS THE
                 NUMBER OF SECONDS AFTER ALL ACTIVITY CEASES BEFORE '
                 ROLLOUT IS TRIGGERED - IF = 0 NO ROLLOUT CODE ASSEMBLES
          IP.RWAIT DEFINES THE NUMBER OF SECONDS CYBERLINK WILL REMAIN
                 ROLLED OUT WHEN THERE ARE JOBS AT THE REMOTE STATIONS
          IP.RSTAY DEFINES THE NUMBER OF SECONDS CYBERLINK WILL REMAIN
                 ROLLED OUT WHEN NO JOBS ARE AT THE REMOTE STATIONS
          IP.ISTAY DEFINES THE NUMBER OF SECONDS CYBERLINK WILL REMAIN
                 ROLLED IN AFTER A LOGIN HAS OCCURED AFTER INITIALIZATIO

IP.NOACT EQU    1*60D+30D     WAIT 1.5 MIN BEFORE ROLLOUT

IP.RWAIT EQU    3*60D         WAIT 3 MIN BEFORE ROLLIN (JOBS AT REMOTE)

IP.RSTAY EQU    5*60D         WAIT 5 MIN BEFORE ROLLIN (NO JOBS AT REM)

IP.ISTAY EQU    4*60D         WAIT 4 MIN BEFORE ROLLOUT (INI LOGIN)

          THE FOLLOWING DEFINES CLOCK VALUES FOR EVENTS
          IP.CLOG DEFINES THE MAX SECONDS TO WAIT FOR A LOGIN OR
                 SHUTDOWN BEFORE ASSUMMING A LOGON
          IP.JSTAT DEFINES THE NUMBER OF MINUTES BETWEEN INTERNAL
                 JOB STATUS REQUESTS ARE ISSUED TO ASSURE JOB STILL AT
                 REMOTE STATION
          IP.BDREF DEFINES NUMBER OF SECONDS BEFORE UPDATING B DISPLAY
          IP.KDREF DEFINES NUMBER OF SECONDS BEFORE UPDATING DYNAMIC
                 K DISPLAYS
          IP.FNSRC DEFINES NUMBER OF SECONDS BETWEEN FNT SEARCHES FOR
          TRANSMIT FILES
     IP.CPRES DEFINES NUMBER OF SECONDS BEFORE UNCONDITIONAL
          RESTART OF CP IS REQUESTED
     IP.MHOLD DEFINES NUMBER OF SECONDS BEFORE AN AUTO ACKNOWLEDGE
          OF NON ERROR MESSAGES WILL BE PERFORMED IF = 0 ALL
          MESSAGES REQUIRE A MANUAL ACKNOWLEDGE

IP.CLOG  EQU    2*60D         WAIT 2 MIN FOR LOGON MESSAGE

IP.JSTAT EQU    20D           STAT JOBS AT REMOTE EVERY 20 MINUTES

IP.BDREF EQU    4             REFRESH B DISPLAY EVERY 4 SECONDS

IP.KDREF EQU    4             REFRESH K DISPLAY EVERY 4 SECONDS

IP.FNSRC EQU    10D           SEARCH FOR TRANSMIT FILES EVERY 10 SECONDS

IP.CPRES EQU    4             UNCODITIONAL RESTART OF CP EVERY 4 SECONDS

IP.MHOLD EQU    3             SET AUTO ACKNOWLEDGE TO EVERY 3 SECONDS


     THE FOLLOWING DEFINES FEATURE PARAMETERS TO ACTIVATE/
          DEACTIVATE CERTAIN FEATURES OR SET INITIAL FEATURES
     IP.TYIN DEFINES THE SUPPORT OF INPUT FILES VIA CYBERLINK
          AND THE FILE TYPE THAT CYBERLINK WILL CONSIDER AS
          AN INPUT FILE - THIS TYPE IS THEN PLACED INTO THE
          KRONOS INPUT QUEUE AND PROCESSED AS A BATCH JOB.
          THE VALUES FOR IP.TYIN ARE-
          0 NO INPUT FILES SUPPORTED
          1 PUNCH  WILL ENTER INPUT
          2 PUNCHB WILL ENTER INPUT
          3 P80    WILL ENTER INPUT
          4 FILMPR WILL ENTER INPUT
          5 FILMPL WILL ENTER INPUT
          6 PLOT   WILL ENTER INPUT
          7 APTPT  WILL ENTER INPUT
     IP.TYPF DEFINES SUPPORT OF RECEIVING PERMANENT FILES
          USER FILES RECEIVED WITH THIS DISPOSITION ARE PLACED AS
          DIRECT ACCESS FILES UNDER THE USERS USER INDEX- IF
          THIS TYPE FILE IS RECEIVED IT WILL ENTER THE INPUT
          QUEUE AS A SYSTEM CLASS JOB AND USEING THE PROGRAM
          -PFLINK-THE FILE WILL BE CREATED OR APPENDED AS A
          DIRECT ACCESS FILE UNDER THE CORRECT USER INDEX- THE
          VALUE OF IP.TYPF CAN BE ANY VALUE WHICH COULD BE USED
          FOR IP.TYIN HOWEVER CANNOT BE THE SAME VALUE USED
          TO DEFINE IP.TYIN EXCEPT 0 IN WHICH CASE THE FEATURE
          IS NOT SUPPORTED- IF IP.TYPF IS DEFINED AS 0 THE
          -ALL- PERMANENT DISPOSITION ACCEPTIABLE ON THE SUBMIT
          CARD -SUBMIT(LFN,H,P)- WILL ALSO NOT BE SUPPORTED
     IP.PSTYP IF = 0 THE INITIAL B DISPLAY WILL BE SET AT LINE
          MODE DISPLAY IF = 1 THE INITIAL B DISPLAY WILL BE SET
          TO LINE ACTIVITY DISPLAY
          IP.KDSP IF = 0 THE CYBERLINK K DISPLAYS WILL NOT BE SUPPORTED
                  IF ≠ 0 DEFINES THE INITIAL DISPLAY SET AFTER
                  INITIALIZATION THE VALUES ARE
                 1 COMMAND DISPLAY
                 2 DAYFILE PAGE DISPLAY
                 3 LINE FILES DISPLAY
                 4 QUEUE DISPLAY
          IP.CMSG DEFINES THE MESSAGES WHICH WILL BE SENT TO THE
                 DAYFILE FOR AUDIT TRAIL CONTROL
                 0 NO LINE AUDIT MESSAGES
                 1 LINE MESSAGES
                 2 ALL MESSAGES BUT CYBERLINK CONSOLE ENTRIES
                 3 ALL MESSAGES INCLUDING CONSOLE ENTRIES
          IP.MSCLR IF ≠ 0 INDICATES THAT ANY OPERATOR CYBERLINK CONSOLE
                 MESSAGE ENTERED WILL ALSO ACT AS A MANUAL ACKNOWLEDGE
                 IF ANY MESSAGE REQUIRES AN ACKNOWLEDGE -IF = 0 ALL
                 CONSOLE INPUTS WILL HAVE NO EFFECT TOWARD AN ACKNOW-
                 LEDGEMENT
          IP.MSAGE DEFINES THE MAXIMUM STACK PRIORITY A WARNING MESSAGE
                 MAY OBTAIN WHILE QUEUED BEFORE BEING FORCED TO THE
                 B DISPLAY
          IP.DSIZE DEFINES THE MAXIMUM NUMBER OF LINES DISPLAYED FOR
                 A GIVEN K DISPLAY PAGE SIZE.
          IP.FHDR IF ≠ 0 THE ONE RECORD HEADER SENT FOR ALL FILES AT
                 THE START OF A FILE WILL NOT BE WRITTEN ON DISK
                 HENCE, RECEIVED FILES CONTAIN ONLY WHAT THE USER
                 PLACED THERE WITH NO ADDED RECORDS.
          IP.SYNC IF ≠ 0 DEFINES ASSEMBLY FOR SYNCHRONOUS COMMUNICATION
                  IF = 0 ASSEMBLY WILL BE ASYNCHRONOUS COMMUNICATIONS
          IP.BONCE ONLY USED IF IP.SYNC ≠ 0 - IF = 0 THE INITIAL STATE
                 OF THE SYNCHRONOUS DRIVER WILL BE DEDICATED - IF ≠ 0
                 THE INITIAL STATE WILL BE UNDEDICATED
          IP.CIO THIS PARAMETER IS ONLY USED IF RUNNING CYBERLINK ON
                 KRONOS 2.0  IF DEFINED ≠ 0 CIO WILL BE USED FOR ALL
                 DISK I/O - IF DEFINED = 0 THE I/O WILL BE PROCESSED
                 VIA THE STACK PROCESSOR 1SL

IP.TYIN  EQU    4            DEFINE FILMPR FOR INPUT FILE

IP.TYPF  EQU    0            DEACTIV PERMANENT FILES=5 WHEN ACTIVE

IP.DSTYP EQU    1            SET INITIAL B DISPLAY TO LINE ACTIVITY

IP.KDSP  EQU    1            SET K DISPLAY INITIALY TO COMMAND DISPLAY

IP.CMSG  EQU    2            SEND ALL BUT CONSOL ENTRIES TO DAYFILE

IP.MSCLR EQU    1            ANY CONSOL INPUT WILL ACT AS ACKNOWLEDGE

IP.MSAGE EQU    2000B        FORCE WARNING TO DISPLAY AT PRIORITY 2000

IP.DSIZE EQU    35D          35 LINES/PAGE ON K DISPLAY
```

```
                CYCLIC ERRORS
         SEQERCNT MAXIMUM NUMBER OF RETRANSMITS ON SEQUENCE ERRORS

         FRTCONT MAXIMUM NUMBER OF RETRANSMITS ON TIMLIMIT EXCEEDED
                ERRORS


TIMLIMIT EQU    10D          WAIT APROX 25 SECONDS FOR INPUT

ERRLIMIT EQU    20D          RETRANS 20 TIMES ON NORMAL ERRORS

SEQERCNT EQU    60D          RETRANS 60 TIMES ON SEQUENCE ERRORS

FRTCONT  EQU    2            RETRANS 2 TIMES ON WAIT INPUT TIMEOUT


         THE FOLLOWING DEFINE THE DEBUG AND TRACE CONTROL FOR
         COMMUNICATION AND SOFTWARE CHECKOUT
         DEBUG  IF ≠ 0 WILL TRIGGER THE DEBUG CODE
                IF = 1 CP DEBUG CODE IS ACTIVATED
                IF .GT. 1 BOTH CP AND PP CODE IS ACTIVATED
         DELAY  IS USED ONLY IF DEBUG IS .GT. 1 AND DEFINES THE
                NUMBER OF MICRO SECONDS TO DELAY NON EXISTANT LINES
                IN THE LINE DRIVER.  THIS SETS SIMULATED TIMING OF
                MULTI LINES
         TRACE  IF -DEFINED- A TRACE OF AN ACTIVE LINE - THE TRACE IS
                WRITTEN TO THE PP MESSAGE BUFFER IN A CIRCULAR FASHION
                (SEE ROUTINE -ERRTRC- IN 1CY FOR TRACED VALUES.
                NOTE- ONLY ONE LINE SHOULD BE ACTIVE WHEN TRACE IS D
                DEFINED FOR TIMING REASONS AND READABILITY


DEBUG    EQU    0            DEACTIVATE DEBUG

DELAY    EQU    0            SET NO SIMULATED DELAY WHEN DEBUG ON

(TRACE   EQU    1)           NO TRACE IN DRIVER-NOT DEFINED-
```

```
IP.FHDR  EQU    1           STRIP ALL EXPORT HEADERS FROM USERS OUTPUT

IP.SYNC  EQU    0           ASSEMBLE ASYNCHRONOUS COMMUNICATIONS

IP.BONCE EQU    0           SET INITIAL SYNCHRONOUS COMMUNICATION DEDIC

IP.CIO   EQU    1           IF KRONOS VERSION 2.0 SET CIO FOR I/O


         THE FOLLOWING CONTROL THE BUFFER SIZES AND ALLOCATION OF
         IO BUFFERS
         IP.BPRU IS THE NUMBER OF PRU,S ALLOCATED FOR EACH I/O
               BUFFER ACTIVE
         IP.BMIN IS THE MINIMUM NUMBER OF BUFFERS HELD AVALIABLE IN
               THE POOL AT ALL TIMES


IP.BPRU  EQU    6           REQUEST 6 PRU,S FOR I/O BUFFER

IP.BMIN  EQU    0           DO NOT HOLD ANY BUFFERS IN RESERVE


         THE FOLLOWING PARAMETERS CONTAIN PARAMETERS WHICH ARE
         NORMALLY ONLY DEVELOPMENT INSTALLATION CONTROLLED
         IP.VSNO DEFINES THE CYBERLINK VERSION NUMBER
         KRONVS DEFINES THE KRONOS VERSION NUMBER - 0 FOR KRONOS 2.0
               - 1 FOR KRONOS 2.1
         CONFIG DEFINES TO EXPORT THE LOGIN CONFIGURATION
            BIT 0 CARD READER
            BIT 1 LINE PRINTER
            BIT 2 CARD PUNCH
            BIT 3 MAG TAPE
            BIT 4 2ND LINE PRINTER
            BIT 5 DIVERT PUNCH TO REMOTE STATION
            BIT 6 SEND HEADERS FOR ALL FILES
            BIT 10 CYBERLINK COMMUNICATION STATION
            BIT 11 COMPRESS/DECOMPRESS DATA
         WRTRIG DEFINES THE NUMBER OF WORDS THAT MUST BE IN BUFFER
               BEFORE BUFFER IS DUMPED
         RDTRIG DEFINES MINIMUM NUMBER OF WORDS IN BUFFER BEFORE
               TRIGGERING MORE INPUT
         PRUSIZE DEFINES NUMBER OF WORDS/PRU ON MASS STORAGE
         IP.BSIZE DEFINES THE NUMBER WORDS PER I/O BUFFER


IP.VSNO  MICRO  1,,*V1.0*

KRONVS   EQU    1           SET ASSEMBLY FOR KRONOS 2.1

CONFIG   EQU    2107B       SET FULL CYBERLINK CONFIGURATION

WRTRIG   EQU    200B        DUMP BUFFERS WHEN 200 WORDS IN BUFFER
```

```
RDTRIG   EQU    300B         READ DATA WHEN ONLY 300 WORDS LEFT IN BUFFE

PRUSIZE  EQU    100B         I/O PRU SIZE

IP.BSIZE EQU    IP.BPRU*PRUSIZE

         THE FOLLOWING DEFINE CONTROL PARAMETERS -JOB ID,PRIORITIES
                ETC FOR FILES DISPOSED DIRECTLY TO DIRECT ACCESS
                PERMANENT FILES-SUPPORTED ONLY IF IP.TYPF ≠ 0.
                IF -PFID-=0 NO SUBMIT TIME PERMANENT FILES ARE
                SUPPORTED AT ALL ONLY DISPOSITION TYPE FILES WILL BE
                SUPPORTED

PFEPR    EQU    30B          CPU PEIORITY OF PF INPUT JOB
PFFL     EQU    2500B/100B   INITIAL FL OF PF INPUT JOB
PFQPR    EQU    7760B        QUEUE PRI OF PF INPUT JOB

         THE FOLLOWING CONTROLS THE PERMANENT AND LOCAL FILE NAMES
         AND DEVICE FOR THE RECOVERY FILE
         IF THE MICRO PFDEVICE IS DEFINED IT DEFINES THE DEVICE THE
         RECOVERY FILE WILL BE PLACED ON IF NOT DEFINED THEN ANY
         DIRECT ACCESS FILE DEVICE WILL BE USED


INDLFN   MICRO  1,,*ALTLFN* ALTERNAT RECOVERY LFN

INDPFN   MICRO  1,,*ALTPFN* ALTERNATE RECOVERY PFN

CHKLFN   MICRO  1,,*PRILFN* PRIMARY RECOVERY LFN

CHKPFN   MICRO  1,,*PRIPFN* PRIMARY RECOVERY PFN

CHKPSWD  MICRO  1,,*CLINK*  RECOVERY PASSWORD

         THE FOLLOWING DEFINES THE ROLLOUT FILE NAME


ROLLLFN  MICRO  1,5,*CROLL* ROLLOUT FILE LFN

         THE FOLLOWING DEFINES PARAMETERS TO CONTROL THE LINE DRIVER
         RETRANSMISSIONS AND RECOVERY ATTEMPTS
         TIMLIMIT DEFINES THE MAXIMUM NUMBER OF WAITLOOPS BEFORE THE
                TIME COUNT ON WAITING FOR FIRST BYTE OVER THE LINE
                IS ADVANCED-EACH TIME LIMIT IS EQUIVALANT TO 4096
                ENTRIES INTO THE INPUT ROUTINE-IF TIMLIMIT IS EXCEEDED
                A RE-TRANSMIT IS ATTEMPTED

         ERRLIMIT MAXIMUM NUMBER OF RETRANSMITS ON MUX AND DATA AND
```

7.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

7.4.1 CYB1

```
CYB1
CYB1,CM54000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF CYBERLINK 1.0 ON KPL13.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL13,R,FI=CYBERLINK1P0*2P1,MT,D=HY,F=I)
COPYBF,KPL13,CYBPL.
UNLOAD,KPL13.
*GET,MOD=CYBMOD.
COMMON,OPL.
RETURN,KPL13.
MODIFY,N,LO=E.
SKIPEI,NPL.
COMPASS,I,B=NPL,L=0,X=OPL.
WRITEF,NPL.
CATALOG,NPL,R.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL13,W,FI=CYBERLINK1P0*2P1,MT,D=HY,F=I)
COPYEI,NPL,KPL13,V.
*WEOR
*OPLFILE  CYBPL
*READ     MOD,*
OD,*
*EDIT     COMSCYB.QFETCH
OMSCYB.QFETCH
*WEOF
```

7.4.2 CYB2

```
CYB2
CYB2,CM15000,P30,T1000.
*ACCOUNT CARD.
*CYBERLINK 1.0 BINARIES PROVIDED ON
*RELEASE TAPE KPL13 CANNOT BE USED,
*SINCE A VALID LOCAL STATION ID MUST BE
*OBTAINED FROM DATA SERVICES DIVISION
*AND ASSEMBLED INTO CYBERLINK VIA CYB1.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL13,R,FI=CYBERLINK1P0*2P1,MT,D=HY,F=I)
SKIPF,KPL13.
COMMON,ZZZPSBF,ZZZZLDF.
SKIPR,KPL13.
COPYBF,KPL13,ZZZPSBF.
RETURN,KPL13.
BKSP,ZZZPSBF.
COPYBR,,ZZZZLDF.
*WEOR
*ADD      LIB33,PP/1CY-ABS/QFETCH
*WEOF
```

7.4.3 CYB3

```
CYB3
CYB3,CM5000,P30.
*ACCOUNT CARD.
COPYBF,,TRANS.
SUBMIT(TRANS,H,B)
*WEOR
*$CHARGE CARD.
CYBT,P10.
FTN(G)
*WEOR
      PROGRAM EX1A(INPUT,OUTPUT)
      READ 1,INVAL
      PRINT 2,INVAL
1     FORMAT(I2)
2     FORMAT(* COMPILATION AND EXECUTION AT REMOTE STATION*/
     1 10X,* VALUE READ IS *,I3)
 10X,* VALUE READ IS *,I3)
      END
*WEOR
99
*WEOF
```

7.5 ADDITIONAL INFORMATION

7.5.1 CYBERLINK LIBRARY

The Cyberlink subsystem as released on the Modify library consists of several peripheral processor (PPU) and central processor (CPU) program decks. One common deck is included, as well as one deck that provides a listing of the common deck. A brief description of the Cyberlink library deck follows; two of the decks (1CY and 1SL) consist of several overlays, all of which are noted.

Deck — Significance

COMSCYB — Cyberlink common deck.

CYBDEF — Nonexecutable program included for the purpose of obtaining a listing of the common deck COMSCYB; this listing is obtained by assembling CYBDEF.

1CY — Primary communication routines that interface with a 6673 or 6674 multiplexer over a dedicated I/O channel; PPU program. The following is a complete list of overlays assembled by COMPASS from program deck 1CY.

- 1CY — Line driver basic routines (for example, initialization and restart code).
- 2CY — Line driver channel routines
- 5CY — Directive/status processor (for example, routines necessary to encode/decode communication protocol)
- 2DC — Routines necessary to generate Cyberlink statistics and drop the line driver from its peripheral processor.
- 1CS — Routines required to load the line driver from its central memory buffer; assembles only if the driver is designated as operating in a synchronous mode (refer to section 7.3.3 description of IP.SYNC) and with a central memory buffer provided for the driver in Cyberlink's field length.

Deck	Significance
1SL	System interface executive and stack processing routines; PPU program deck. The following is a complete list of overlays assembled by COMPASS from program deck 1SL.

1SL	Main control routines
4JL	Job lost processor
3SR	Buffer allocation processor
3IM	Routines necessary to process executive messages
3IL	Routines necessary to idle down an abnormally terminated line
3CM	Routines necessary to process external commands
3FN	Find file processor
2AR	Verify I/O parameters
2OP	Open transmit file
2CL	Close, close/unload, and evict files processor
3EV	Routines to process the PURGEFILE and RE-SUBMIT commands
2JE	Job card error processor
3RL	Routines necessary to recover Cyberlink files
3CK	Checkpointing routines
3D1	Routines to initiate command display
3D2	Routines to initiate dayfile display
3D3	Routines to initiate files display
3D4	Routines to initiate queues display

Deck	Significance
INH	Cyberlink initialization routines; PPU program deck
CFM	Support software function manager that is used by routines QSTATUS and QFETCH; PPU program deck
CYBLINK	Cyberlink central processor executive; CPU program deck
QSTATUS	QSTATUS command processor; returns to the user a report on job files (as well as any returned output files) submitted to Cyberlink; CPU program deck
CLINKPF	Places received files into user permanent file catalog; CPU program deck
QFETCH	QFETCH command processor; allows the user to access returned output files, specifically those files originating from a teletypewriter with file type LCFT (locked to all except originating user); CPU program deck

Cyberlink may also call the following KRONOS system routines.

0BF	Create FTN/FST entry for a file
2TJ	Verify job card parameters
CIO	System input/output processor

The programs 1SL and 1CS should reside in the central memory library as they are recalled on a periodic recall. This is accomplished by adding the correct SYSEDIT directive (*CM) to the LIBDECK when building the deadstart tape. Refer to part III, section 1.1.5.

7.5.2 NONDEDICATED OPERATION

The operation of Cyberlink in nondedicated mode introduces critical time response problems. MARCIV EXPORT requires a response maximum in the order of 3 seconds. With Cyberlink in nondedicated mode, the line driver releases the peripheral processor after completion of a communications cycle for all active lines and is recalled at a period of AR* milliseconds. The AR (automatic recall) parameter is described in part II, section 5.6. If the system is under load, the actual recall time could exceed the theoretical delay recall AR by several orders. If the response limits are exceeded before recall of the line driver, MARCIV EXPORT may shut down the line due to no response (or at best will now be out of synchronization with the Cyberlink line driver). It is recommended, therefore, that the value of AR be not greater than 400 milliseconds to ensure reliable operation of Cyberlink while it is functioning in non-dedicated mode.

7.5.3 FIELD LENGTH REQUIREMENTS

The central memory field length required by Cyberlink varies according to the values of the installation parameters and the number of active lines. In general, the approximate field length may be calculated from the following formula (all values are octal):

FL = n + k* 462B+nlinmx* 75B+bact * PRUSIZE * IP.BPRU + cm * 1300B

n Equivalent to 1700 if Cyberlink is rolled in or 300 if Cyberlink is rolled out.

k Equals 1 if the K displays are activated (IP.KDSP ≠ 0)

n.linmx Equals the highest port defined available on the 6673 or 6674 multiplexer

bact Equals the number of active buffers (0 if no lines transmitting or receiving)

cm Equals 1 if the line driver is placed in central memory resident at initialization time; otherwise cm=0. If Cyberlink is rolled out, cm=0 in all cases.

In addition to the above field length required at the control point, placing 1SL and 1CS in system central residents requires approximately 370B words.

FORTRAN 2.3 8

8.1 RELEASE DESCRIPTION

8.1.1 HARDWARE REQUIREMENTS

FORTRAN 2.3 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length requirement to run FORTRAN 2.3 is $40,000_8$.

8.1.2 CORRECTIONS

FORTRAN 2.3 includes all eligible PSR corrective code published through PSR Summary 348.

8.1.3 DEFICIENCIES

The following modification must be available in FORTRAN Run if it is to operate under KRONOS 2.1. It will be released in a future PSR summary under the IDENT FT30169A.

```
*IDENT         FT30169A
*/             APPLIES TO MODSET FT30169
*/             RELEASED AT PSR LEVEL 344.
*D,FT30169.    11
               BX6        X1
```

8.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update FORTRAN 2.3 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for FORTRAN 2.3, the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for FORTRAN 2.3 are RUN1, RUN2, and RUN3.

RUN1 Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.

RUN2 Use to collect the FORTRAN 2.3 binary code and generate directive files for later use as input to the procedure file GENSYS.

RUN3 Use to verify that the product is correctly installed.

KPL14 contains the FORTRAN 2.3 binary code.

PL21 contains the FORTRAN 2.3 source code.

PL2 contains the COMPASS 3.0 source code.

The tapes are described in detail in part I, sections 2.1.1, 2.1.2, and 2.8.

8.3 INSTALLATION PARAMETERS

In the standard release of FORTRAN 2.3, binary blocking code has been assembled into the I/O routines; the default condition for unformatted files is set to unblocked. To set the default condition for unformatted files to blocked, remove the identifier RM1567B by using the *YANK directive. If the identifier RM1567A is removed by *YANK, the code associated with binary blocking will not be assembled.

A trigger value is used to trigger an access to an I/O device for blocked binary files. On input, it represents the number of buffer words that must be available before a read buffer request is issued; on output, it is the number of words that must be ready for output before a write buffer request is issued.

The trigger value is set to a fixed percentage of the buffer length; however, if that percentage (or the remainder of the buffer) is less than one PRU, the PRU size is taken as the trigger value. The default is set to 80 percent; to change it, the value of the micro TRIGGER in SIO$ should be changed. (The trigger value must be less than 100.) For example, to change the triggering percentage to 50 percent:

```
*DELETE,SCU316M.94
TRIGGER MICRO 1,0,/50/
*COMPILE SIO$
```

The default text used for COMPASS assemblies is CPCTEXT in the released version; it must be changed for KRONOS 2.1. The modification is in the released binary code; but it must be added to the source code the first time it is modified and/or updated. The format of the modification is:

```
*D,R400014.1
CC8        DATA          OLSYSTEXT
```

Two user libraries (SYSLIB and RUN2P3) can be used with the FORTRAN 2.3 compiler. The arithmetic routines are the same in these two libraries; however, the I/O routines are not the same. RUN2P3 contains the SCOPE 3.4 object time I/O routines. SYSLIB contains object time I/O routines developed in conjunction with KRONOS using the KRONOS system I/O common decks. The user library RUN2P3 supports binary blocking. SYSLIB does not support binary blocking. However, the routines in SYSLIB, using the KRONOS common decks, provide interactive I/O to a terminal and take advantage of the CEJ option if it is available. RUN2P3 is the user library used with the FORTRAN 2.3 compiler in the released system. The user library can be changed to SYSLIB through use of the LDSET control card. SYSLIB is the default user library for all loads when unsatisfied externals exist after the primary user library has been used to satisfy the unsatisfied externals. FORTRAN 2.3 can be called using either the RUN23 or RUN control cards.

Refer to part III, section 1.7.3 for a list of FORTRAN 2.3 routines that reference the SCOPE 3.4 COMDECK IPARAMS.

8.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

8.4.1 RUN1

```
RUN1
RUN1,CM55000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF RUN 2.3.
COMMENT. THE FIRST FILE OF THE NEW KPL14 WILL BE THE NEWPL.
COMMENT. THE 2ND FILE WILL BE THE ABS. BINARY OF THE RUN 2.3 COMPILER.
COMMENT. THE THIRD FILE WILL BE THE RELOCATABLE BINARY OF THE
COMMENT. LIBRARY ROUTINES.
*VSN CARD.
LABEL(PL2,R,FI=COMPASS3P0*3P4,MT,D=HY,F=SI) LATEST COMPASS PL.
UPDATE,A,P=PL2,R,N=COMPCOM,I=0.
UNLOAD,PL2.
RETURN,PL2.
*VSN CARD.
LABEL(OLDPL,R,FI=RUN2P3*3P4,MT,D=HY,F=SI)   LATEST FORTRAN 2.3 PL.
UPDATE(F,N,W,X)
UNLOAD,OLDPL.
SKIPF,NEWPL.
RETURN,OLDPL.
COMPASS(I,S=CPCTEXT,S=IPTEXT,L=0,B=REL)     ACGOER - RECOVR$
COMPASS(I,S=CPCTEXT,S=IPTEXT,L=0,B=NEWPL.   RUN - Q8QDIAGP
SKIPEI,NEWPL.
WRITEF,NEWPL.
GTR(REL,LIB,,NR)REL/GETBA,SIO$
REWIND,REL.
COPYBR,REL,LIB,58.
SKIPR,REL,1.
COPYBR,REL,LIB,10.
SKIPR,REL,1.
COPY,REL,LIB.
REWIND,LIB.
COPYBF,LIB,NEWPL.            RELOCATABLE BINARY OF LIBRARY ROUTINES.
REWIND,NEWPL.
*VSN CARD.
LABEL(KPL14,W,L=RUN2P3*3P4,MT,D=HY,F=SI)    SCRATCH FOR NEW KPL14.
COPYEI,NEWPL,KPL14.
CATALOG,KPL14,R,N.
UNLOAD,KPL14.
*WEOR
*/        PLACE *FORTRAN 2.3* MODSET HERE.
*WEOF
```

8.4.2 RUN2

```
RUN2
RUN2,CM45000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB ADDS THE RUN V2.3 BINARIES TO THE SYSTEM COMMON
COMMENT. FILE ZZZPSBF FROM THE KPL14 RELEASE TAPE.
COMMENT. EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED
COMMENT. BY JOB RUN1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT. TO COMMON FILE ZZZZLDF.
*VSN CARD.
LABEL(KPL14,R,FI=RUN2P3*3P4,MT,D=HY,F=SI)   LATEST FORTRAN 2.3 PL.
SKIPF,KPL14,1.               SKIP OLDPL
COPYBF,KPL14,RUN.            ABSOLUTE BINARY OF OVERLAYS.
COPY,KPL14,RLIB.               RELOCATABLE ROUTINES FOR RUN2P3.
UNLOAD,KPL14.
RETURN,KPL14.
GTR(RLIB,SLIB)REL/GETBA,SIO$
REWIND,RLIB.
GTR(RLIB,MLIB,,NR)REL/DBLE-IBAIEX,LEGVAR-SQRT,TAN,ALNLOG-DVCHK
REWIND,RLIB.
GTR(RLIB,MLIB,,NR)REL/IDINT,OVERFL,RANF-RBAREX,TANH
REWIND,RLIB,MLIB,SLIB.
CATALOG,RUN,R.
COMMON,SYSTEM.
LIBEDIT(P=RUN,B=0)
SKIPR,NEW,1.
COPY,RUN,NEW.
REWIND,NEW.
COMMON,ZZZPSBF,ZZZZLDF.
COPYBF,NEW,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSIO
IF(EF=0)GOTO,10.
SET(R1=1)
GTR(SYSTEM,OLD)ULIB/SYSIO
10,LIBEDIT,N=LGO,B=SLIB,L=0.
LIBGEN,F,P=SYSIO,NX=1.
CATALOG,SYSIO,R,U.
SET(EF=0)
RETURN,OLD,LGO.
GTR(ZZZPSBF,OLD,,,S)ULIB/RUN2P3
IF(EF=0)GOTO,20.
SET(R2=1)
GTR(SYSTEM,OLD)ULIB/RUN2P3
20,LIBEDIT,N=LGO,B=RLIB,L=0.
LIBGEN,F,P=RUN2P3.
CATALOG,RUN2P3,R,U.
SET(EF=0)
RETURN,OLD,LGO.
GTR(ZZZPSBF,OLD)ULIB/SYSLIB
IF(EF=0)GOTO,30.
SET(R3=1)
GTR(SYSTEM,OLD)ULIB/SYSLIB
30,LIBEDIT,B=RLIB,N=LGO,L=0.              UPDATE SYSLIB.
LIBGEN,F,P=SYSLIB.
CATALOG,SYSLIB,R,U.
```

```
RETURN,LGO,NEW.
COPYEI,SYSIO,LGO.
BKSP,LGO.
COPYEI,RUN2P3,LGO.
BKSP,LGO.
COPYEI,SYSLIB,LGO.
REWIND,LGO.
LIBEDIT,P=ZZZPSBF,L=0.
RELEASE,ZZZPSBF.
RETURN,ZZZPSBF.
RENAME,ZZZPSBF=NEW.
COMMON,ZZZPSBF.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R1=1)COPYBR,,ZZZZLDF.
IF(R1=0)SKIPR,INPUT,1.
IF(R2=1)COPYBR,,ZZZZLDF.
*WEOR
*RENAME    OVL/RUN,RUN23
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,ULIB/SYSIO
*BEFORE    *,ULIB/RUN2P3
*BEFORE    *,ULIB/SYSLIB
*WEOR
*ADD       LIB20,OVL/RUN23-Q8DIAGP
*WEOR
*ADD       LIB21,ULIB/SYSIO
*WEOR
*ADD       LIB20,ULIB/RUN2P3
*WEOR
*ADD       LIB16,ULIB/SYSLIB
*WEOF
```

8.4.3 RUN3

```
RUN3
VRUN,CM55000,T200.
*ACCOUNT CARD.
COMMENT. THIS SIMPLE PROGRAM VERIFIES CORRECT INSTALLATION OF
COMMENT.   RUN 2.3
RUN(G)
COMMENT.   END OF JOB
EXIT.
COMMENT.     JOB FAILED
OB FAILED
*WEOR
      PROGRAM   SIMPLE (OUTPUT)
      REAL      A(1,10), B(10,2), C(1,2), D(1,2)
      DATA      D(1,1), D(1,2)  / 385.0, 5885.0 /
      DO  10  I=1, 10
        A(1,I) = I
   10 CONTINUE

      DO  30  J=1, 2
        DO  20  I=1, 10
          B(I,J) = I + 100*(J - 1)
   20   CONTINUE
   30 CONTINUE

      DO  60  I=1, 1
        DO  50  J=1, 2
          T = 0.0
          DO  40  K=1, 10
            T = T + A(I,K)*B(K,J)
   40     CONTINUE
          C(I,J) = T
   50   CONTINUE
   60 CONTINUE

      IF (C(1,1) .NE. D(1,1))  GO TO  70
      IF (C(1,2) .NE. D(1,2))  GO TO  70
      CALL REMARK(22HCOMPUTATION SUCCESSFUL)
      STOP

   70 PRINT 1000, C, D
 1000 FORMAT (≠1    FAILURE≠/*0 C-ARRAY IS *2F20.5/*0  SHOULD BE *2F20.
     . 5)
      CALL REMARK (7HFAILURE)
      CALL EXIT
      END
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
01.14.48.VRUN,CM55000,T200.
01.14.48.*ACCOUNT CARD.
01.14.48.ACCOUNT,USER1.
01.14.48.COMMENT. THIS SIMPLE PROGRAM VERIFIES CO
01.14.48.RRECT INSTALLATION OF
01.14.48.COMMENT.   RUN 2.3
01.14.50.RUN(G)
01.14.51.LGO
01.14.53.COMPUTATION SUCCESSFUL
01.14.53.STOP
01.14.54.COMMENT.   END OF JOB
```

FORTRAN EXTENDED 4.0 9

9.1 RELEASE DESCRIPTION

9.1.1 HARDWARE REQUIREMENTS

FORTRAN Extended 4.0 requires at least 49K of memory for maintenance purposes. The minimum field length required to run FORTRAN Extended 4.0 is $45,000_8$.

9.1.2 CORRECTIONS

FORTRAN Extended 4.0 includes all eligible PSR corrective code published through PSR Summary 348.

9.1.3 DEFICIENCIES

When the debugging compilation mode (control card parameter D) and the full cross reference features (R=2 or R=3) are used, more core may be required for compilation than otherwise. Detailed information is contained in the FORTRAN Extended Version 4.0 Reference Manual.

The intrinsic function SHIFT will not accept double word arguments (double precision or complex words).

FORTRAN Extended is designed to produce efficient object code. The rate of compilation tends to be higher on program units which avoid lengthy sequences of complicated arithmetic replacement statements (such as contiguous statements with no branching entries or exits). However, lengthy sequences tend to produce faster object code.

Code produced under the 6400 compiler option may not work properly on a 6600 because of optimization considerations. The compiler option to produce code for a 6600 produces binaries that can be run on either a 6400 or a 6600.

When the FTN control statement specifies either the C or E option, the compiler generates the object program in COMPASS source language form, rather than binary machine language. Since a local library set cannot be specified in a COMPASS source program, the user must place the loader directive LDSET (LIB=FORTRAN/SYSO) at appropriate points in the control card section of the job deck.

Binary blocked and/or random indexed files created under FORTRAN 3.0 cannot be handled directly by FORTRAN 4.0; they must be converted to a file structure accepted by FTN 4.0.

To run FORTRAN Extended 4.0 under KRONOS 2.1, it is necessary to add the modification FC40065, FC40083, and FCL0197 from PSR Summary 351 to the source code. This modification is included in the binary code on KPL15B, but it is not in the source code on PL8.

```
*PURGE    FEAT30G
*IDENT    FC40065
*INSERT     FE20033.40
          BX2    X1
*INSERT      FEAT30.3481
 SQ2AF    SA2    A1+B5       RESET RLIST2(J)
*DELETE     SQUEEZE.325,SQUEEZE.325
          COMUTMM SQ2AF
*DELETE      FEAT30.3490,FEAT30.3490
          EQ     SQ2AF
*DELETE      FEAT30.3492,FEAT30.3492
          EQ     SQ2AF
*DELETE     SQUEEZE.340,SQUEEZE.340
 LDMATCH  COMUTMM SQ2AF
*DELETE      FEAT30.3496,FEAT30.3496
          EQ     SQ2AF
*DELETE      USEDEF.98,USEDEF.98
      IH=TEMP.AND.7777B
          IH = SYMBOL( SYM1-2*IH+1) .A. SHIFT(1,DIM(X))
      IH=SHIFT(IH,30-DIM(X))
      USE(USELOC)=TEMP.OR.MODE.OR.IH
*IDENT    FC40083
*INSERT       ARITH.216
 PROGRAM  EQU    56B
*INSERT      FEAT30.1092
          SA3    ALLARR
          NZ     X3,SSP6     IF DEBUGGING ALL ARRAYS
*DELETE      FEAT30.1097,FEAT30.1100
          NZ     X3,SSP6     IF ANY DEBUGGING OF THIS ARRAY
*INSERT     F600105.90
          BX3    X2
          AX2    59
          BX2    X2-X3
*INSERT      FEAT30.1432
          SA3    PROGRAM
          UX2    B3,X3
          ZR     B3,IFL.2    IF A MAIN PROGRAM
*INSERT      FEAT30.1435
          SA1    =X0.CEP
          ADDREF X1,REF
*INSERT        POST.392
          SA3    =XX1R
          MI     X3,RJX.F    IF X1 IS LOCKED
          MX6    0
          SA6    =XX1SCR     X1SCR = 0
          PX6    X6
          SA6    =XX1S       USES[X1S] = 0
 RJX.F    BSS    0
*DELETE        POST.468,POST.468
          SA6    PARCEL.
*INSERT         JAM.913
          ZR     X6,GETRDF1   IF DEF HAS NO USES
*DELETE         JAM.922,JAM.922
 GETRDF1  SA1    ISSUED
          ZR     X1,CHOOSE
*DELETE        OPTB.56,OPTB.56
 X1R      ENTRY.
*INSERT        OPTB.87
          ENTRY  X1S,X1SCR
*DELETE        OPTB.239,OPTB.239
          BX6    -X0*X2
 *DELETE     FEAT30B.68,FEAT30B.68
           KSHFT = SHIFT(1,56)
 *DELETE     FEAT30B.70,FEAT30B.70
           IF( SHIFT(TEMP1,3) .GT. 0 ) GO TO 1510
*/ $$FCL0197      42    BUGCTL
*ID FCL0197
*/         MODS TO BUGCTL TO ELIMINATE THE CALL TO THE PP ROUTINE DSP
*/         IN NON SCOPE 3.4 SYSTEMS.
*I,BUGCTL.7
 ≡MD       IFC    LT, ≠MODEL≠ 75
           IPARAMS
 ≡MD       ELSE
 OS.NAME   MICRO  1,,≡SCOPE ≡
 OS.VER    MICRO  1,,≡2.0 ≡
 ≡MD       ENDIF
*I,BUGCTL.20
 ≡SD       IFC    EQ,≡≠OS.NAME≠≠OS.VER≠≡SCOPE 3.4 ≡
*D BUGCTL.22
 ≡SD       ENDIF

*D BUGCTL.37
*I BUGCTL.40

 ≡SD       IFC    EQ,≡≠OS.NAME≠≠OS.VER≠≡SCOPE 3.4 ≡
           BX6    X6-X6
*I BUGCTL.41
 ≡SD       ENDIF
*I BUGCTL.49
           SA7    DBGFIT.
*D BUGCTL.58
*D BUGCTL.60,BUGCTL.61
           BX4    X0*X6        FILE NAME
           SX7    X6           FIT ADDRESS
           SA7    DBGFIT.

 ≡SD       IFC    EQ,≡≠OS.NAME≠≠OS.VER≠≡SCOPE 3.4 ≡
*D BUGCTL.64
*D BUGCTL.65
*D BUGCTL.68
*D BUGCTL.70,BUGCTL.77
           SYSTEM DSP,RECALL,DSPTBL
*I BUGCTL.80
 ≡SD       ELSE
 BGC3B     BSS    0
 ≡SD       ENDIF
*C BUGCTL

 ≡MD       IFC    LT, ≠MODEL≠ 75
           IPARAMS
 ≡MD       ELSE
 OS.NAME   MICRO  1,,/SCOPE /
 ≡MD       ENDIF
*I FORSYS=.225
 ≡SD       IFC    EQ,/≠OS.NAME≠/KRONOS/
 Q8PL      CON    0LPL+54B     TEST FOR PL    *
 ≡SD       ELSE
*I FORSYS=.226
 ≡SD       ENDIF
*C FORSYS=
*/         THERE ARE  42 CORRECTIONS CARDS INCLUDING THIS COMMENT.
```

9.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update FORTRAN Extended is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for FORTRAN Extended, the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for FORTRAN Extended are FTNC1, FTNC2, FTNC3, F4LIB1, and F4LIB2.

FTNC1	Use to modify installation parameters or to add PSR updates to compiler overlay source code and to create a new tape (current KPL) containing updated source and binary code.
FTNC2	Use to collect the compiler overlay binary code and generate directive files for later use as input to the procedure file GENSYS.
FTNC3	Use to verify that the compiler overlays are correctly installed.
F4LIB1	Use to modify installation parameters or to add PSR updates to FORTRAN Extended library routines and to create a new tape (current KPL) containing updated source and binary code.
F4LIB2	Use to collect the library routines' binary code and generate directive files for later use as input to the procedure file GENSYS.

KPL15A contains the FORTRAN Extended 4.0 compiler overlay binary code.

KPL15B contains the FORTRAN Extended 4.0 library routines' binary code.

PL7 contains the FORTRAN Extended 4.0 compiler overlay source code.

PL8 contains the FORTRAN Extended 4.0 library routines' source code.

PL2 contains the COMPASS 3.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.9.

9.3 INSTALLATION PARAMETERS

FORTRAN Extended 4.0 routines that reference COMDECK IPARAMS are listed in part III, section 1.7.3.

The amount of core needed to compile jobs can be altered by modifying the size of compiler tables and scratch file buffers. Parameters that modify table sizes are included in the options listed at the end of this section. File buffer sizes can be changed by modifying the controlling routine FTN. FTN is the main overlay of the compiler (level 0, 0).

System compatibility parameters provide the following features. At the time of release, these parameters are set as listed below; UPDATE sequence numbers for all installation parameters may be determined by listing the OPTIONS portion of routine FTNTEXT and listing the routine FTN from the FORTRAN Extended 4.0 program library.

Parameter	Released Default Value	Significance
CCABT	Abort	Abort job on FTN control card error
CTIMO	Time message sent	Compiler message issued to control point dayfile (CPU seconds necessary for compilation)
LMAX	57	Lines per page listing limit (not applicable to intermixed COMPASS programs)

File names used by the compiler are:

INPUT	Source input
OUTPUT	Compiler listing
LGO	Relocatable object code
COMPS	COMPASS card images
ZZZZZRL	Intermediate language
ZZZZZRM	Reference map
ZZZZZOP	OPT=2 and D mode scratch

If the following control card options are not specified, the default settings at the time of release are as indicated.

Control Card Option	Released Default Value	Significance
A	No abort	Abort to EXIT(S) card if fatal compilation error occurs
B	Produces object code on standard file (LGO)	Produce object code file
C	C is not set	Use COMPASS assembler for compiler-generated code. If C is not selected, the FORTRAN assembler is used. (FORTRAN assembler saves about 60 percent of CPU time compared with COMPASS assembler.)
D	No debug mode	Debug mode of compilation
E	No file for editing	Format file for editing (COMPASS card image file is produced with *DECK cards for each program unit, suitable as input for UPDATE)
G	No compile and go	Compile and go option

I = lfn	lfn = INPUT
	lfn = COMPILE
		Select compiler input file
		I Not specified
		I Specified without lfn

list = lfn	lfn = OUTPUT
	list = L
		Select compiler listing file and listing options as follows:

Control Card Option	Released Default Value	Significance
	L	L List source code
	No list	O List COMPASS card images
	No list	X List ANSI violation diagnostics
	No list	N Suppress informative diagnostics
	No list	R (equivalent to long reference map option R=2)
		The R option may be used as a stand-alone option of the form R=n if a reference level other than the default is required. The values for n select the following reference map options.
		0 No reference map
	1	1 Short reference map
		2 Full cross reference map
		3 Full cross reference map plus common and equivalence information
OPT=level	OPT=1	Select level of optimization:
		0 Lowest optimization
		1 Slightly above FORTRAN Extended 2.0 optimization
		2 Program unit flow analysis used in optimization.
Q	Q is not set	Program verification option
ROUND=s	No rounding	s = */+- Select 1-4 of these operators to round arithmetic
SYSEDIT=ss		The following is intended for system programmer usage.
	None	ss=FILES Form execution time input/output unit references through indirect search of low core table rather than by using entry points and external references.
T	No error checking	Maximum error checking in mathematical library routines (basic external functions).
V		Selects minimal input/output buffer allocation (513 words per buffer) for compiler buffers during compilation. This may increase compile time but will allow jobs with a large number of declarative statements to compile in smaller field length than would be possible otherwise.
S	SYSTEXT	Specifies system text files to be used for intermixed COMPASS programs.

Control Card Option	Released Default Value	Significance
GT	SYSTEXT	Specifies system text files (sequential binary file) to be used for intermixed COMPASS programs.
Z	None	Forces all subroutine calls with no parameters to pass a parameter list consisting of a zero word.
PL	5000	Selects maximum number of allowable records on the OUTPUT file.
XT	OLDPL	Specifies external text (XTEXT) to be used for intermixed COMPASS programs.
	OPL	XT specified without lfn.

9.3.1 OPTIONS

The following listing is the options portion of FORTRAN Extended 4.0.

```
****                                                                            F600250      2
***      O P T I O N S - FORTRAN EXTENDED INSTALLATION PARAMETERS.              F600250      3
*                                                                               F600250      4
*                                                                               F600250      5
*        THE PARAMETERS DEFINED IN THIS DECK DETERMINE THE SPECIFIC             F600250      6
*        CONFIGURATION OF THE FORTRAN EXTENDED COMPILER.  WHEN INSTAL-          F600250      7
*        LING THE COMPILER, PARAMETERS SHOULD BE REVISED AS NECESSARY           F600250      8
*        TO DESCRIBE THE INSTALLATION*S EXACT HARDWARE CONFIGURATION            F600250      9
*        AND SPECIFIC FUNCTIONAL REQUIREMENTS.                                  F600250     10
*                                                                               F600250     11
*        ADDITIONAL INSTALLATION OPTION PARAMETERS WILL BE FOUND IN             F600250     12
*        THE PROGRAM DECK  < FTN > .                                            F600250     13


*        ****************************************************************** F600250     15
*        *                                                                * F600250     16
*        *                                                                * F600250     17
*        *   I N S T A L L A T I O N   W A R N I N G   N O T I C E        * F600250     18
*        *                                                                * F600250     19
*        *                                                                * F600250     20
*        *      AN ASTERISKED BOX SIMILAR TO THIS ENCLOSES EACH           * F600250     21
*        *      PARAMETER THAT IS INTENDED TO BE CHANGED WHEN             * F600250     22
*        *      INSTALLING THE FORTRAN EXTENDED COMPILER.                 * F600250     23
*        *                                                                * F600250     24
*        *      NO OTHER PARAMETER IN THE *OPTIONS* DECK SHOULD           * F600250     25
*        *      BE CHANGED.                                               * F600250     26
*        *                                                                * F600250     27
*        *      CONTROL DATA CORPORATION ACCEPTS NO RESPONSIBILITY        * F600250     28
*        *      FOR SATISFACTORY PERFORMANCE OF THE FORTRAN               * F600250     29
*        *      EXTENDED COMPILER IF THIS NOTICE IS DISREGARDED.          * F600250     30
*        *                                                                * F600250     31
*        *                                                                * F600250     32
*        ****************************************************************** F600250     33

***      PROCESSOR NAME AND VERSION NUMBER                                      F600307      2

LPNAME   MICRO  1,7,/FTN    /      LANGUAGE PROCESSOR NAME                      F600307      4
VER      MICRO  1,3,/4.0/                                                       F600307      5
```

```
***      CENTRAL PROCESSOR MODEL / SERIES DECLARATIONS.                         F600250     35
                                                                                F600250     36
                                                                                F600250     37
*        ****************************************************************** F600250     38
*        *                                                                * F600250     39
*        *      TO INSTALL FORTRAN EXTENDED ON A CYBER 70/ MODEL XX       * F600250     40
*        *      PROCESSOR, SET THE FOLLOWING SYMBOL TO THE APPRO-         * F600250     41
*        *      PRIATE MODEL NUMBER, AND DELETE THE SUBSEQUENT            * F600250     42
*        *      SYMBOL [ MACHINE. ].                                      * F600250     43
*        *                                                                * F600250     44
*        *                                                                * F600250     46
*        *      REFERENCES -- FTN, FTNTEXT.                               * F600250     47
*        ****************************************************************** F600250     48
```

```
               *          ************************************************************ F600250     50
               *          *                                                          * F600250     51
               *          *      TO INSTALL FORTRAN EXTENDED ON A 6X00 / 7X00        * F600250     52
               *          *      PROCESSOR, SET THE FOLLOWING SYMBOL TO THE APPRO-   * F600250     53
               *          *      PRIATE SERIES NUMBER, AND DELETE THE PRECEDING      * F600250     54
               *          *      SYMBOL [ CYBERMOD ].                                * F600250     55
               *          *                                                          * F600250     56
      6600     MACHINE.  EQU     6600B         6X00 / 7X00 SERIES NUMBER               F600250     57
               *          *                                                          * F600250     58
               *          *      REFERENCES -- FAX, FTN, FTNMAC, OPTB, REFMAP.       * F600250     59
               *          ************************************************************ F600250     60


               C          IF     DEF,CYBERMOD                                          F600250     62
               C          ELSE                                                         F600250     70
                          IFLT   MACHINE.,6600B,2                                      F600250     71
                          IFEQ   MACHINE.,6600B,2                                      F600250     74
        74     CYBERMOD  EQU     74B                                                   F600250     75
               S          SKIP                                                         F600250     76
               S          ENDIF                                                        F600250     79
                                                                                       F600250     80
               *          [MACHINE.] REFERENCES -- FAX, FTN, FTNMAC, OPTB, REFMAP.     F600250     81
                                                                                       F600250     82
      6600     MACHINE   EQU     MACHINE.    FORMER NAME OF SYMBOL                     F600250     83

      6600     TARGET    OCTMIC MACHINE.                                               F600307      7
               TARGET    MICRO  1,2,/≠TARGET≠/                                         F600307      8
               X         MICRO  1,1,/≠TARGET≠/                                         FC40002      1
               Y         MICRO  2,1,/≠TARGET≠/                                         FC40002      2
               Y         MICRO  ≠X≠*10-59+≠Y≠,1,/012345XXXXXXXXXXXXXX/                 FC40002      3
               VALID     MICRO  1,,/≠X≠≠Y≠/                                            FC40002      4


               **         DEFINE ≠MODEL≠ MICRO.                                        F600250     85
                                                                                       F600250     86
        74     MODEL     OCTMIC CYBERMOD                                               F600250     87
                                                                                       F600250     88
               *          [≠MODEL≠] REFERENCES -- FTN, FTNTEXT.                        F600250     89


FTNTEXT - FORTRAN EXTENDED ASSEMBLY TEXT          COMPASS 3.73129      73/05/22. 02.23.04.      PAGE      5
OPTIONS - FORTRAN EXT INSTALLATION PARAMETERS     OPTIONS

               *          ************************************************************ F600250     91
               *          *                                                          * F600250     92
               *          *      EARLY MODEL 6600 PARAMETER (STORE OUT OF ORDER).    * F600250     93
               *          *                                                          * F600250     94
               *          *      WHEN THE TARGET, I.E., OBJECT TIME, CPU IS A 6600   * F600250     95
               *          *      CLASS A, B OR C MACHINE, SERIAL NUMBER 1→47, THAT   * F600250     96
               *          *      LACKS FCO 20436, SET THE FOLLOWING PARAMETER TO ZERO. * F600250   97
               *          *      (FCO 20436 RESOLVES A STORE OUT OF ORDER PROBLEM.)  * F600250     98
               *          *                                                          * F600250     99
         1     STOR6600  EQU     1                                                     F600250    100
               *          *                                                          * F600250    101
               *          *      REFERENCES -- OPTB.                                 * F600250    102
               *          ************************************************************ F600250    103
```

```
          *        ************************************************************ F600250    105
          *        *                                                          * F600250    106
          *        *      HARDWARE INTEGER MULTIPLY FEATURE.  VALUES ARE --   * F600250    107
          *        *                                                          * F600250    108
          *        *      1 = INTEGER MULTIPLY INSTALLED.                     * F600250    109
          *        *      0 = INTEGER MULTIPLY NOT INSTALLED.                 * F600250    110
          *        *                                                          * F600250    111
          *        ************************************************************ F600250    112
          *        *                                                          * F600250    113
          *        *      [CPTIMULT] - DEFINE AVAILABILITY AT COMPILE TIME.   * F600250    114
          *        *                                                          * F600250    115
0         CPTIMULT EQU    0            INTEGER MULTIPLY NOT AVAIL AT COMPILE TIME  F600250    116
          *        *                                                          * F600250    117
          *        *      REFERENCES -- FTNTEXT.                              * F600250    118
          *        ************************************************************ F600250    119
          *        *                                                          * F600250    120
          *        *      [OBJIMULT] - DEFINE AVAILABILITY AT OBJECT TIME.    * F600250    121
          *        *                                                          * F600250    122
0         OBJIMULT EQU    0            INTEGER MULTIPLY NOT AVAIL AT OBJECT TIME   F600250    123
          *        *                                                          * F600250    124
          *        *      REFERENCES -- ARITH, DOPRE, MACROX, MACRS.          * F600266      1
          *        ************************************************************ F600250    126


          **       7DM (DATA MANAGER) APPLICABILITY.  DENOTES WHETHER OR NOT 7DM  F600250    141
          *        WILL BE USED, BASED ON THE PROCESSOR MODEL.  SINCE THE SYMBOL  F600250    142
          *        IS MORE CONVENIENT IN A NEGATIVE SENSE, IT IS DEFINED AS --    F600250    143
          *                                                                       F600250    144
          *        1      7DM DOES NOT APPLY (MODEL 74 OR LOWER / 6X00 SERIES).   F600250    145
          *        0      7DM APPLIES (MODEL 75 OR HIGHER / 7X00 SERIES).         F600250    146
                                                                                  F600250    147
                   IFC    GE, ≠MODEL≠ 75 ,3                                       F600250    148
1         NEDM     EQU    1                                                       F600250    152



FTNTEXT - FORTRAN EXTENDED ASSEMBLY TEXT          COMPASS 3.73129      73/05/22. 02.23.04.      PAGE     6
OPTIONS - FORTRAN EXT INSTALLATION PARAMETERS     OPTIONS

          MODEL    IFC    LT, ≠MODEL≠ 75                                          F600250    154
0         CPERM    EQU    0            DO NOT ASSEMBLE 6RM                        F600250    165
          MODEL    ENDIF                                                          F600250    168


          *        ************************************************************ F600250    170
          *        *      FILE CONTAINING COMPASS RANDOM PL WITH /COMPCOM/.   * F600250    171
          *        *                                                          * F600250    172
          PLCMPS   MICRO  1,, COMPCOM                                             F600270      1
          *        *                                                          * F600250    174
          *        ************************************************************ F600250    175


          PLRM     IFNE   CPERM*NEDM,0                                            F600250    177
          PLRM     ENDIF                                                          F600250    185
          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      CHARACTER APPENDED TO FILE NAMES FOR UNIQUENESS.      * F600250
          *        *                                                            * F600250
           C       MICRO  1,,/E/                                                  F600250
          *        *                                                            * F600250
          *        *      REFERENCES -- BUGCTL,DBGFET,FTNMAC,LISTIO,PH1CTL.     * F600250
          *        ************************************************************** F600250
```

```
          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      F I X E D   L E N G T H   T A B L E S.                * F600250
          *        *                                                            * F600250
          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      LENGTH OF INTERMEDIATE STORAGE AREA FOR ALL CON-      * F600250
          *        *      STANTS IN ANY ONE SOURCE STATEMENT.  EACH CONSTANT    * F600250
          *        *      BEGINS ON A NEW WORD BOUNDARY, AND IS PACKED AS TEN   * F600250
          *        *      DISPLAY-CODED CHARACTERS PER WORD.                    * F600250
          *        *                                                            * F600250
     454   CONSTORS EQU   3000D                                                   F600250
          *        *                                                            * F600250
          *        *      REFERENCES -- SCANNER.                                * F600250
          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      LENGTH OF SAVED OPERATOR STACK.                       * F600250
          *        *                                                            * F600250
     240   MXOSL   EQU    160D                                                    F600250
          *        *                                                            * F600250
          *        *      REFERENCES -- ARITH.                                  * F600250
          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      LENGTH OF SAVED FUNCTION RESULT TABLE.                * F600250
          *        *                                                            * F600250
     132   MXFRSTB EQU    90D                                                     F600250
          *        *                                                            * F600250
          *        *      REFERENCES -- ARITH.                                  * F600250
          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      LENGTH OF <ARLIST> BUFFER.                            * F600250
          *        *                                                            * F600250
    3200   ARLSZ   EQU    3200B                                                   F600250
          *        *                                                            * F600250
          *        *      REFERENCES -- PS1CTL.                                 * F600250
          *        ************************************************************** F600250


          *        ************************************************************** F600250
          *        *                                                            * F600250
          *        *      SELECT DIVISION BY RECIPROCAL MULTIPLICATION.         * F600250
          *        *                                                            * F600250
          *        *      WHEN THE FOLLOWING PARAMETER IS ZERO, <ARITH> WILL    * F600250
          *        *      ATTEMPT TO REPLACE X/C BY X*(1/C), WHERE [X] IS AN    * F600250
          *        *      EXPRESSION AND [C] IS A CONSTANT (REDUCTION IN        * F600250
          *        *      STRENGTH).  SET NON-ZERO TO DEFEAT THE FEATURE.       * F600250
          *        *                                                            * F600250
       0   NOINVERT EQU   0                                                       F600250
          *        *                                                            * F600250
          *        *      REFERENCES -- ARITH.                                  * F600250
          *        ************************************************************** F600250
```

```
                 *          ************************************************************ F600250    247
                 *          *                                                           * F600250    248
                 *          *      ERROR LISTING LINE LIMIT.                            * F600250    249
                 *          *                                                           * F600250    250
         144     ERRMAX     EQU    1000D                                                  F600250    251
                 *          *                                                           * F600250    252
                 *          *      REFERENCES -- ERPRO, LSTPRO.                         * F600250    253
                 *          ************************************************************ F600250    254
```

```
***        MICROS TO DEFINE THE DEFAULT CONTROL CARD OPTIONS.                       F600250   1162
*                                                                                   F600131    154
*                                 OPTIONS  SELECTED     NOT SELECTED   FILE         F600131    155
 CC.A      MICRO  1,, 0            A             1             0        NA          FE20005      1
* CHANGE ↑LGO↑ MICRO BELOW         B=LFN  ALWAYS              NA       LFN          F600131    157
 CC.G      MICRO  1,, 0            G            -1             0        NA          F600131    158
 CC.C      MICRO  1,, 0            C            -1             0        NA          F600131    159
 CC.D      MICRO  0                D            NA             0       LFN          F600131    160
 CC.E      MICRO  1,, 0            E            -1             0                    F600131    161
 CC.GT     MICRO  1,,$SYSTEXT$    GT=LFN ALWAYS               NA       LFN          F600250   1163
* CHANGE ↑COMPS↑ MICRO BELOW       E=LFN  ALWAYS              NA       LFN          F600131    162
* CHANGE ↑INPUT↑ MICRO BELOW       I=LFN  ALWAYS              NA       LFN          F600131    163
 CC.I      MICRO  1,,$COMPILE$     I                                                F600261      1
 CC.L      MICRO  1,, 1            L             1             0        NA          F600250   1164
 CC.LCM    MICRO  1,, 0           LCM           -1             0        NA          F600162      1
 CC.O      MICRO  1,, 0            O            -1             0        NA          F600131    165
 CC.X      MICRO  1,, 0            X            -1             0        NA          F600131    166
 CC.N      MICRO  1,, 0            N            -1             0        NA          F600131    167
* CHANGE ↑OUTPUT↑ MICRO BELOW      L=LFN  ALWAYS              NA       LFN          F600131    168
 CC.PL     MICRO  1,, 5000         PL=N          N          5000        NA          F600215      1
 CC.Q      MICRO  1,, 0            Q            -1             0        NA          F600158      1
 CC.R      MICRO  1,, 1            R=     0, 1, 2, OR 3       NA        NA          F600131    169
 CC.S      MICRO  1,,$SYSTEXT$     S=LFN  ALWAYS              NA       LFN          F600131    170
 CC.T      MICRO  1,, 0            T            -1             0        NA          F600131    171
 CC.XT     MICRO  1,,$OPL$         XT                                               F600296      1
 CC.Z      MICRO  1,, 0            Z            -1             0        NA          F500159      4
 CC.OPT    MICRO  1,, 1            OPT=   0, 1, OR 2          NA        NA          F600131    172
 CC.ROUND  MICRO  1,, 0BS19        ROUND= 1BS19 FOR /                               FE30787      1
*                                         2BS19 FOR *                               FE30787      2
*                                         4BS19 FOR -                               FE30787      3
*                                        10BS19 FOR +                               FE30787      4
 CC.IDENT  MICRO  1,, 0            SYSEDIT=IDENT                                    F600131    177
*                                               -1             0       NA           F600131    178
 CC.FILES  MICRO  1,, 1            SYSEDIT=FILES                                    F600138      7
*                                                0             1        NA          F600138      8
           IFC    EQ,/≠CC.OPT≠/0/,1                                                 FE20041      1
           ***                                                                        FTN          59
           *         ASSEMBLY OPTIONS                                                 FTN          60
           *                                                                          FTN          61
                     IFEQ    TEST,0                                                   FTN          62
                     RMT                                                              FTN          63
            W=SBW    EQU     CCERR             W=SPY BIN WIDTH                        FTN          64
            M=LFN    EQU     CCERR             MACROX PATCH FACILITY                  FTN          65
                     RMT                                                              FTN          67
                     ENDIF                                                            FTN          71
                                                                                      FTN          72
                                                                                      FTN          73
           ***                                                                        F600138       9
           *         LOCAL ASSEMBLY OPTIONS REFERENCED ONLY IN ≠FTN≠                  F600138      10
           *                                                                          F600138      11
       1    CCABT    EQU     1                 SET TO NON ZERO VALUE FOR ABORT ON     FTN          75
           *                                   CONTROL CARD ERRORS                    FTN          76
            EM       IFNE    CPERM,0                                                  F600250    1165
                                                                                      F600250    1166
                     ENTRY   LMAX                                                     F600250    1167
                                                                                      F600250    1168
            EM       ENDIF                                                            F600250    1169
                                                                                      F600250    1170
      71    LMAX     EQU     57D               LINES / PAGE                           FTN          77
           PSR       MICRO   1,4,/P348/                                               PSRLEVEL      1
       1    CTIMO    EQU     1                                                        F600240       1
           *                                   =0 NO MESSAGE                          F600131     184
           *                                   =1 ISSUE DAYFILE MESSAGE FOR CPU       F600131     185
           *                                      COMPILATION TIME                    F600131     186


    6600    CDC6466  OCTMIC  MACHINE.          MACHINE THAT CODE IS GENERATED FOR     F600131     188
                                                                                      FTN          82
            K        MICRO   1,,/BS9/          *1000B                                 FTN          85
            TV       MICRO   1,,*1/2*          TRIGGER VALUE                          FTN          86
       1    TVS      EQU     1                 LOG2( 1/TV )                           FTN          87
                                                                                      F600077       2
           ***                                                                        F600077       3
           *         MIN.FL,MIN.DFL = MINIMUM FIELD LENGTH TO EXECUTE IN.             F600077       4
           *         MIN.FL = MAX(LWA LOAD)+400B+1000B ROUNDED UP TO NEXT 1000B       F600077       5
           *         WHERE:                                                           F600077       6
           *          MAX(LWA LOAD) = LAST WORD ADDR OF THE OVERLAY LOADS             F600077       7
           *          400B IS FOR MINIMUM RLIST AND COMPS BUFFERS                     F600077       8
           *          1000B IS FOR WORKING STORAGE                                    F600077       9
           *         CURRENTLY PASS 1 ( THE 1.1 OVERLAY ) IS THE LARGEST              F600077      10
           *                                                                          F600077      12
           *         FOR FL > MIN.FL+3K STANDARD SIZE BUFFERS (L.RLIST,L.COMPS)       F600077      13
           *         ARE ALLOCATED FOR RLIST AND COMPS.                               F600077      14
           *                                                                          F600077      15
   42000    MIN.FL   EQU     42≠K≠+TEST*2000B  MINIMUM FL FOR EXECUTION               F600322       1
   61000    MIN.DFL  EQU     MIN.FL+17≠K≠      MINIMUM FL IF D OPTION SELECTED        F600077      17
                                                                                      F600077      18
   42000    MIN.FL   OCTMIC  MIN.FL                                                   F600077      19
            MIN.FL   MICRO   1,2,/≠MIN.FL≠/                                           F600077      20
   61000    MIN.DFL  OCTMIC  MIN.DFL                                                  F600077      21
            MIN.DFL  MICRO   1,2,/≠MIN.DFL≠/                                          F600077      22
```

```
FTN - 0.0 OVERLAY , INITIALIZATION AND I/O CONTROL          COMPASS 3.73129      73/05/17. 11.30.27.       PAGE      9
           ASSEMBLY OPTIONS

                      *         DEFAULT FILE NAMES                                                 FTN          94
                                                                                                   FTN          95
                       INPUT    MICRO  1,,$INPUT$                                                  FTN          96
                       OUTPUT   MICRO  1,,$OUTPUT$                                                 FTN          97
                       LGO      MICRO  1,,$LGO$           BINARY FILE NAME                         FTN          98
                       COMPS    MICRO  1,,$COMPS$         COMPASS IMAGES                           FTN          99
                       RLIST    MICRO  1,,$ZZZZZRL$       INTERMEDIATE CODE SCRATCH FILE           F600172      20
                       RMAP     MICRO  1,,$ZZZZZRM$       REFERENCE MAP SCRATCH FILE               F600172      21
                       OPT      MICRO  1,,$ZZZZZOP$       SUPER MODE SCRATCH FILE                  F600172      22
                                IFNE   TEST,0,3                                                    F600172      23
                       RLIST    MICRO  1,,$RLIST$         SCRATCH FILE NAME ( IN TEST MODE )       F600081       2
                       RMAP     MICRO  1,,$FTNRMAP$                                                F600172      24
                       OPT      MICRO  1,,$FTNOPT$                                                 F600172      25


                      *         DEFAULT BUFFER SIZES                                               FTN         105
                      *          MAY BE ADJUSTED BY FTN, DEPENDING ON FIELD LENGTH AND             FTN         106
                      *          CONTROL CARD OPTIONS                                              FTN         107
                                                                                                   FTN         108
                                                                                                   F600250    1173
                       BUFL     MICRO  1,, 1002B   MINIMUM BUFFER SIZE                             F600250    1174
                                                                                                   F600250    1175
                                RMT                                                                F600250    1176
                       L.INPUT  EQU    IBUFL                                                       F600250    1177
                       L.OUTPUT EQU    OBUFL                                                       F600250    1178
                                RMT                                                                F600250    1179
                                                                                                   F600250    1180
                 1002  L.COMPS  EQU    ≠BUFL≠      COMPASS SOURCE IMAGE BUFFER LENGTH              F600250    1181
                 2004  L.LGO    EQU    ≠BUFL≠*2    BINARY OUTPUT BUFFER LENGTH                     F600250    1182
                 2004  L.OPT    EQU    ≠BUFL≠*2    MAX OPT (OPT=2) SCRATCH FILE BUFFER LENGTH      F600250    1183
                 2004  L.RLIST  EQU    ≠BUFL≠*2    INTERMED LANG (R-LIST) FILE BUFFER LENGTH       F600250    1184
                 1002  L.RMAP   EQU    ≠BUFL≠      LONG REFERENCE MAP (R=3) BUFFER LENGTH          F600250    1185
                                                                                                   FTN         116
                      *         THE LENGTH≠S OF THE ≠INPUT≠ AND ≠LGO≠ BUFFERS MUST BE              FTN         117
                      *         .GT. 1000B SINCE THEY MAY BE ON TAPE.                              FTN         118
                                                                                                   FTN         119
                    3  NOPTLVL  EQU    3                  NUMBER OF LEVELS OF OPTIMIZATION         FTN         120
```

9.3.2 COMPILER PROGRAM LIBRARY STRUCTURE

When a full update is performed on the program library, eight records are produced on the compile file as follows:

	Contents	Overlay Level	Deck Names Needed to Compile
1.	FTNMAC		FTNMAC
2.	FTNTEXT		FTNTEXT
3.	Controller	(0, 0)	FTN
		(1, 0)	LSTPRO. SNAP
	Pass 1--normal	(1, 1)	PS1CTL. PH1CTL
	Pass 2	(1, 2)	CLOSE2. MACROX
		(1, 3)	PS2CTL
4.	Pass 1-1/2	(1, 5)	PASS15. CHECK
5.	Pass 1--debug	(1, 4)	DBGPHCT. SAVREGS
6.	Debug COPYL deck		FORMDBG
7.	RDUMP		MACROR. RDUMPCP
8.	CMPAR		CMPAR

FTNMAC

This routine is a text collection of the macros necessary to assemble an FTN compiled job using the COMPASS assembler. Assembling the FTNMAC record from the compile file will produce a record suitable for entering the system through SYSEDIT. The assembled routine is in system text format and can be used in FTN and COMPASS assemblies through the S parameter specification.

FTNTEXT

This record consists of a collection of macros, micros, and symbol definitions which facilitate the assembly of the FTN version 4. 0 compiler. Assembling the FTNTEXT record from the compile file will produce a record in system text format that will be referenced through the GT parameter for compiler assemblies.

(0, 0) OVERLAY (FTN)

This overlay is a batch controller that accomplishes the following.

Breaks down the parameters on the FTN control card

Initializes the compiler to reflect compile time options

Contains basic I/O routines and system communication routines

Handles COMPASS-FTN communication for intermixed COMPASS language program units

(1, 0) OVERLAY (LSTPRO$)

This overlay holds information necessary for communication between passes of the compiler. It contains the symbol/label table lookup routine (needed by pass 1, pass 1-1/2, and the assembly phase of pass 2), a routine to allow the usage of formatted I/O in the FORTRAN coded routines which are a part of the debug and pass 1-1/2 portions of the compiler, and a snap facility active only when the compiler is in test mode.

(1, 1) OVERLAY (PS1CTL$)

This overlay is the first pass of the compiler under normal mode (when the debug mode is not selected by the D control card option); it handles lexical, syntactical and semantic analysis of a FORTRAN program unit. The FORTRAN program unit is translated through a lexical element language (E list) to an intermediate language (R list).

(1, 2) OVERLAY (CLOSE2$)

This overlay is the second pass of the compiler. Using the R list generated by pass 1, this pass produces COMPASS instructions which are assembled (optionally by the COMPASS assembler at the expense of CPU time) into an object deck.

(1, 3) OVERLAY (PS2CTL)

This overlay is loaded only if errors occur in the FORTRAN program. This overlay issues full line error messages for both fatal and informative errors.

(1, 5) OVERLAY (PASS15$)

This overlay is selected optionally (OPT=2 on the FTN control card); it is executed between the normal pass 1 and pass 2. This pass (1-1/2) uses the F list language generated by pass 1 as input, and utilizing flow analysis, produces a modified R list file for input to pass 2. This file will reflect the movement of invariant code from frequently executed regions to those less frequently executed and it will allow register allocation over loops.

(1, 4) OVERLAY (PASS14$)

This overlay serves the same purpose as overlay (1, 1) with the addition that debug statements will be acted upon as specified by the D option in the FTN control card. On the program library, only the routines unique to the debug mode constitute the fifth record. The sixth record of the compile file constitutes a full overlay (1, 4).

DEBUG LIBEDIT DECK

This deck contains a series of zero length COMPASS routines with identifiers of the (1, 1) overlay routines needed to complete the (1, 4) overlay. By compiling the fifth and sixth records of the compile file to the same object file, the object file may be used as an OLD file for LIBEDIT with the object file of the (1, 1) overlay serving as the replacement file to produce a full (1, 4) overlay.

RDUMP

This utility routine is to be used with the test mode of the compiler. For a single program unit, RDUMP will dump the R list file generated by pass 1 of the compiler. (Since an attempt is made to keep this file completely in core, the V (for very small buffers) option on the control card should be used; it is active in test mode only.)

CMPAR

This routine compares two object records and lists discrepancies by loader table. It is unlikely that a FORTRAN program using the FTN 4.0 internal assembler will have the same object representation as when it is handled by the COMPASS assembler (though a load from either of the object routines will produce the same core image in non-BSS storage).

9.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

9.4.1 FTNC1

```
FTNC1
FTNC1,CM65000,T7000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB UPDATES THE FORTRAN EXTENDED COMPILER AND CREATES
COMMENT.  A NEW KPL15A WITH PROGRAM LIBRARY AND COMPILER BINARIES.
COMMENT.
COMMENT.  THE CONTENTS OF THE NEW KPL15A ARE --
COMMENT.          FILE 1 -- FORTRAN COMPILER PROGRAM LIBRARY
COMMENT.          FILE 2 -- RELOCATABLE BINARIES OF THE NON DEBUG
COMMENT.                    OVERLAYS
COMMENT.          FILE 3 -- RELOCATABLE BINARIES OF THE DEBUG OVERLAYS
COMMENT.          FILE 4 -- FTNMAC AND ABSOLUTE BINARIES OF THE COMPILER
COMMENT.
COMMENT.  THIS JOB USES THE COMPASS V3.0 PL TO CAPTURE THE COMPASS
COMMENT.  COMMON DECK COMPCOM WHICH IS NECESSARY FOR COMPASS/FTN
COMMENT.  INTERFACE.
COMMENT.
COMMENT.  W A R N I N G -- THIS JOB DOES A SYSEDIT OF THE FIRST FIVE
COMMENT.  COMPASS GENERATED OVERLAYS.  THIS IS NECESSARY SO THAT THESE
COMMENT.  OVERLAYS CAN BE USED TO COMPILE FORTRAN CODE IN THE LAST
COMMENT.  TWO OVERLAYS.  THIS SYSEDIT MAY CONFLICT WITH A
COMMENT.  CONCURRENTLY RUNNING TIMESHARING SUBSYSTEM (TELEX, ETC... ).
COMMENT.
*VSN CARD.
LABEL(OLDPL,R,FI=COMPASS3P0*3P4,MT,D=HY,F=SI)
UPDATE(Q,P=OLDPL,N=COMPCOM,C=0)
UNLOAD(OLDPL)
RETURN,OLDPL.
*VSN CARD.
LABEL(OLDPL,R,FI=FTN4P0COMP*3P4,MT,D=HY,F=SI)
REWIND,OLDPL.
UPDATE(F,P=OLDPL,N=NEWPL,W,R=C)
REWIND,NEWPL.
SKIPF,NEWPL.
UNLOAD(OLDPL)
RETURN,OLDPL.
COMPASS,I,L=0,B=FTNMAC,S=0.
COMPASS,I,L=0,B=FTNTEXT,S=0.
COMPASS,I,L=0,B=SYSMAIN,S=IPTEXT,G=FTNTEXT.
REWIND,SYSMAIN.
MAP.
RFL,70000.
LOAD,SYSMAIN.
SYSEDIT,I=0,B=FTN.
RETURN,FTN.
RFL,65000.
```

```
FTN,I,L=0,GT=FTNTEXT,S=0,OPT=1,B=SYSOPT.
FTN,I,L=0,GT=FTNTEXT,S=0,OPT=1,B=FDEBUG.
FTN,I,L=0,GT=FTNTEXT,S=0,OPT=1,B=FDEBUG.
REWIND,SYSMAIN,SYSOPT,FDEBUG.
COPYBR,FDEBUG,SYSDBG.
NOEXIT.
LIBEDIT,P=FDEBUG,R,B=SYSMAIN,I=0,N=SYSDBGA.
REWIND,SYSMAIN,SYSOPT,SYSDBGA.
COPYBF,SYSDBGA,SYSDBG.
REWIND,SYSDBG.
COPYBF,SYSMAIN,SYSFTN.
BKSP,SYSFTN.
COPYBF,SYSOPT,SYSFTN.
REWIND,SYSFTN.
COPYBF,SYSFTN,NEWPL.
BKSP,SYSFTN.
COPYBF,SYSDBG,SYSFTN.
REWIND,SYSFTN.
RETURN,FTN.
RFL,76000.
LOAD(SYSFTN)
RFL,30000.
REWIND,SYSFTN,FTN,FTNMAC.
COPYBF,SYSFTN,NEWPL.
COPYBR,FTNMAC,NEWPL.
COPYBF,FTN,NEWPL.
CATALOG,NEWPL,R,N=0.
*VSN CARD.
LABEL(KPL15A,W,FI=FTN4P0COMP*3P4,MT,D=HY,F=SI)
REWIND,KPL15A.
COPYEI,NEWPL,KPL15A,V.
*WEOR
*C COMPCOM
*WEOR
*/  PLACE FTN4.0 MODIFICATIONS HERE
*WEOF
```

9.4.2 FTNC2

```
FTNC2
FTNC2,CM55000,T1000,P30.
*ACCOUNT CARD.
COMMENT. THIS JOB ADDS THE FTN V4.0 COMPILER BINARY TO THE SYSTEM
COMMENT. COMMON FILE ZZZPSBF FROM THE KPL15A RELEASE TAPE.
COMMENT. EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT. JOB FTNC1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT. TO COMMON FILE ZZZZLDF FOR GENERATING A NEW SYSTEM DEADSTART.
COMMENT.
*VSN CARD.
LABEL(KPL15A,R,FI=FTN4P0COMP*3P4,MT,D=HY,F=SI)
REWIND,KPL15A.
SKIPF,KPL15A,3.
COPYBF,KPL15A,FTNC.
UNLOAD,KPL15A.
RETURN,KPL15A.
CATALOG,FTNC,R.
```

```
COMMON,ZZZPSBF,ZZZZLDF.
COPYBF,FTNC,ZZZPSBF.
BKSP,ZZZPSBF.
COPYBR,,ZZZZLDF.
*WEOR
*ADD      LIB26,OVL/FTNMAC-PASS14$
*WEOF
```

9.4.3 FTNC3

```
FTNC3
FTNC3,CM55000,T200.
*ACCOUNT CARD.
COMMENT.  THIS PROGRAM VERIFIES CORRECT INSTALLATION OF
COMMENT.  FORTRAN EXTENDED 4.0.
FTN(G,OPT=1)
BKSP(INPUT)
RETURN(LGO)
FTN(G,OPT=2)
BKSP(INPUT)
RETURN(LGO)
FTN(G,OPT=0)
COMMENT.    FTN 4.0 INSTALLED.
EXIT.
COMMENT.  FTN 4.0 VERIFICATION FAILED.
*WEOR
      PROGRAM   SIMPLE (OUTPUT)
      REAL      A(1,10), B(10,2), C(1,2), D(1,2)
      DATA      D(1,1), D(1,2)  / 385.0, 5885.0 /

      DO  10  I=1, 10
        A(1,I) = I
  10  CONTINUE

      DO  30  J=1, 2
        DO 20  I=1, 10
          B(I,J) = I + 100*(J - 1)
  20    CONTINUE
  30  CONTINUE

      DO  60  I=1, 1
        DO  50  J=1, 2
          T = 0.0
          DO  40  K=1, 10
            T = T + A(I,K)*B(K,J)
  40      CONTINUE
          C(I,J) = T
  50    CONTINUE
  60  CONTINUE

      IF (C(1,1) .NE. D(1,1))  GO TO  70
      IF (C(1,2) .NE. D(1,2))  GO TO  70
      CALL REMARK(22HCOMPUTATION SUCCESSFUL)
      STOP
  70  PRINT 1000, C, D
 1000 FORMAT (≠1    FAILURE≠*0 C-ARRAY IS *2F20.5/*0  SHOULD BE *2F20.
     .5)
5)
      CALL REMARK (7HFAILURE)
      CALL EXIT
      END
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
01.05.08.FTNC3,CM55000,T200.
01.05.08.*ACCOUNT CARD.
01.05.08.ACCOUNT,USER1.
01.05.09.COMMENT.  THIS PROGRAM VERIFIES CORRECT
01.05.09.INSTALLATION OF
01.05.09.COMMENT.  FORTRAN EXTENDED 4.0.
01.05.09.FTN(G,OPT=1)
01.05.10.       .391 CP SECONDS COMPILATION TIME
01.05.10.LGO
01.05.12.COMPUTATION SUCCESSFUL
01.05.12.    STOP
01.05.12.BKSP(INPUT)
01.05.12.RETURN(LGO)
01.05.13.FTN(G,OPT=2)
01.05.17.       .618 CP SECONDS COMPILATION TIME
01.05.17.LGO
01.05.18.COMPUTATION SUCCESSFUL
01.05.18.    STOP
01.05.19.BKSP(INPUT)
01.05.19.RETURN(LGO)
01.05.19.FTN(G,OPT=0)
01.05.22.       .275 CP SECONDS COMPILATION TIME
01.05.22.LGO
01.05.24.COMPUTATION SUCCESSFUL
01.05.24.    STOP
01.05.24.COMMENT.    FTN 4.0 INSTALLED.
```

9.4.4 F4LIB1

```
F4LIB1
F4LIB1,CM60000,T3000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB UPDATES AND CREATES THE BINARIES OF THE FTN 4.0
COMMENT.  OBJECT LIBRARY ROUTINES AND CREATES A NEW KPL15B.
COMMENT.  THE FIRST FILE OF THE NEW KPL15B WILL BE THE NEWPL.
COMMENT.  THE SECOND FILE WILL BE THE BINARIES OF THE LIBRARY ROUTINES.
COMMENT.
*VSN CARD.
LABEL(OLDPL,R,FI=FTNLIBS*3P4,MT,D=HY,F=SI)
REWIND,OLDPL.
UPDATE(F,P=OLDPL,N=NEWPL,W,R=C)
SKIPF,NEWPL.
UNLOAD(OLDPL)
RETURN,OLDPL.
COMPASS,I,L=0,S=IOTEXT,S=IPTEXT,B=LIBRARY.
FTN,I,L=0,S=IOTEXT,S=IPTEXT,B=LIBRARY.   FTN DEBUG LIBRARY
REWIND,LIBRARY.
COPYBF,LIBRARY,NEWPL.
*VSN CARD.
LABEL(KPL15B,W,FI=FTNLIBS*3P4,MT,D=HY,F=SI)
CATALOG,NEWPL,R,N=0.
REWIND,NEWPL,KPL15B.
COPYEI,NEWPL,KPL15B,V.
UNLOAD,KPL15B.
*WEOR
*/  PLACE FTN 4.0 LIBRARY MODIFICATIONS HERE
*WEOF
```

9.4.5 F4L1B2

```
F4LIB2
F4LIB2,CM50000,T1000,P30.
*ACCOUNT CARD.
COMMENT.  THIS JOB ADDS THE FTN V4.0 OBJECT LIBRARY TO THE SYSTEM
COMMENT.  COMMON FILE ZZZPSBF FROM THE KPL15B RELEASE TAPE AS THE USER
COMMENT.  LIBRARY -FORTRAN-.
COMMENT.  EITHER THE RELEASED VERSION OR THE LATEST VERSION CREATED BY
COMMENT.  JOB F4LIB1 MAY BE USED. IT ALSO ADDS THE NECESSARY DIRECTIVES
COMMENT.  TO COMMON FILE ZZZZLDF FOR GENERARING A NEW SYSTEM DEADSTART.
COMMENT.
*VSN CARD.
LABEL(KPL15B,R,FI=FTNLIBS*3P4,MT,D=HY,F=SI)
REWIND,KPL15B.
SKIPF,KPL15B.
COPYBF,KPL15B,LGO.
LIBGEN,F,P=FORTRAN.
COMMON,ZZZPSBF,ZZZZLDF.
CATALOG,FORTRAN,U,R.
COPYBR,,ZZZZLDF.
COPYBF,FORTRAN,ZZZPSBF.
BKSP,ZZZPSBF.
*WEOR
*ADD      LIB26,ULIB/FORTRAN
*WEOF
```

PERT/TIME 1.2 10

10.1 RELEASE DESCRIPTION

10.1.1 HARDWARE REQUIREMENTS

PERT/Time 1.2 requires three tape units and 65K central memory to run.

10.1.2 CORRECTIONS

PERT/Time 1.2 includes all eligible PSR corrective code published through PSR Summary 348.

10.1.3 DEFICIENCIES

No user file can be named TAPE1 through TAPE6 because these file names are used internally.

10.2 INSTALLATION PROCEDURE

The procedure to update PERT/Time 1.2 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for PERT/Time, the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for PERT/Time 1.2 are PRT1 and PRT3.

PRT1	Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
PRT3	Use to verify that the product is correctly installed.

KPL16 contains the PERT/Time 1.2 binary code.

PL13 contains the PERT/Time 1.2 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.10.

The installation deck PRT1 compiles PERT/Time 1.2, forms the necessary overlays, and writes these overlays on the tape containing the PERT/Time program library. PERT/Time must be executed from this overlay tape; it cannot be executed from the running system.

To execute from the overlay tape (KPL16), run a job similar to the following.

```
PERT, T1000, CM10000.
ACCOUNT(usernum, password)
LABEL(KPL16, R, FI=PERT*3P4, MT, D=HY, F=SI)
COPYBF(KPL16, KPL16)
COPYBF(KPL16, PERT66)
PERT66.
7/8/9      )
  .        |
  .        |
7/8/9      }  PERT networks
  .        |
  .        |
6/7/8/9    )
```

If a master tape is made, insert the following after card 3.

```
REQUEST, TAPE6, MT.
```

If an old master tape is input and a new master tape saved, insert the following after card 3.

```
REQUEST, TAPE4, MT.        OLD MASTER
REQUEST, TAPE6, MT.        NEW MASTER
```

If either option is used, insert a RESOURC card before the REQUEST card.

Successive PERT/Time networks may be processed by batching as follows:

```
K
L
M
X
Y
W
A     PERT BATCHED NETWORKS
●
●
●
A
Z
K
L
●
●
●
Z
```

(Input for each network starts with the K control card and ends with the Z control card.)

10.3 INSTALLATION PARAMETERS

None.

10.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

10.4.1 PRT1

```
PRT1
PRT1,CM55000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF PERT 1.2 ON KPL16.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=PERT*3P4,MT,D=HY,F=SI)
ASSIGN,NE,X.
UPDATE,F,N,W.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
RETURN,OLDPL.
RUN,S,,,COMPILE,X.
MAP.
LINK,P=RUN2P3,B=NEWPL.
SKIPEI,NEWPL.
REWIND,LGO.
WRITEF,NEWPL.
COPYBF,LGO,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL16,W,FI=PERT*3P4,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL16,V.
UNLOAD,KPL16.
*WEOR
*/         PLACE *PERT/TIME* MODSET HERE.
*WEOF
```

10.4.2 PRT3

```
PRT3
PRT3,CM100000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
*THIS SIMPLE PROGRAM VERIFIES CORRECT
*INSTALLATION OF PERT/TIME.
LABEL(KPL16,R,FI=PERT*3P4,MT,D=HY,F=SI)
COPYBF,KPL16,KPL16.
COPYBF,KPL16,PERT66.
UNLOAD,KPL16.
RETURN,KPL16.
PERT66.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
K        9        64         3         5         0
L        1         7
M
XC36 PERT VERIFICATION PROGRAM               333
Y091764091764     1    0CLASSIFICATION
W  123456                                                  503
A        00    01  00  00  00A091764                            012345678900ORG1
A  19    01    01             091764EVENT  1
A        01    02  50  80  90                                       01      ORG4
A        01    03  10  40  50                                   001234567890ORG3
A        01    04  40  50  80                                       51      ORG1
A  28    02    02             111364EVENT  2
A        02    05  10  40  90                                       02      ORG4
A        02    09  40  40  60                                       31      ORG1
A  38    03    03             101664EVENT  3
A        03    05  20  40  70                                       03      ORG4
A        03    06  20  40  70                                       01      ORG1
A  48    04    04             102664EVENT  4
A        04    07  20  20  40                                       53      ORG2
A        04    08  20  40  50                                       52      ORG1
A  57    05    05             121464EVENT  5
A        05    10  20  40  60                                   023456718560ORG5
A  67    06    06             111664EVENT  6
A        06    10  20  30  40                                       03      ORG1
A        06    11  20  30  40                                       81      ORG2
A  77    07    07             111664EVENT  7
A        07    11  30  60  90                                       52      ORG2
A  87    08    08             112364EVENT  8
A        08    12  30  40  70                                       54      ORG1
A  9     09    09             122464EVENT  9
A        09    13  00  00  00                                       31      ORG1
A 106    10    10             010865EVENT 10
A        10    15  30  50  70                                       01      ORG2
A 116    11    11             122464EVENT 11
A        11    16  20  30  70                                       81      ORG2
```

```
A 126    12    12                122464EVENT 12
A        12    14   20   30   50                                      51      ORG5
A 136    13    13                122464EVENT 13
A        13    18   10   10   20                                      71      ORG4
A 145    14    14                011865EVENT 14
A        14    17   30   50   50                                      41      ORG2
A 155    15    15                021265EVENT 15
A        15    19   10   30   40                                      61      ORG5
A        15    22   20   70  110                                      71      ORG1
A 165    16    16                012365EVENT 16
A        16    20   20   50   60                                      82      ORG2
A 17     17    17                022265EVENT 17
A        17    21   20   25   30                                      42      ORG1
A 185    18    18                011165EVENT 18
A        18    22   50   60   80                                      72      ORGJ
A 194    19    19                030565EVENT 19
A        19    20   00   00   00                                      82      ORG2
A 204    20    20                030565EVENT 20
A        20    22   50   80   90                                      71      ORG4
A        20    25  100  150  170                                      11      ORG1
A        20    27  120  130  180                                      21      ORG4
A 214    21    21                031565EVENT 21
A        21    24   25   30   40                                      41      ORG2
A 224    22    22                050365EVENT 22
A        22    23                                                     72      ORG2
A 234    23    23                050365EVENT 23
A        23    26   30   40   60                                      22      ORG2
A        23    27   30   50   80                                      21      ORG1
A 244    24    24                040665EVENT 24
A        24    25   30   50  110                                      42      ORG5
A 253    25    25                061865EVENT 25
A        25    27   40   50   90                                      11      ORG1
A 263    26    26                060465EVENT 26
A        26    27   20   30   40                                      23      ORG5
A 273    27    27                080265EVENT 27
Z
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.31.20.PRT3,CM100000,P30,T1000.
00.31.20.ACCOUNT, USER1 .
00.31.21.*ACCOUNT CARD.
00.31.21.VSN,KPL16=0KPL16.
00.31.22.*THIS SIMPLE PROGRAM VERIFIES CORRECT
00.31.22.*INSTALLATION OF PERT/TIME.
00.31.22.LABEL(KPL16,R,FI=PERT*3P4,MT,D=HY,F=SI)
00.31.23.MT51, ASSIGNED TO KPL16   , VSN=0KPL16.
00.31.24.COPYBF,KPL16,KPL16.
00.31.27.COPY COMPLETE.
00.31.28.COPYBF,KPL16,PERT66.
00.31.32.COPY COMPLETE.
00.31.33.UNLOAD,KPL16.
00.31.33.RETURN,KPL16.
00.31.34.PERT66.
00.31.58.STOP
00.31.58.*END OF JOB.
```

11.1 RELEASE DESCRIPTION

SIMSCRIPT 3.0 consists of a compiler and an execution library. All deck names used in both the compiler and the execution library are listed below. They are contained in a single Update record, and they are ordered to make generation of an overlay tape as easy as possible.

The first 73 decks (ADEF through ASML) belong to the SIMSCRIPT 3.0 compiler and are written in SIMSCRIPT.

ADEF contains the definition deck necessary to compile the compiler. It must always be written on the COMPILE file when any compiler routine written in SIMSCRIPT is to be recompiled.

MAINAD through WASM (15 decks) represent the definition processor part of the compiler (overlay 1.0).

DEFINE through REFL (52 decks) represent the translator part of the compiler (overlay 2.0).

ASMBL through ASML (5 decks) represent the assembler phase of the compiler (overlay 3.0).

The deck TREND$ marks the end of the SIMSCRIPT language routines; it contains a *WEOR.

The following 38 decks (SIMI5 through ALLOUT) belong to the SIMI5 compiler and are written in COMPASS.

SIMI5 through XXERROR represent the root segment of the compiler (overlay 0,0).

XX$DEFS through CONUM are part of the definition processor overlay (overlay 1,0).

XX$TRAN through PACKIT are part of the translator overlay (overlay 2,0).

XX$FAKE represents overlay 2,1.

XX$ASMB through ALLOUT are part of the assembler overlay (overlay 3,0).

The deck UTEND$ marks the end of the compiler routines written in COMPASS; it contains a *WEOR.

The 26 decks from XX$MAIN through XX$FMT represent the SIMI5 execution library and are written in COMPASS.

Loader control cards are included in the following decks: SIMI5, XX$DEFS, XX$TRAN, XX$FAKE, and XX$ASMB.

The SIMI5 compiler automatically generates the following LDSET table to reference the necessary libraries.

SIMLIB	Should contain the SIMI5 execution library
FORTRAN	
SYSIO	

11.1.1 HARDWARE REQUIREMENTS

SIMSCRIPT 3.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length required to run SIMSCRIPT 3.0 is $55,000_8$.

11.1.2 CORRECTIONS

SIMSCRIPT 3.0 includes all eligible PSR corrective code published through PSR Summary 348.

11.1.3 DEFICIENCIES

None.

11.2 INSTALLATION PROCEDURE

The procedure to install and update SIMSCRIPT 3.0 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for SIMSCRIPT 3.0, the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for SIMSCRIPT 3.0 are SMS1, SMS2, and SMS3.

SMS1	Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
SMS2	Use to collect the SIMSCRIPT 3.0 binary code and generate directive files for later use as input to the procedure file GENSYS.
SMS3	Use to verify that the product is correctly installed.

KPL17 contains the SIMSCRIPT 3.0 binary code.

PL27 contains the SIMSCRIPT 3.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.11.

11.3 INSTALLATION PARAMETERS

There are no installation parameters.

SIMSCRIPT 3.0 routines that reference SCOPE 3.4 Texts IPARAMS are listed in part III, section 1.7.3.

11.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1

11.4.1 SMS1

```
SMS1
SMS1,CM60000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF SIMSCRIPT 3.0 ON KPL17.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=SIMI53P0*3P4,MT,D=HY,F=SI)
UPDATE,F,N,W,C=0.
UNLOAD,OLDPL.
UPDATE,F,P=NEWPL,L=0.
RETURN,OLDPL.
SIMI5,I=COMPILE,D=0,B=SIMTR,L=0.
COMPASS,I,S=CPCTEXT,B=SIMUT,L=0.
COMPASS,I,S=IOTEXT,B=SIMLB,L=0.
REWIND,SIMUT,SIMTR.
COPYX,SIMUT,LGO,REL/CONUM.
COPYX,SIMTR,LGO,REL/WASM.
COPYX,SIMUT,LGO,REL/PACKIT.
COPYX,SIMTR,LGO,REL/REFL.
COPYBF,SIMUT,LGO.
BKSP,LGO.
COPYBF,SIMTR,LGO.
SKIPEI,NEWPL.
MAP.
LINK,F,P=SYSIO,B=NEWPL.
SKIPEI,NEWPL.
LIBGEN,F=SIMLB,P=SIMLIB.
REWIND,SIMLIB,SIMLB,LGO.
COPYBF,SIMLIB,NEWPL.
COPYBF,SIMLB,NEWPL.
COPYBF,LGO,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL17,W,FI=SIMI53P0*3P4 MT,D=HY,F=SI)
COPYEI,NEWPL,KPL17,V.
*WEOR
*/           PLACE *SIMSCRIPT* MODSET HERE.
*WEOR
*IDENT       KRONOS
*I,XX$NIT.10
             RJ       =XLOADER     FORCE REDUCE(-)
*WEOF
```

11.4.2 SMS2

```
SMS2
SMS2,CM15000,P30,T1000.
*ACCOUNT CARD
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL17,R,FI=SIMI53P0*3P4,MT,D=HY,F=SI)
SKIPF,KPL17.
COPYBF,KPL17,SMSCRPT.
RETURN,KPL17.
COMMON,ZZZZLDF,ZZZPSBF.
CATALOG,SMSCRPT,R.
COPYBR,,ZZZZLDF.
COPYBF,SMSCRPT,ZZZPSBF.
BKSP,ZZZPSBF.
*WEOR
*ADD      LIB27,OVL/SIMI5-ULIB/SIMLIB
*WEOF
```

11.4.3 SMS3

```
SMS3
SMS3,CM60000,P30,T1000.
*ACCOUNT CARD.
*THIS SIMPLE PROGRAM VERIFIES CORRECT
*INSTALLATION OF SIMSCRIPT.
SIMI5.
LGO.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
+                                                                 RRAY I
      MAIN  MAGIC REPORT
      DIMENSION ARRAY(25,25)
      DO, FOR N = (3)(5)(2)
      LET I = N
      LET J =(N+1)/2
      DO, FOR K = (1)(N*N)
      STORE K IN ARRAY(I,J)
      IF K EQ K/N*N, GO 1
      IF J EQ N,LET J = 0
      IF I EQ N, LET I = 0
      LET I = I+1
      LET J = J+1
      GO 2
    1 LET I = I-1
    2 LOOP
      CALL MAGIC(N,(N*N+1)*N/2,ARRAY)
      LOOP
      STOP
      END
      FUNCTION RRAY (V,I,J)
      DIMENSION V(25,25)
      STORE V(I,J) IN RRAY
      RETURN
      END
      REPORT MAGIC(N,KSUM,V)
   21       FOR I = (1)(N)                                             X
X                            THIS IS A *** BY *** MAGIC SQUARE.  ALL OF X
THE ROWS,COLUMNS AND DIAGONALS SUM TO        *                    12   X
 X                N   N   KSUM
X                     CONTINUATION OF THE *** BY *** MAGIC SQUARE      X
 X             N  N                                               22
X          *     *     *     *     *     *     *     *     *     *     *
     *     *     *     *     *     *     *     *     *     *       1     *
 X        21(RRAY(V,I,J))
  X         FOR J = (1)(N)                                             X
            END                                                        X
     END
*WEOR
1X         0                                          SYSTEM SPEC. CARD
                                                  END OF INITIALIZATION.
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.39.45.SMS3,CM60000,P30,T1000.
00.39.46.ACCOUNT,USER1.
00.39.46.*ACCOUNT CARD.
00.39.47.*THIS SIMPLE PROGRAM VERIFIES CORRECT
00.39.47.*INSTALLATION OF SIMSCRIPT.
00.39.50.SIMI5.
00.40.03.LGO.
00.40.08.    STOP  SIMI5
00.40.09.*END OF JOB.
```

SIMULA 1.0 12

12.1 RELEASE DESCRIPTION

12.1.1 HARDWARE REQUIREMENTS

SIMULA 1.0 can be maintained on the same hardware configuration as KRONOS 2.1. The minimum field length to run SIMULA 1.0 is $30,000_8$.

12.1.2 CORRECTIONS

SIMULA 1.0 includes all eligible PSR corrective code published through PSR Summary 348.

12.1.3 DEFICIENCIES

Code procedures and direct files are not implemented.

The KRONOS control card REDUCE cannot be used when SIMULA programs are executed because the space following the program is used as the stack area for variables.

Segment mode loading (SIMULA control card options S, U, R, and G) is deactivated; attempts to use these options result in job termination.

SIMULA expects compiler input to conform to the KRONOS 63 character set (IP.CSET=IP.C63). Accordingly, if the system is configured to a different character set, input must be translated to conform to the expected display code values for colon, less than, left bracket, and right bracket.

12.2 INSTALLATION PROCEDURE

The procedure to install and update SIMULA 1.0 is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for SIMULA, the following materials are necessary.

KPL2 contains the jobs to update, install, and verify installation; the jobs required for SIMULA are SIM1, SIM2, and SIM3.

SIM1	Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
SIM2	Use to collect the SIMULA binary code and generate directive files for later use as input to the procedure file GENSYS.
SIM3	Use to verify that SIMULA is correctly installed.

KPL18 contains the SIMULA 1.0 binary code.

PL19 contains the SIMULA 1.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.12.

12.3 INSTALLATION PARAMETERS

None.

12.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

12.4.1 SIM1

```
SIM1
SIM1,CM56000,P30,T1400.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF SIMULA 1.0 ON KPL18.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=SIMULA1P0*3P4,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
RETURN,OLDPL.
COMPASS,I,S=IPTEXT,L=0.
MAP.
LINK,B=NEWPL.
SKIPEI,NEWPL.
WRITEF,NEWPL.
COMPASS,I,S=IPTEXT,B=NEWPL,L=0.
WRITEF,NEWPL.
REWIND,LGO.
COPYBF,LGO,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL18,W,FI=SIMULA1P0*3P4,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL18,V.
*WEOR
*/        PLACE *SIMULA* MODSET HERE.
*WEOF
```

12.4.2 SIM2

```
SIM2
SIM2,CM45000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL18,R,FI=SIMULA1P0*3P4,MT,D=HY,F=SI)
SKIPF,KPL18.
COPYBF,KPL18,SIMULA.
COPYBF,KPL18,LIB.
UNLOAD,KPL18.
RETURN,KPL18.
CATALOG,SIMULA,R.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBF,SIMULA,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/SYSMISC
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R3=1)
GTR(SYSTEM,OLD)ULIB/SYSMISC
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=SYSMISC.
CATALOG,SYSMISC,R,U.
LIBEDIT,P=ZZZPSBF,B=SYSMISC,C,L=0.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,ULIB/SYSMISC
*WEOR
*ADD       LIB28,OVL/SIMULA-SIM5
*WEOR
*ADD       LIB21,ULIB/SYSMISC
*WEOF
```

12.4.3 SIM3

```
SIM3
SIM3,CM56000,P30,T100.
*ACCOUNT CARD.
*THIS SIMPLE PROGRAM VERIFIES CORRECT
*INSTALLATION OF SIMULA 1.0.
SIMULA(L,X)
LGO.
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
≠COMMENT≠ VALIDATION PROGRAM
IF SIMULA IS WELL IMPLEMENTED ON THE SYSTEM, FOLLOWING LINES MUST OCCUR
         LIBRARY 0 EXISTS
         LIBRARY 1 EXISTS
         LIBRARY 2 EXISTS
         LIBRARY 3 EXISTS
         GARBAGE COLLECTOR EXISTS;

SIMULATION ≠BEGIN≠ ≠REF≠(LINK) Z; ≠TEXT≠ T; ≠REAL≠ X;
   ≠PROCEDURE≠ PRINT(T); ≠TEXT≠ T; ≠BEGIN≠ OUTTEXT(T); OUTIMAGE ≠END≠;

                                          PRINT(↓LIBRARY 0 EXISTS↓);
X..= COS(0)+SQRT(4);                      PRINT(↓LIBRARY 1 EXISTS↓);
Z..- ≠NEW≠ LINK;                          PRINT(↓LIBRARY 2 EXISTS↓);
X..= LINE;
T..- TEXT(↓ABC↓);                         PRINT(↓LIBRARY 3 EXISTS↓);
GARBAGE(0);                               PRINT(↓GARBAGE COLLECTOR EXISTS↓)
≠END≠
         FINIS
*WEOR
DATASET,END
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.32.36.SIM3,CM56000,P30,T100.
00.32.36.ACCOUNT,USER1.
00.32.37.*ACCOUNT CARD.
00.32.37.*THIS SIMPLE PROGRAM VERIFIES CORRECT
00.32.37.*INSTALLATION OF SIMULA 1.0.
00.32.37.SIMULA(L,X)
00.32.40.LGO.
00.32.42.*END OF JOB.
```

SORT/MERGE 4.0 13

13.1 RELEASE DESCRIPTION

Sort/Merge 4.0 runs in conjunction with KRONOS 2.1 and Record Manager 1.0. The system consists of two control card directive formats and a macro sort format. Directive format 1 is based on upward compatibility toward 7000 Sort/Merge and format 2 is based on the Sort/Merge 3.0 control card format. The macro sort format is also based on 7000 compatibility. Sort/Merge 4.0 is a more modularized package, consisting of overlay modules that are in core only when necessary. For example, a disk sort does not require the tape merge overlay modules. This product is designed to optimize speed and core space as well as to utilize Record Manager 1.0 capabilities.

13.1.1 HARDWARE REQUIREMENTS

Sort/Merge 4.0 can be maintained on the same minimum hardware configuration as KRONOS 2.1. If the tape sort option is used, additional magnetic tape units are required: polyphase requires three; balanced requires four. The minimum field length requirement to run Sort/Merge 4.0 is 25,000$_8$.

13.1.2 CORRECTIONS

Sort/Merge 4.0 includes all eligible PSR corrective code published through PSR Summary 348.

13.1.3 DEFICIENCIES

The following modification must be available in Sort/Merge 4.0 if it is to operate under KRONOS 2.1. Corrective code similar to the following will be released at PSR level 357 under the IDENT ST40083.

```
*IDENT     SORTMOD
*/         THE FOLLOWING IS A MODIFICATION     IN SORT TO ELIMINATE
*/         A BUFFER ARG. ERROR IN SORT LEVEL 348. USE S=IOTEXT,S=CPUTEXT.
*D,MRGCON.788,819
*D,MRGCON.601,608
           SB3     X5
           BX0     X4
           SB5     B1
           BX3     X1
 GNM.5     SA2     X3+B5
           SX2     X2
           EVICT   X2
           SB5     B5+B1
           LE      B5,B3,GNM.5
           SB5     B1
 GNM.6     SA2     X3+B5
           SX2     X2
           RECALL  X2
           SB5     B5+B1
           LE      B5,B3,GNM.6
           BX2     X3
*C MRGCON
```

13.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update Sort/Merge is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for Sort/Merge, the following materials are necessary.

KPL2 contains the jobs to modify, update, install, and verify installation; the jobs required for Sort/Merge are SRM1, SRM2, and SRM3.

SRM1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
SRM2	Use to collect the binary code and generate directive files for later use as input to the procedure file GENSYS.
SRM3	Use to verify that the product is correctly installed.

KPL19 contains the Sort/Merge 4.0 binary code.

PL10 contains the Sort/Merge 4.0 source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.13.

13.3 INSTALLATION PARAMETERS

Sort/Merge 4.0 will assemble to use the CMU hardware, depending upon IP.CMU from IPTEXT. (Refer to part III, section 1.7.3 for definition of IP.CMU.) To override this parameter, make the following changes.

To install Sort/Merge 4.0 without CMU code:

```
*I FEAT24B.4
  BDP.INST     EQU     BDP.NO
```

To install Sort/Merge 4.0 with CMU code:

```
*I FEAT24B.4
  BDP.INST     EQU     BDP.YES
```

13.4 INSTALLATIONS JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

13.4.1 SRM1

```
SRM1
SRM1,CM56000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF SORT 4.0 ON KPL19.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(OLDPL,R,FI=SORT4P0*3P4,MT,D=HY,F=SI)
UPDATE,F,N,W,X.
UNLOAD,OLDPL.
SKIPEI,NEWPL.
RETURN,OLDPL.
COMPASS,I,S=IOTEXT,S=CPUTEXT,S=IPTEXT,B=REL,L=0.
GTR(REL,LIB,,NR)REL/DSMCON-MACPRO,SOCHKR
GTR(REL,LIB,,NR)REL/ENDPRO,EXTRACT,KYCPL-BUFALL
GTR(REL,LIB,,NR)REL/TRNSRT,TSC-NEXRCM
GTR(REL,LIB,,NR)REL/FMC-FSRTGET,MRGCON-SRTGET
REWIND,REL,LIB.
COPYX,REL,REL,TEXT/OVERLAY,3.
COPY,REL,LGO.
MAP.
COMPASS,I,S=0,B=NEWPL,L=0.
LINK,F,P=SYSIO,B=SMC.
REWIND,SMC.
COPYX,SMC,NEWPL,OVL/DUMMY,2.
COPYBF,SMC,NEWPL.
COPYBF,LIB,NEWPL.
COPYBF,LGO,NEWPL.
REWIND,NEWPL.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL19,W,FI=SORT4P0*3P4,MT,D=HY,F=SI)
COPYEI,NEWPL,KPL19,V.
*WEOR
*/        PLACE *SORT/MERGE* MODSET HERE.
*WEOF
```

13.4.2 SRM2

```
SRM2
SRM2,CM45000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL19,R,FI=SORT4P0*3P4,MT,D=HY,F=SI)
SKIPF,KPL19.
COPYBF,KPL19,SORTMRG.
COPYBF,KPL19,LIB.
UNLOAD,KPL19.
RETURN,KPL19.
COMMON,ZZZZLDF,ZZZPSBF.
CATALOG,SORTMRG,R.
COPYBF,SORTMRG,ZZZPSBF.
NOEXIT.
GTR(ZZZPSBF,OLD,,,S)ULIB/COBOL
IF(EF=0)GOTO,10.
COMMON,SYSTEM.
SET(R3=1)
GTR(SYSTEM,OLD)ULIB/COBOL
10,LIBEDIT,N=LGO,B=LIB,L=0.
LIBGEN,F,P=COBOL.
CATALOG,COBOL,R,U.
LIBEDIT,P=ZZZPSBF,B=COBOL,C,L=0.
SKIPEI,ZZZPSBF.
COPYBR,,ZZZZLDF.
BKSP,ZZZPSBF.
IF(R3≠0)COPYBR,,ZZZZLDF.
*WEOR
*BEFORE    *,REL/*
*WEOR
*BEFORE    *,ULIB/COBOL
*WEOR
*ADD       LIB15,OVL/SMTEXT
*ADD       LIB30,OVL/SORTMRG-SORT40
*WEOR
*ADD       LIB25,ULIB/COBOL
*WEOF
```

13.4.3 SRM3

```
SRM3
SRM3,CM54000,P30,T500.
*ACCOUNT CARD.
*DIRECTIVE FORMAT 1 VERIFICATION DECK - SORT/MERGE 4.0
*JOB SORTS 10 RECORDS - ASCENDING ORDER
FILE(INPUT,RT=Z,BT=C,FL=80,ERL=1)
FILE(OUTPUT,RT=Z,BT=C,FL=80,ERL=1)
SORTMRG(7C)
*DIRECTIVE FORMAT 2 VERIFICATION DECK - SORT/MERGE 4.0
*JOB SORTS 10 RECORDS - ASCENDING ORDER
SORTMRG(6C)
*MACRO SORT VERIFICATION DECK -         SORT/MERGE 4.0
*JOB SORTS 10 RECORDS - ASCENDING ORDER
COMPASS,S=SMTEXT,S=IOTEXT.
LDSET(LIB=COBOL/SYSIO)
LGO.
*WEOR
      SORT     VAR=DISK
      FILE     INPUT=INPUT(N)
NPUT=INPUT(N)
      FILE     OUTPUT=OUTPUT(N)
UTPUT=OUTPUT(N)
      FIELD    A(1,5,LOGICAL)
      KEY      A(A)
      END
*WEOR
M1111IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M2222IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M6666IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M3333IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M4444IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M7777IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M5555IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M9999IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M8888IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M8765IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*WEOR
SORT(1,1,80,,1)
FILE(INPUT,S,B,,N,N)
FILE(OUTPUT,O,B,,N,N)
KEY(A,C,1,5)
RECORD(I,U,80)
END
*WEOR
M1111IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M2222IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M6666IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M3333IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M4444IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M7777IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M5555IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M9999IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M8888IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M8765IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*WEOR
```

```
          IDENT   START
          SST
          ENTRY   START
 START    BSS     0
          SORT    VAR=DISK
          FILES   (INPUT,INPUT),(OUTPUT,OUTPUT)
          KEY     1,1,5,0,LOGICAL,A
          ENDRUN
          FILE    LFN=INPUT,RT=Z,BT=C,FL=80,FET=FT1,ERL=1
          FILE    LFN=OUTPUT,RT=Z,BT=C,FL=80,FET=FT2,ERL=1
 FT1      BSSZ    13
 FT2      BSSZ    13
          END     START
*WEOR
M1111IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M2222IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M6666IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M3333IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M4444IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M7777IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M5555IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M9999IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M8888IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
M8765IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.

```
00.40.38.SRM3,CM54000,P30,T500.
00.40.38.ACCOUNT,USER1.
00.40.39.*ACCOUNT CARD.
00.40.39.*DIRECTIVE FORMAT 1 VERIFICATION DECK -
00.40.39.SORT/MERGE 4.0
00.40.39.*JOB SORTS 10 RECORDS - ASCENDING ORDER
00.40.41.FILE(INPUT,RT=Z,BT=C,FL=80,ERL=1)
00.40.42.FILE(OUTPUT,RT=Z,BT=C,FL=80,ERL=1)
00.40.43.SORTMRG(7C)
00.40.44.        1 NON-FATAL DIRECTIVE ERRORS
00.40.47. ** INSERTIONS DURING INPUT   *********0
00.40.47. ** DELETIONS DURING INPUT    *********0
00.40.47. ** TOTAL RECORDS SORTED      ********10
00.40.47. ** INSERTIONS DURING OUTPUT  *********0
00.40.47. ** DELETIONS DURING OUTPUT   *********0
00.40.47. ** TOTAL RECORDS OUTPUT      ********10
00.40.47. **END SORT RUN
00.40.48.*DIRECTIVE FORMAT 2 VERIFICATION DECK -
00.40.48.SORT/MERGE 4.0
00.40.48.*JOB SORTS 10 RECORDS - ASCENDING ORDER
00.40.48.SORTMRG(6C)
00.40.52. ** INSERTIONS DURING INPUT   *********0
00.40.52. ** DELETIONS DURING INPUT    *********0
00.40.52. ** TOTAL RECORDS SORTED      ********10
00.40.52. ** INSERTIONS DURING OUTPUT  *********0
00.40.52. ** DELETIONS DURING OUTPUT   *********0
00.40.52. ** TOTAL RECORDS OUTPUT      ********10
00.40.52. **END SORT RUN
00.40.52.*MACRO SORT VERIFICATION DECK -
00.40.52.SORT/MERGE 4.0
00.40.52.*JOB SORTS 10 RECORDS - ASCENDING ORDER
00.40.52.COMPASS,S=SMTEXT,S=IOTEXT.
00.41.19.    3.765  CPU SECONDS  ASSEMBLY TIME.
00.41.19. ASSEMBLY COMPLETE.   53200B  CM USED.
00.41.19.LDSET(LIB=COBOL/SYSIO)
00.41.19.LGO.
00.41.27.
00.41.27.
00.41.28. ** INSERTIONS DURING INPUT   *********0
00.41.28. ** INSERTIONS DURING INPUT   *********0
00.41.28. ** TOTAL RECORDS SORTED      ********10
00.41.28. ** INSERTIONS DURING OUTPUT  *********0
00.41.28. ** DELETIONS DURING OUTPUT   *********0
00.41.28. ** TOTAL RECORDS OUTPUT      ********10
00.41.28. **END SORT RUN
```

TIME-SHARING FORTRAN 1.0 14

14.1 RELEASE DESCRIPTION

14.1.1 HARDWARE REQUIREMENTS

Time-Sharing FORTRAN can be maintained on the same minimum hardware configuration as KRONOS 2.1. The minimum field length to run Time-Sharing FORTRAN is $44,000_8$.

14.1.2 CORRECTIONS

Time-Sharing FORTRAN 1.0 contains corrective code through KRONOS 2.0 level 12 PSR release.

14.1.3 DEFICIENCIES

None.

14.2 INSTALLATION PROCEDURE

The procedure to install and update Time-Sharing FORTRAN is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for Time-Sharing FORTRAN, the following materials are necessary.

KPL2 contains the jobs to update, install, and verify installation; the jobs required for Time-Sharing FORTRAN are TSR1, TSR2, and TSR3.

TSR1	Use to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
TSR2	Use to collect the Time-Sharing FORTRAN binary code and generate directive files for later use as input to the procedure file GENSYS.
TSR3	Use to verify that the product is correctly installed.

KPL20 contains the Time-Sharing FORTRAN binary and source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.14.

14.3 INSTALLATION PARAMETERS

None.

14.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1.

14.4.1 TSR1

```
TSR1
TSR1,CM62000,P30,T1000.
*ACCOUNT CARD.
*THIS JOB CREATES A NEW PROGRAM LIBRARY
*AND BINARIES OF TSRUN 1.0 ON KPL20.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(TSRUNPL,R,FI=TSRUN1P0*2P1,MT,D=HY,F=I)
COPYBF,TSRUNPL,RPL.
*GET,MOD=TSRMOD.
UNLOAD,TSRUNPL.
RETURN,TSRUNPL.
MODIFY,N,Q.
CATALOG,NPL,R.
REWIND,LGO.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL20,W,FI=TSRUN1P0*2P1,MT,D=HY,F=I)
COPYBF,NPL,KPL20.
COPY,LGO,KPL20.
REWIND,KPL20,NPL,LGO.
VERIFY,NPL,KPL20.
VERIFY,LGO,KPL20.
REWIND,KPL20.
*WEOR
*OPLFILE  RPL
*READ     MOD,*
OD,*
*EDIT     VOMCFTE.RETURN
*WEOF
```

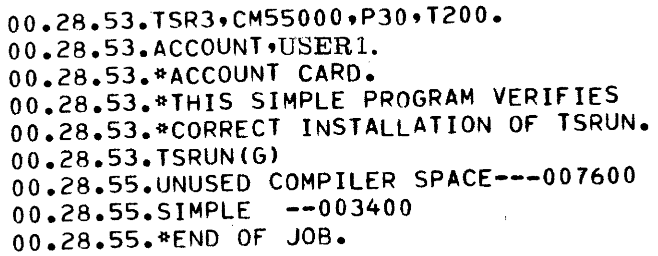
14.4.2 TSR2

```
TSR2
TSR2,CM15000,P30,T1000.
*ACCOUNT CARD.
*VSN CARD FOR AUTO TAPE ASSIGNMENT.
LABEL(KPL20,R,FI=TSRUN1P0*2P1,MT,D=HY,F=I)
SKIPF,KPL20.
COPYBF,KPL20,TSRUN.
UNLOAD,KPL20.
RETURN,KPL20.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBR,,ZZZZLDF.
CATALOG,TSRUN,R.
COPYBF,TSRUN.ZZZPSBF.
BKSP,ZZZPSBF.
*WEOR
*ADD      LIB31,OVL/TSRUN-COS/RETURN
*WEOF
```

14.4.3 TSR3

```
TSR3
TSR3,CM55000,P30,T200.
*ACCOUNT CARD.
*THIS SIMPLE PROGRAM VERIFIES
*CORRECT INSTALLATION OF TSRUN.
TSRUN(G)
*END OF JOB.
EXIT.
*JOB FAILED.
*WEOR
00010       PROGRAM    SIMPLE (OUTPUT)
00020       REAL      A(1,10), B(10,2), C(1,2), D(1,2)
00030       DATA      D(1,1), D(1,2)  / 385.0, 5885.0 /
00040       DO  10  I=1, 10
00050         A(1,I) = I
00060    10 CONTINUE
00070       DO  30  J=1, 2
00100         DO  20  I=1, 10
00110           B(I,J) = I + 100*(J - 1)
00120    20   CONTINUE
00130    30 CONTINUE
00140       DO  60  I=1, 1
00150         DO  50  J=1, 2
00160           T = 0.0
00170           DO  40  K=1, 10
00200             T = T + A(I,K)*B(K,J)
00210    40     CONTINUE
00220           C(I,J) = T
00230    50   CONTINUE
00240    60 CONTINUE
00250       IF (C(1,1) .NE. D(1,1))  GO TO  70
00260       IF (C(1,2) .NE. D(1,2))  GO TO  70
00270       CALL REMARK(23H1COMPUTATION SUCCESSFUL)
00300       STOP
00310    70 PRINT 1000, C, D
00320 1000  FORMAT (*1    FAILURE*/*0 C-ARRAY IS *2F20.5/*0  SHOULD BE *
00330+2F20.5)
00340       CALL REMARK (7HFAILURE)
00350       CALL EXIT
00360       END
*WEOF
```

Dayfile information from running the installation verification program should appear similar to the following.



```
00.28.53.TSR3,CM55000,P30,T200.
00.28.53.ACCOUNT,USER1.
00.28.53.*ACCOUNT CARD.
00.28.53.*THIS SIMPLE PROGRAM VERIFIES
00.28.53.*CORRECT INSTALLATION OF TSRUN.
00.28.53.TSRUN(G)
00.28.55.UNUSED COMPILER SPACE---007600
00.28.55.SIMPLE  --003400
00.28.55.*END OF JOB.
```

15.1 RELEASE DESCRIPTION

The released version of KRONOS Transaction Subsystem 1.0 consists of three parts. These parts are listed along with their OPL deck names. All deck names are on a single Modify OPL.

Programs	OPL Deck Names
Transaction Subsystem 1.0 programs that interface with KRONOS 2.1	COMBACM through TRANSIM
User libraries to link user's programs to the Transaction Subsystem. These user libraries are:	
BDMI user library	BDMI through TSIM
TRANLIB user library	CALLTSK through TSIM
Task library containing four Transaction Subsystem tasks to interface with other user-developed tasks.	ITASK through OFFTASK

15.1.1 HARDWARE REQUIREMENTS

Transaction Subsystem can be maintained on the same minimum hardware configuration as does KRONOS 2.1. To run the Transaction Subsystem, the Time-Sharing Executive must be active. The minimum field length requirements for the Transaction Subsystem, excluding the Time-Sharing Executive, is $24,000_8$.

15.1.2 CORRECTIONS

The Transaction Subsystem is a new product; therefore, there are no corrections.

15.1.3 DEFICIENCIES

Journalling to tape is not implemented in the released system.

15.2 INSTALLATION PROCEDURE

The procedure to install, modify, and update Transaction Subsystem is in part I, section 3.1 (General Procedure) and section 3.2 (Procedure to Create a New Deadstart Tape). To use these procedures for the Transaction Subsystem, the following materials are necessary.

KPL1 contains the KRONOS 2.1 program library.

KPL2 contains the jobs to modify, update, and install; the jobs required for Transaction Subsystem are KTS1 and KTS2.

KTS1	Use to modify installation parameters or to add PSR updates to source code and to create a new tape (current KPL) containing updated source and binary code.
KTS2	Use to collect the binary code and generate directive files for later use as input to the procedure file GENSYS.

KPL21 contains Transaction Subsystem binary and source code.

The tapes are described in detail in part I, sections 2.1.1 and 2.15.

When using the procedures mentioned above to install the Transaction Subsystem, note the following additional requirements.

1. When creating VALIDUX, assign a user index, a user number and a password to the Transaction Subsystem.
2. When modifying installation parameters, include these assignments in KTS1 by modifying the three parameters described in part III, section 15.3.4.
3. After the Transaction Subsystem is installed on the deadstart tape, but before it is used, create a task library permanent file containing the four tasks required by the Transaction Subsystem. These tasks, which are on the release tape KPL21, are:

ITASK	Task supervisor
KDIS	TRANEX K display driver
MSABT	Message sender for abnormally terminating tasks
OFFTASK	Inactive task controller

To create a task library containing these tasks, run a job similar to the following.

```
JOB,CM55000.
ACCOUNT card.                    Use the user number and password assigned
                                 to the Transaction Subsystem previously.
VSN card for KRONOS OPL
LABEL(KPL1,R,FI=KRONOS*2P1,MT,D=HY,F=I)
COPYEI (KPL1,OPL)
```

```
UNLOAD (KPL1)
RETURN (KPL1)
VSN card for KPL21
LABEL (KPL21,R,FI=TRANSACTSYS*2P1,MT,D=HY,F=I)
                                         Transaction Subsystem OPL
COPYEI (KPL21,KTSPL)
UNLOAD (KPL21)
RETURN (KPL21)
MODIFY (LO=E)
COMPASS(I, L=0)
DEFINE (TASKLIB/CT=PU)
PERMIT (TASKLIB,usernumber=W) Usernumber must be same as specified
                              on the ACCOUNT card.
LIBTASK,CR.
7/8/9
*OPLFILE    KTSPL
*EDIT       ITASK,FDIS,MSABT,OFFTASK
7/8/9
*ITASK      S.
6/7/8/9
```

15.3 INSTALLATION PARAMETERS

The following parameters may be changed in deck TRANEX to fit the requirements of a specific installation. The parameters preceded by an asterisk (*) are critical scheduling values and should be changed with caution.

15.3.1 COMPILE TIME CONSTANTS

Parameter and Released Default Value			Significance
*TMDTL	EQU	100B	Milliseconds between time dependent loop routines; TRANEX 571.
KDSTL	EQU	1000B	Milliseconds between K-display updates; TRANEX 572.
*RCLTL	EQU	20B	Milliseconds for task recall; TRANEX 573.
*SCHTL	EQU	60B	Milliseconds between timed scheduler runs; TRANEX 574.

Parameter and Released Default Value			Significance
CORTL	EQU	1000	Milliseconds between core usage checks; TRANEX 575.
TSKTL	EQU	10	Maximum number of time slices per task; TRANEX 585.
TSL	EQU	200	Exchange jump function time slice; TRANEX 581.
MAXSM	EQU	63+1	Maximum number of words of journal; TRANEX 587.
FCML	EQU	100B	Free core memory reduce limit; TRANEX 590.
MAXBW	EQU	3	Maximum task branch count; TRANEX 591.
MAXRA	EQU	10000	Number of legal RA+1 requests; TRANEX 592.

The following millisecond counts may exceed 131,000 and therefore require a word to contain their value.

Parameter and Released Default Value			Significance
TACTL	CON	2*60*1000	Milliseconds between activity checks; TRANEX 580.
SJTTL	CON	20*60*1000	Milliseconds between periodic journalling; TRANEX 581.
ITRTL	CON	1500D	Milliseconds to idle before rolling out; TRANEX 582 .
TROTL	CON	10*60*1000	Millisecond duration of rollout; TRANEX 583.

15.3.2 DSDUMP DEFAULT PARAMETERS

Parameter and Released Default Value			Significance
DFWA	EQU	100B	Default FWA for task dump; TRANEX 600.
DLWA	EQU	300B	Default LWA for task dump; TRANEX 602.
DEXP	EQU	1	Default exchange package dump; TRANEX 604.
DDMB	EQU	0	Default data manager buffer dump; TRANEX 606.
DORT	EQU	0	Default origin type for task; TRANEX 608.
DORC	EQU	BCOT	Default origin code; TRANEX 610.
DQDS	EQU	0	Default queue destination value; TRANEX 611.

15.3.3 INITIALIZATION TIME OF K-DISPLAY OPTION

Parameter and Released Default Value			Significance
NSCP	EQU	12	Number of subcontrol points; TRANEX 622.
MINSCP	EQU	2	Minimum number of subcontrol points; TRANEX 624.
NCMB	EQU	4	Number of communication blocks; TRANEX 631.
MINCMB	EQU	4	Minimum number of communication blocks; TRANEX 633.
MAXCMB	EQU	40	Maximum number of communication blocks; TRANEX 634.
MAXCMBM	DECMIC	MAXCMB	Maximum number of C.B.S.; TRANEX 637.
SCMFL	EQU	100000B	Default maximum field length for TRANEX: TRANEX 642.

Parameter and Released Default Value			Significance
MINRFL	EQU	3000B	Minimum amount of FL to request; TRANEX 643.
IFL=	EQU	50000B	Initial field length; TRANEX 644.
MINMFL	EQU	40000B	Minimum limit of a maximum running field length; TRANEX 647.
MAXMFL	EQU	300000B	Maximum limit for running field length; TRANEX 648.
IMDM	EQU	2	Number of sets of data manager buffers; TRANEX 653.
MINMDM	EQU	1	Minimum for multiple value; TRANEX 655.
MAXMDM	EQU	4	Maximum for multiple value; TRANEX 656.

15.3.4 DEFAULT USER NUMBER/PASSWORD AND USER INDEX

Change the following parameters to correspond to the user number, password, and user index assigned to the Transaction Subsystem.

Parameter and Released Default Value			Significance
USNM	MICRO	1,7,/KB100DC/	TRANEX user number; TRANEX 557.
PWDM	MICRO	1,,//	TRANEX password; TRANEX 558.
TRUI	EQU	16B	User index for Transaction Subsystem; TRANEX 559.

15.4 INSTALLATION JOBS

The following jobs are included to indicate the format of the released installation jobs. They are released with the initial release of KRONOS 2.1 and may be modified in subsequent updates of KRONOS 2.1

15.4.1 KTS1

```
KTS1
KTS1,CM62000,T2000.
*ACCOUNT,CARD.
COMMENT. THIS JOB MODIFIES AND CREATES THE KRONOS TRANSACTION
COMMENT. SUBSYSTEM BINARY AND USER LIBRARIES. THE FIRST FILE OF THE NEW
COMMENT. KPL21 WILL BE THE NEWPL. THE SECOND FILE WILL BE THE
COMMENT. ABSOLUTE BINARY OF THE SUBSYSTEM AND USER LIBRARIES.
COMMENT. THE THIRD FILE WILL BE THE RELOCATABLE BINARY OF THE
COMMENT. LIBRARY ROUTINES.
COMMENT. THE KRONOS OPL IS ASSUMED TO BE STAGED AS *OPL*.
COMMON(OPL)      KRONOS OPL.
*VSN CARD.
LABEL(KTSPLIN,R,FI=TRANSACTSYS*2P1,MT,D=HY,F=I)      LATEST KPL21.
ATEST KPL21.
COPYBF,KTSPLIN,KTSPL.
VERIFY,KTSPLIN,KTSPL,R.
UNLOAD,KTSPLIN.
RETURN,KTSPLIN.
MODIFY,N,LO=E,C=0.            CREATE NEW PROGRAM LIBRARY.
MODIFY,LO=E,NR.
MODIFY,LO=E,NR.
REWIND,COMPILE.
COMPASS,I,L=0.                KTS SUBSYSTEM PROGRAMS.
COMPASS,I,L=0,B=LIB.          KTS LIBRARY ROUTINES.
GTR(LIB,TRLIB)REL/CALLTSK-TSIM
LIBGEN,F=TRLIB,P=TRANLIB.
LIBGEN,F=LIB,P=BDMLIB.
REWIND,LGO,LIB,TRANLIB,BDMLIB.
COPYBF,LGO,KTS.
BKSP,KTS.
COPYBF,TRANLIB,KTS.
BKSP,KTS.
COPYBF,BDMLIB,KTS.
*VSN CARD.
LABEL(KPL21,W,FI=TRANSACTSYS*2P1,MT,D=HY,F=I)    SCRATCH FOR NEW KPL21.
REWIND,KTS,NPL.
COPYBF,NPL,KPL21.
COPYBF,KTS,KPL21.
COPYBF,LIB,KPL21.
REWIND,KPL21,NPL,KTS,LIB.
VERIFY,KPL21,NPL.
VERIFY,KPL21,KTS.
VERIFY,KPL21,LIB.
UNLOAD,KPL21.
*WEOR
*OPLFILE  KTSPL
*/        PLACE *TRANSACTION SUBSYSTEM* MODSET HERE.
*EDIT     COMBACM.OFFTASK
```

```
OMBACM.OFFTASK
*WEOR
*OPLFILE  NPL
*/        EDIT KTS SUBSYSTEM PROGRAMS.
*EDIT     CS1.TRANSIM
*WEOR
*OPLFILE  NPL
*/        EDIT KTS USER LIBRARY ROUTINES.
*EDIT     BDMI.TSIM
DMI.TSIM
*WEOF
```

15.4.2 KTS2

```
KTS2
KTS2,CM45000,T1000.
*ACCOUNT CARD.
COMMENT. THIS JOB ADDS THE TRANSACTION SUBSYSTEM BINARIES TO THE
COMMENT. SYSTEM COMMON FILE ZZZPSBF FROM KPL21. EITHER THE RELEASED
COMMENT. VERSION OR THE LATEST VERSION CREATED BY JOB KTS1 MAY BE USED.
COMMENT. IT ALSO ADDS THE NECESSARY DIRECTIVES TO COMMON FILE ZZZZLDF.
*VSN CARD.
LABEL(KPL21,R,FI=TRANSACTSYS*2P1,MT,D=HY,F=I)    LATEST KPL21.
SKIPF,KPL21,1.
COMMON,ZZZZLDF,ZZZPSBF.
COPYBF,KPL21,ZZZPSBF.
BKSP,ZZZPSBF.
COPYBR,,ZZZZLDF.
UNLOAD,KPL21.
*WEOR
*ADD      LIB9,PP/CS1-ULIB/BDMLIB
*WEOF
```

USER VALIDATION 1

The user validation capability of the KRONOS Time-Sharing System is based on two special system files, VALIDUS and PROFILO. These files are used to define and control:

- who can use the system
- what they can use (hardware and software)
- to what extent they can use it

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 PROFILO 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 PROFILO can establish master users who may add, modify, and delete charge and project admissions for their 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 account file† is created and managed by MODVAL, the account file manager. The PROFILO project profile file†† is created and managed by PROFILE, the project profile manager. 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.

†VALIDUS is referred to as the account file since it contains the account numbers of users requesting admission to the system. It should not be confused with the account dayfile.

††PROFILO is referred to as the project profile file since the project validations it contains form a usage profile for each account (user) number.

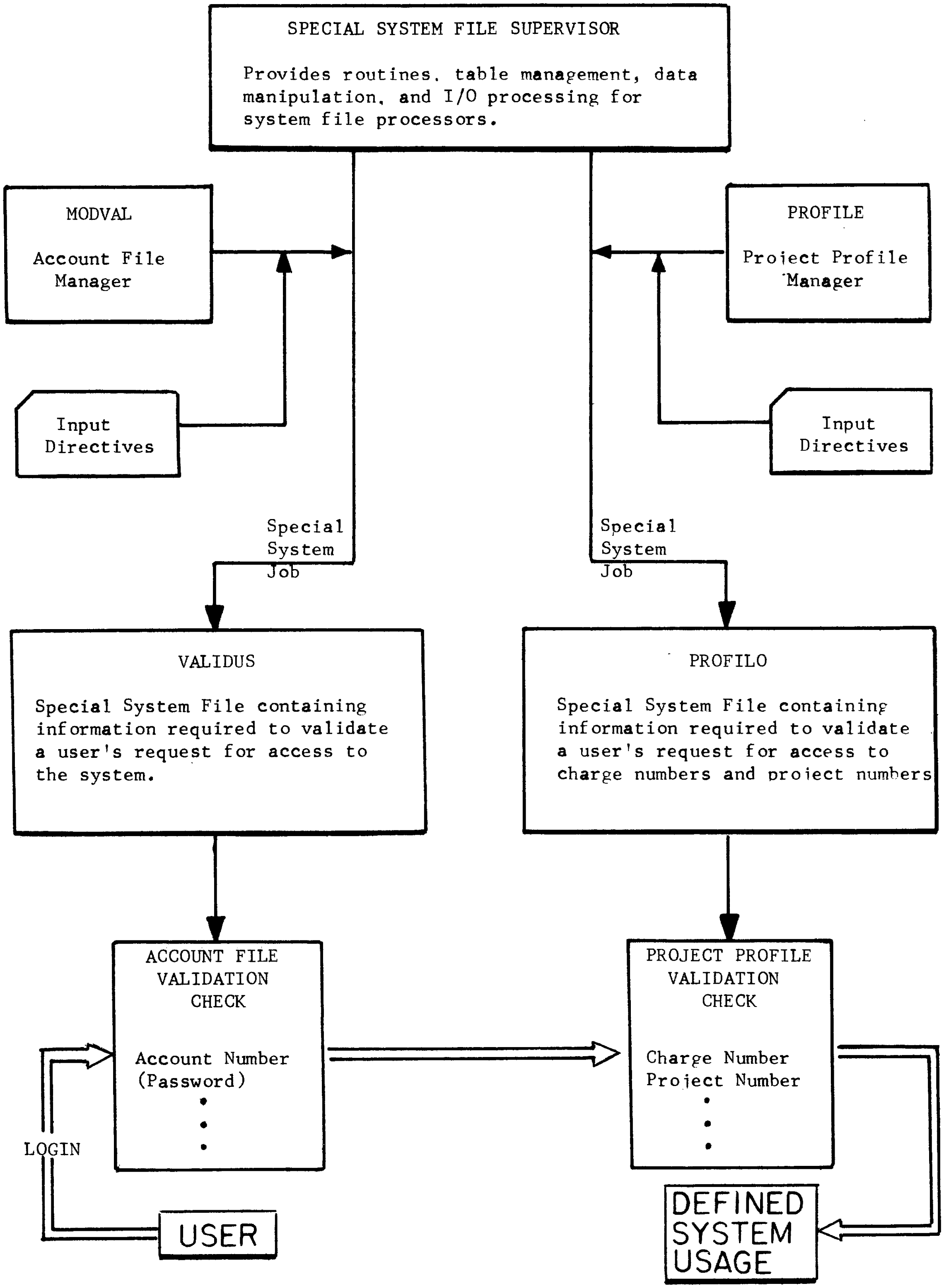

SPECIAL SYSTEM FILE SUPERVISOR
Provides routines, table management, data manipulation, and I/O processing for system file processors.
MODVAL
Account File Manager
PROFILE
Proiect Profile Manager
Input Directives
Input Directives
Special System Job
Special System Job
VALIDUS
Special System File containing information required to validate a user's request for access to the system.
PROFILO
Special System File containing information required to validate a user's request for access to charge numbers and proiect numbers
ACCOUNT FILE VALIDATION CHECK
Account Number (Password)
PROJECT PROFILE VALIDATION CHECK
Charge Number
Proiect Number
LOGIN
USER
DEFINED SYSTEM USAGE

Figure IV-1-1. Creation and Use of Validation Files

1.1 VALIDUS (VALIDUX)† SPECIAL SYSTEM FILE

The VALIDUS special system file contains the account numbers that validate individual user access to the KRONOS Time-Sharing System. A second level of security, the password, can also be specified. In addition, VALIDUS defines the extent to which each account can make use of the system's resources. These specifications include:

• Equipment usage	Maximum number of magnetic tapes this account may use
	Maximum number of disk packs this account may use
	Maximum number of mass storage tracks (for future use)
	On what master device the permanent files and catalog for each account will be stored
• File usage	Maximum number of permanent files allowed
	Maximum number of files while running
	Ability to create direct access permanent files
	Ability to create indirect access permanent files
	Maximum size of indirect access permanent files or cumulative size of all indirect access permanent files
• Machine usage	Maximum CPU time
	Maximum CM 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

Priorities and the number of deferred batch jobs can also be specified.

† The term VALIDUX is used with KRONOS 2.1 release. This text uses VALIDUS, the term associated with 2.0.

1.1.1 MODVAL ACCOUNT FILE MANAGER

The account file manager, MODVAL, is used to create and manage the VALIDUS account file. This can be done either from the system console (system origin job) or by batch.

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, accnumb — Inquire option relative to the user identified by the account number accnumb. Information is given on the second MODVAL display (Figure 1-3). This option cannot be used for data entry.

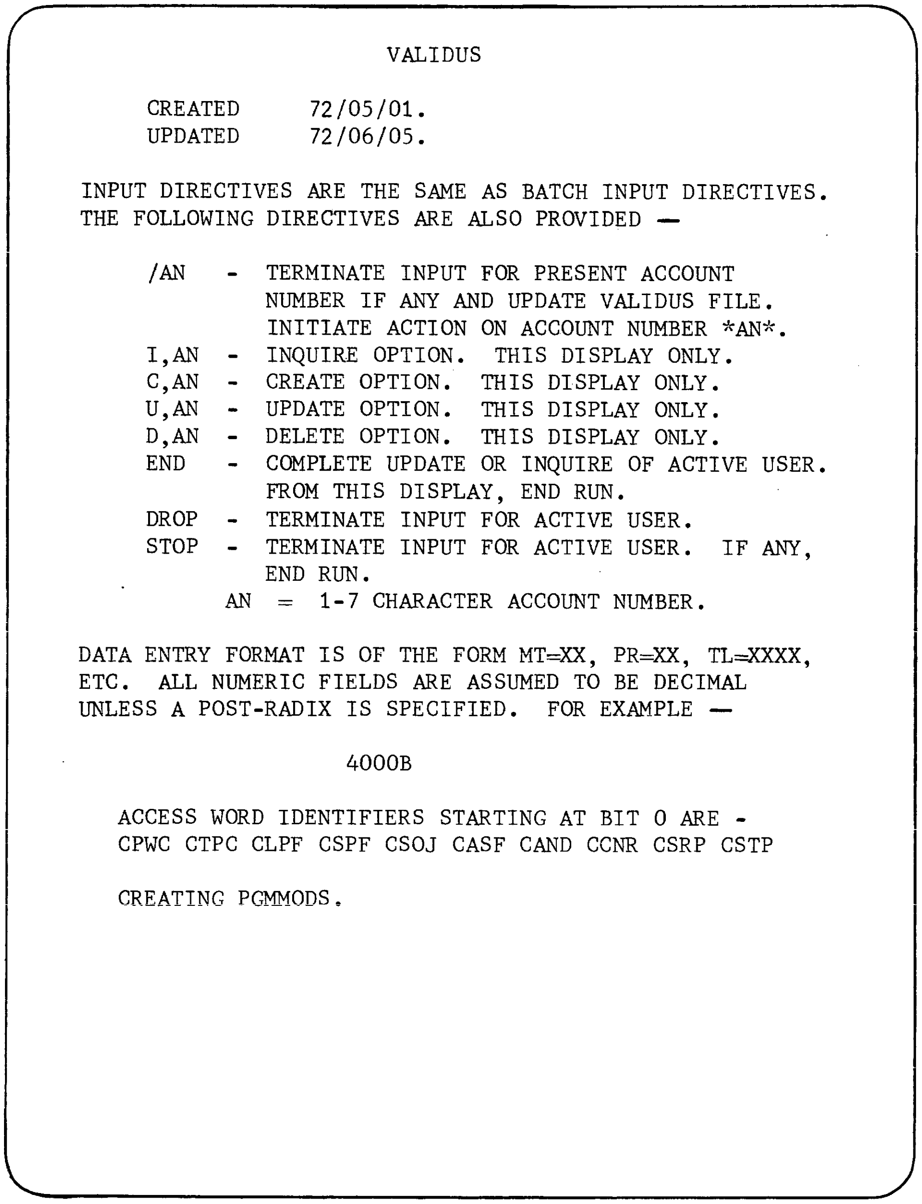
K.U, accnumb — Update modification for the account number accnumb which is on the existing VALIDUS file. Modification data is input following the input of this directive.

K.C, accnumb — The account number accnumb is entered on a VALIDUS file that is being created. The following input line(s) can contain identifiers for this account number.

K.D, accnumb — The account number accnumb is deleted from the existing VALIDUS file.

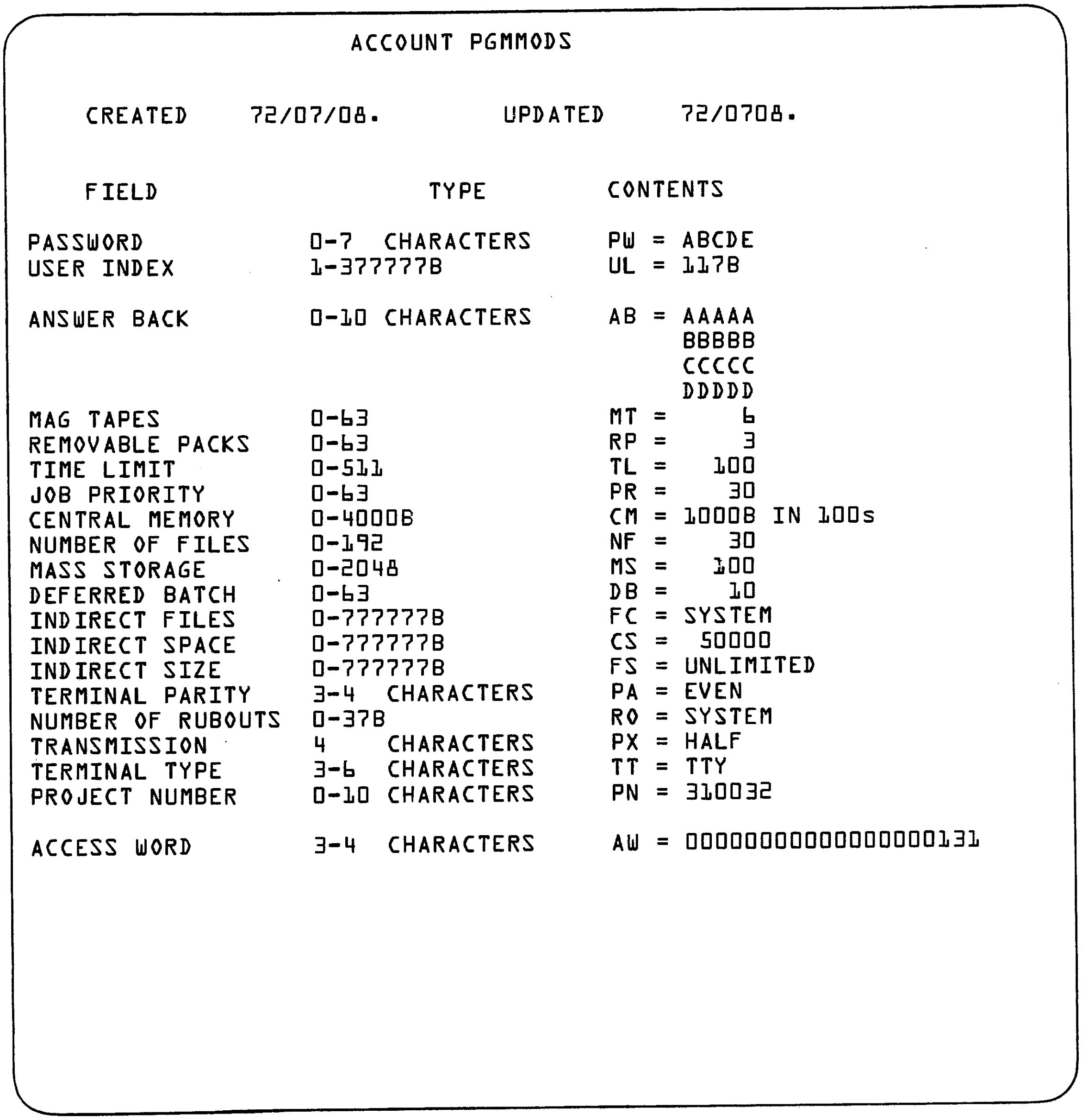
Information for the previous options is provided on two MODVAL displays (Figures IV-1-2 and IV-1-3) for each account number specification. For the option types, identifiers must be entered on separate lines following the account number specification.

K./accnumb, ident=data — This slash (/) directive is used to update accnumb as with the U option; however, if accnumb is not found, a new account number is created automatically. It is possible under this option to switch control directly from one account 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 account number and identifier on the same line.



```
                         VALIDUS

     CREATED     72/05/01.
     UPDATED     72/06/05.

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 ACCOUNT NUMBER *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.
     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=XXXX,
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

   CREATING PGMMODS.
```

Figure IV-1-2. First MODVAL Display



```
                          ACCOUNT PGMMODS

    CREATED     72/07/08.         UPDATED     72/0708.

    FIELD                 TYPE          CONTENTS

PASSWORD            0-7  CHARACTERS     PW = ABCDE
USER INDEX          1-377777B           UL = 117B

ANSWER BACK         0-10 CHARACTERS     AB = AAAAA
                                             BBBBB
                                             CCCCC
                                             DDDDD
MAG TAPES           0-63                MT =     6
REMOVABLE PACKS     0-63                RP =     3
TIME LIMIT          0-511               TL =   100
JOB PRIORITY        0-63                PR =    30
CENTRAL MEMORY      0-4000B             CM = 1000B IN 100s
NUMBER OF FILES     0-192               NF =    30
MASS STORAGE        0-2048              MS =   100
DEFERRED BATCH      0-63                DB =    10
INDIRECT FILES      0-777777B           FC = SYSTEM
INDIRECT SPACE      0-777777B           CS =  50000
INDIRECT SIZE       0-777777B           FS = UNLIMITED
TERMINAL PARITY     3-4  CHARACTERS     PA = EVEN
NUMBER OF RUBOUTS   0-37B               RO = SYSTEM
TRANSMISSION        4    CHARACTERS     PX = HALF
TERMINAL TYPE       3-6  CHARACTERS     TT = TTY
PROJECT NUMBER      0-10 CHARACTERS     PN = 310032

ACCESS WORD         3-4  CHARACTERS     AW = 00000000000000000131
```

Figure IV-1-3. Second MODVAL Display

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.

   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 account file VALIDUS (Figure IV-1-2).

5. The analyst types in one of the five console options. This is either

   K. option, accnumb CR

   or

   K. /accnumb, ident=data CR

   If a delete (K.D, accnumb) is entered, the account number accnumb is deleted from the account 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 account 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 account 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 account numbers and their associated parameters.

To terminate any run without erasure, the operator types

```
END. CR
```

Table IV-1-1 summarizes the basic input coding for console options.

TABLE IV-1-1. INPUT CODING FOR CONSOLE OPTIONS

Create	Update	Inquire	Delete
K. C, accnumb K. ident=data K. END	K. U, accnumb K. ident=data K. END	K. I, accnumb K. END	K. D, accnumb
K. /accnumb, ident=data K. END			

1.1.3 MODVAL BATCH INPUT

Batch jobs that call the MODVAL account 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 use 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 account file VALIDUS.
NEWVAL	The interim copy of the new account file that is to be created or reformatted.
VALIDUS	The old account file that is to be updated or reformatted.
SOURCE	Receives the source input for each account number.
VALINDX	Contains all the available user indices (definition in section 2. 1) for the present VALIDUS file. It is always used in conjunction with one of the account files: new=NEWVAL, old=VALIDUS.
OUTPUT	File to receive output listings.

For a batch create under MODVAL, two stages of operation are necessary.

1. Input at the card reader is used by MODVAL to create a local copy of a new account file. This copy is saved as an indirect or direct access permanent file.
2. From the console, the analyst gets or attaches the new account 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 card 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 account numbers with many identifiers is to be entered.

MODVAL CONTROL CARD

Batch input accesses the MODVAL account file manager by means of the MODVAL control card. The following is the format of the card.

MODVAL(p1, p2, . . . , pi, . . . , pn)

where the pi'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 account 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 account file. Default is NEWVAL.

S = source

File that will receive source data for each account number. Default is SOURCE.

U = valindx

File containing the available user indices for the current VALIDUS file. Default value is VALINDX.

L = output

File to receive list output. Default is OUTPUT.

CV

Convert VALIDUS option. Suppresses automatic creation of system and library user indices and allows the creation of two account numbers having the same user index during a create run. This allows source input from KRONOS 2.0 MODVAL to be used as input for a creation run under KRONOS 2.1 MODVAL. It is also used in the conversion from KRONOS 2.1 MODVAL to KRONOS 2.0 MODVAL to suppress the identifiers not used in KRONOS 2.0 (refer to example 13 in section 1.1.4).

D

If specified, MODVAL will not abort when directive errors are detected.

OP=C

Create option. Processes the input file and creates the interim account file (N-newval) and the file of associated user indices (U=valindx).

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 valindx 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

Card update option. This is like the update option except that directives are included on the MODVAL card image. The Z parameter in this option must be used alone.

OP=Z, LO=E

Initiates the card 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 account 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=I

Inquire option. Gives a listing of validation parameters for the account specified by the last ACCOUNT card or by terminal login. This option must occur alone. The following is a sample list.

```
 AB =,
 AB =,
 AB =,
 AB =,
MT =             4,
RP =             4,
TL =         7777B,
PR =           70B,
CM =         1000B,
NF =            20,
MS =           100,
 PN =,
DB =            10,
FC =        SYSTEM,
CS =        SYSTEM,
FS =        SYSTEM,
PA = EVEN         ,
RO =            31,
PX = HALF         ,
TT = TTY          ,
AW = 00000000000000000177
```

OP=L
or
OP=L, LO=A

Reads the validation file, sorts the copy by account 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 account number, and writes it to the output file for listing according to the format in Figure IV-1-4.

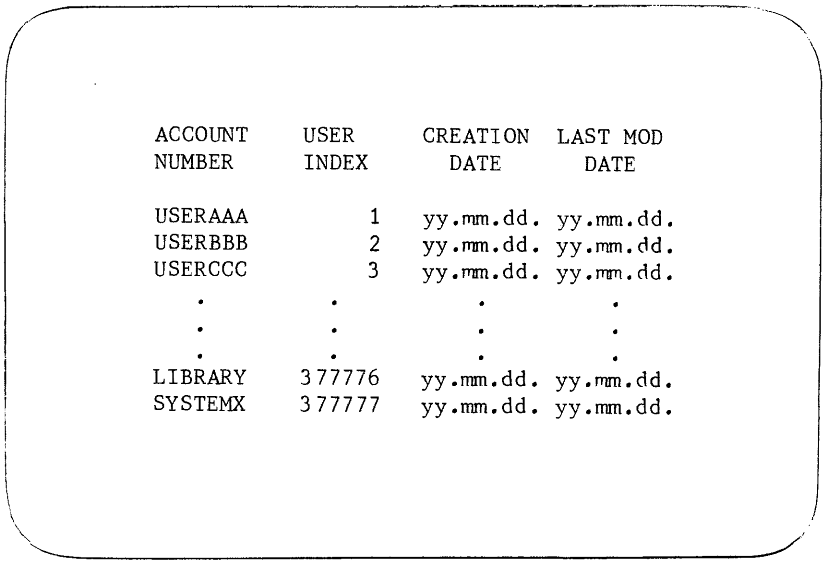
OP=L, LO=AL — Same as LO=L since A is a default value.

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.



```
ACCOUNT      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.
SYSTEMX     377777   yy.mm.dd.  yy.mm.dd.
```

Figure IV-1-4. Format of File Listing

INPUT DIRECTIVE

The input directive is used to enter account numbers under a create run (OP=C) and to modify existing account numbers under an update run (OP=U). Format of the input directive is

```
/accnumb,ident1=data1,ident2=data2,...
```

where accnumb is the one-to seven-character account number being referenced and each ident=data is a system usage definition for this number.

In addition to the standard separators (period, comma, parens, dollar-sign, etc.), end-of-card and end-of-line are recognized. Thus, if an input directive item occurs last on a card, none of the standard separators are needed.

Example: the following is acceptable.

```
/ROBERTR, AW=CSPF

AW=CLPF
```

However, data cannot lap from card to card.

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 one-to ten-character answerback code. Blanks are significant. The answerback code restricts the user to a particular terminal. Up to four answerback entries are permissible per account number. If this identifier is omitted, the system supplies an answerback code of all blanks which gives this user access through any terminal.
MT = nnr	This identifier specifies the number of magnetic tapes allowed. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B. If this identifier is omitted, the system supplies a value of 0.
RP = nnr	This identifier specifies the number of removable disk packs allowed. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B. If this identifier is omitted, the system supplies a value of 0.
TL = nnnnr	This identifier specifies the maximum CPU time (in units of 10 octal seconds) that a user may run. nnnnr consists of four numeric characters followed by a radix. Blanks are suppressed. The maximum value is 7777B. The default value is 7777B units of 10 octal seconds.
PR = nnr	This identifier specifies the maximum priority at which a user can run a job. (Normally, the job origin type determines the priority.) nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B. The default value is 12B.
CM = nnnnr	This identifier specifies the maximum central memory space a user is allowed in units of 100 octal words. nnnnr consists of four numeric characters followed by a radix. Blanks are suppressed. The maximum value is 4000B. The default value is 1000B.
NF = nnnnr	This identifier specifies the maximum number of files a user is allowed when running. nnnnr consists of four numeric characters followed by a radix. Blanks are suppressed. The maximum value is 300B. The default value is 24B. Because the code that checks this identifier can add significant overhead to the system, it is not implemented as part of this release.
MS = nnnnr	This identifier specifies the maximum number of mass storage tracks a user is allowed when running. nnnnr consists of four numeric characters followed by a radix. Blanks are suppressed. The maximum value is 4000B. (However, this is not in the system as yet.)

Identifier — Description

DB = nnr

This identifier indicates the number of deferred batch jobs the user is allowed to have in the system at one time. nnr consists of two numeric characters followed by a radix. Blanks are suppressed. The maximum value is 77B. The default value is one.

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 ten access bits defined in the system.

- CPWC (bit 0) User may change his password
- CTPC (bit 1) User may use the ACCESS commands
- 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
- CASF (bit 5) User may access system files (common)
- 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

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

The following identifiers can only be used in update and K display options.

- DAC Deletes the account number from the VALIDUS file
- FUI=nnnnnnr Forces the user index to be inserted or changed. Parameters are the same as for the UI=nnnnnnr

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 answerback changes are permitted per record.

PN = project number

A one-to ten-alphanumeric identifier available to the local site. No blank suppression. One entry per record is permitted.

The following three 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 permanent files allowed the user. n may assume the following values.

n	Upper Limit Allowed (Octal)
0	Use job origin control.
1	100
2	200
3	400
4	1000
5	5000
6	10000
7	Unlimited

If FC is not specified, 0 will be assumed.

CS = n

Cumulative size for all indirect access files for this user. n specifies the upper limit allowed.

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, 0 will be assumed.

FS = n

File size allowed for an individual indirect access permanent file. n may assume the following values.

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, 0 will be assumed.

The following four specifications manipulate fields describing the user's terminal.

Identifier — Description

PX = xxxx

Specifies transmission mode. Only one entry should occur per account number record. Since the terminal operates in full or half duplex mode, either of the following values are available for xxxx.

Value	Explanation
FULL	System enters echoplex mode automatically.
HALF	System does not enter echoplex mode automatically.

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 account 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 account number record.

EVEN

ODD

TT = xxxxxx

xxxxxx specifies the terminal type. One of the following values may be specified for each account 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

1.1.4 PASSWOR CONTROL CARD

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 card. The format of the card is

```
PASSWOR(oldpass,newpass)
```

The passwords are one to seven characters.

PASSWOR ERROR MESSAGES

ERROR IN PASSWOR ARGUMENTS.

Invalid control card arguments.

Action: Correct and rerun.

ILLEGAL CONTROL CARD.

User is not permitted to change password.

Action: User must acquire validation.

1.1.5 LIMITS CONTROL CARD

Each user can obtain a listing of all the limiting parameters that apply to his user number by means of the LIMITS control card. MODVAL processes this card exactly as the OP=I option. The format of the card is

```
LIMITS.
```

The explanation of the inquire option under MODVAL Control Card (section 1.1.3) gives a sample listing.

LIMITS ERROR MESSAGE

ERROR IN LIMITS ARGUMENTS.

Invalid control card arguments. This card has no arguments associated with it.

1.1.6 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 377777.

Example 1:

Example 1 is a create at the console with the C,accnumb 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 account numbers with a password ident for each.

```
X.MODVAL.

K,n.

K.C,ACCN201

K.PW=ADMIT1

K.END

K.C,ACCN202

K.PW=ADMIT2

K.END

K.C,ACCN203

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./ACCN201,PW=ADMIT1

K./ACCN202,PW=ADMIT2

K./ACCN203,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 card input.

```
JOBCR1.

ACCOUNT(ANLST)

MODVAL(OP=C)

SAVE(NEWVAL)
SAVE(VALINDX=VAL)
7/8/9
/ACCN201,PW=ADMIT1
/ACCN202,PW=ADMIT2
/ACCN203,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
ACCOUNT,ANLIST.
     or SUI,xxxx. where xxxx is the user index for ANALST.
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,VALINDX).
DEFINE,VALIDUS.
DEFINE,VALINDX.
COPY,NEWVAL,VALIDUS,V.
COPY,VAL,VALINDX,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
/ACCN201,PW=ADMIT1
/ACCN202,PW=ADMIT2
/ACCN203,PW=ADMIT3
```

Input at the card reader is:

```
JOBCR2.

ACCOUNT,ANLST.

GET,PUTIN.

MODVAL,OP=C,I=PUTIN,N=VALNEW.

SAVE,VALNEW.

SAVE,VALINDX=VALX.

6/7/8/9
```

After this job is executed, the following entries are made at the console.

```
X.DIS.

ACCOUNT,ANLST.

     or SUI,xxxx. where xxxx is the user index for ANALST.

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,VALINDX).

DEFINE,VALIDUS.

DEFINE,VALINDX.

COPY,VALNEW,VALIDUS,V.

COPY,VALX,VALINDX,V.
```

Example 5:

Example 5 is an update at the console with the U,accnumb format. The first two account numbers entered via the previous creates have their password changed.

```
X.MODVAL.

K,20.

K.U,ACCN201
```

```
K.PW=ENTER1

K.END

K.U,ACCN202

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./ACCN201,PW=ENTER1

K./ACCN202,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 VALINDX are copied to permanent files (direct or indirect) that can be accessed by the batch input and used in the MODVAL control card. This is done at the console as follows:

```
X.DIS.

SUI,377777.

ISF(R=VALIDUS)

ATTACH(VALIDUS, VALINDX)

COPY(VALIDUS, VAL)

COPY(VALINDX, VALX)

RETURN(VALIDUS, VALINDX)

ISF.

ACCOUNT(ANLST)

SAVE(VAL, VALX)
```

The batch input is:

```
JOBUP.

ACCOUNT(ANLST)

GET(VAL, VALX)

MODVAL(OP=U, P=VAL, U=VALX)

REPLACE(VAL, VALX)

7/8/9

/ACCN201, PW=ENTER1

/ACCN202, 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.

ACCOUNT(ANLST)

GET(VAL, VALX)

SUI, 377777.

ISF(R=VALIDUS)

ATTACH(VALIDUS, VALINDX/M=W)

COPY(VAL, VALIDUS)

COPY(VALX, VALINDX)

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)/ACCN201, PW=ENTER1

     (one account at a time)
```

Example 8:

In example 8, a delete is done from the console only.

```
X.MODVAL
K,20.
K.D,ACCN203
K.END.
```

Example 9:

In example 9, reformatting of the account file is initiated from the console. The DIS is used and card images 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,VALINDX/M=W)
MODVAL(OP=R)
REWIND(VALIDUS,NEWVAL)
COPY(NEWVAL,VALIDUS)
OUT.
ISF.
```

Example 10:

In example 10, to reformat the account 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,VALINDX)
COPY(VALIDUS,VAL)
COPY(VALINDX,VALX)
RETURN(VALIDUS,VALINDX)
```

```
ISF.
ACCOUNT(ANLST)
SAVE(VAL, VALX)
```

Then, from the card reader:

```
JOBREF.
ACCOUNT(ANLST)
GET(VAL, VALX)
MODVAL(OP=R, P=VAL, U=VALX)
SAVE(NEWVAL)
REPLACE(VALX)
6/7/8/9
```

Then, from the console:

```
X.DIS.
ACCOUNT(ANLST)
GET(NEWVAL, VALX)
SUI,377777.
ISF(R=VALIDUS)
ATTACH(VALIDUS, VALINDX/M=W)
COPY(NEWVAL, VALIDUS, V)
COPY(VALX, VALINDX, V)
ISF.
```

Instead of the ATTACH, in which each COPY would write over an old file, it would be possible to use

```
PURGE(VALIDUS, VALINDX)
DEFINE(VALIDUS, VALINDX/M=W)
```

and then copy onto the empty files.

Example 11:

In example 11, the account file is returned to source code via the console.

```
X.DIS.

SUI,377777.

ISF(R=VALIDUS)

ATTACH(VALIDUS,VALINDX)

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 account file to source code via batch, the following procedure is followed.

From the console:

```
X.DIS.

SUI,377777.

ISF(R=VALIDUS)

ATTACH(VALIDUS,VALINDX)

COPY(VALIDUS,VAL)

COPY(VALINDX,VALX)

RETURN(VALIDUS,VALINDX)

ISF.

ATTACH(VALIDUS,VALINDX)

COPY(VALIDUS,VAL)

COPY(VALINDX,VALX)

ACCOUNT(ANLST)

SAVE(VAL,VALX)
```

From the card reader:

```
JOBSRS

ACCOUNT(ANLST)

GET(VAL, VALX)

MODVAL(OP=S, P=VAL, U=VALX)

SAVE(SOURCE)
```

From the console:

```
X.DIS.

ACCOUNT(ANLST)

GET(SOURCE)

SUI, 377777.

SAVE(SOURCE)
```

Example 13:

In example 13, a KRONOS 2.0 VALIDUS file is converted to KRONOS 2.1 format. The analyst first deadstarts a KRONOS 2.0 system and then enters the following sequence of commands at the console.

```
X.DIS.

SUI(377777)

GET(VALIDUS)

DEFINE(SOURCE)

MODVAL(OP=S, P=VALIDUS)

DROP.
```

When the VALIDUS file is successfully converted to source, the analyst should deadstart a KRONOS 2.1 system. After bringing up a DIS package, the following sequence of commands should be entered.

```
X.DIS.

SUI(377777)

ATTACH(INPUT=SOURCE)

PURGE(VALIDUS, VALINLK, SOURCE)
```

```
DEFINE(VALIDUS, VALINDX)

MODVAL(OP=C,N=VALIDUS,CV)

DROP.

X. ISF.
```

The CV parameter allows use of a source dump of VALIDUS to create a new 2.1 VALIDUS file.

If it is desired to reverse the conversion sequence (that is, starting from 2.1 and converting to 2.0), the CV parameter should be used on the 2.1 MODVAL call to dump to SOURCE to suppress 2.1 source identifiers not recognizable by KRONOS 2.0 MODVAL. After a KRONOS 2.1 deadstart, the following sequence of commands should be entered.

```
X.DIS.

SUI(377777)

ISF(R=VALIDUS)

ATTACH(VALIDUS)

DEFINE(SOURCE)

MODVAL(OP=S,CV)

DROP.
```

Then, the analyst should deadstart a KRONOS 2.0 system and enter the following commands.

```
X.DIS.

SUI(377777)

ATTACH(INPUT=SOURCE)

PURGE(VALIDUS, VALINDX,SOURCE)

MODVAL(OP=C)

SAVE(NEWVAL=VALIDUS)

DEFINE(VALINLK)

DROP.

X. ISF.
```

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 card 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.

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 ACCOUNT NUMBER.

Illegal data was encountered where account number was expected.

Action: Illegal data is disregarded and MODVAL goes to the next account entry.

**** DUPLICATE ACCOUNT NUMBER.

The account number encountered is a duplicate of an account 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 account 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 account 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 account 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 account number can be assigned to any user index with the UI identifier.

Action: This account 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.

**** 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 account 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 account file can be reformatted to eliminate block linkage, searches will be faster for account numbers residing in linked blocks and nonexistent account numbers which would have resided in linked blocks.

CREATING account.

User number account is being created.

UPDATING account.

User number account is being updated.

The following informative messages are self-explanatory.

CREATION COMPLETE.

UPDATE COMPLETE.

REFORMAT COMPLETE.

CONVERSION TO SOURCE COMPLETE.

INQUIRY COMPLETE.

1.2 PROFILO SPECIAL SYSTEM FILE

The special system file PROFILO contains the information required to control a user's access to the system. This access is defined not only by charge numbers and project numbers, but also by time limits (time in, time out, CP time, 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.

Because company secrecy is involved, programs that access PROFILO must run as special system jobs.

PROFILO 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 PROFILO 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 that apply to this project.

Project numbers and their associated parameters can be entered and changed by company supervisors who have been declared master users in the creation of the current PROFILO 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 (account) 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 PROFILO 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 PROFILO file with 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 PROFILO file is illustrated in Figure IV-1-5.

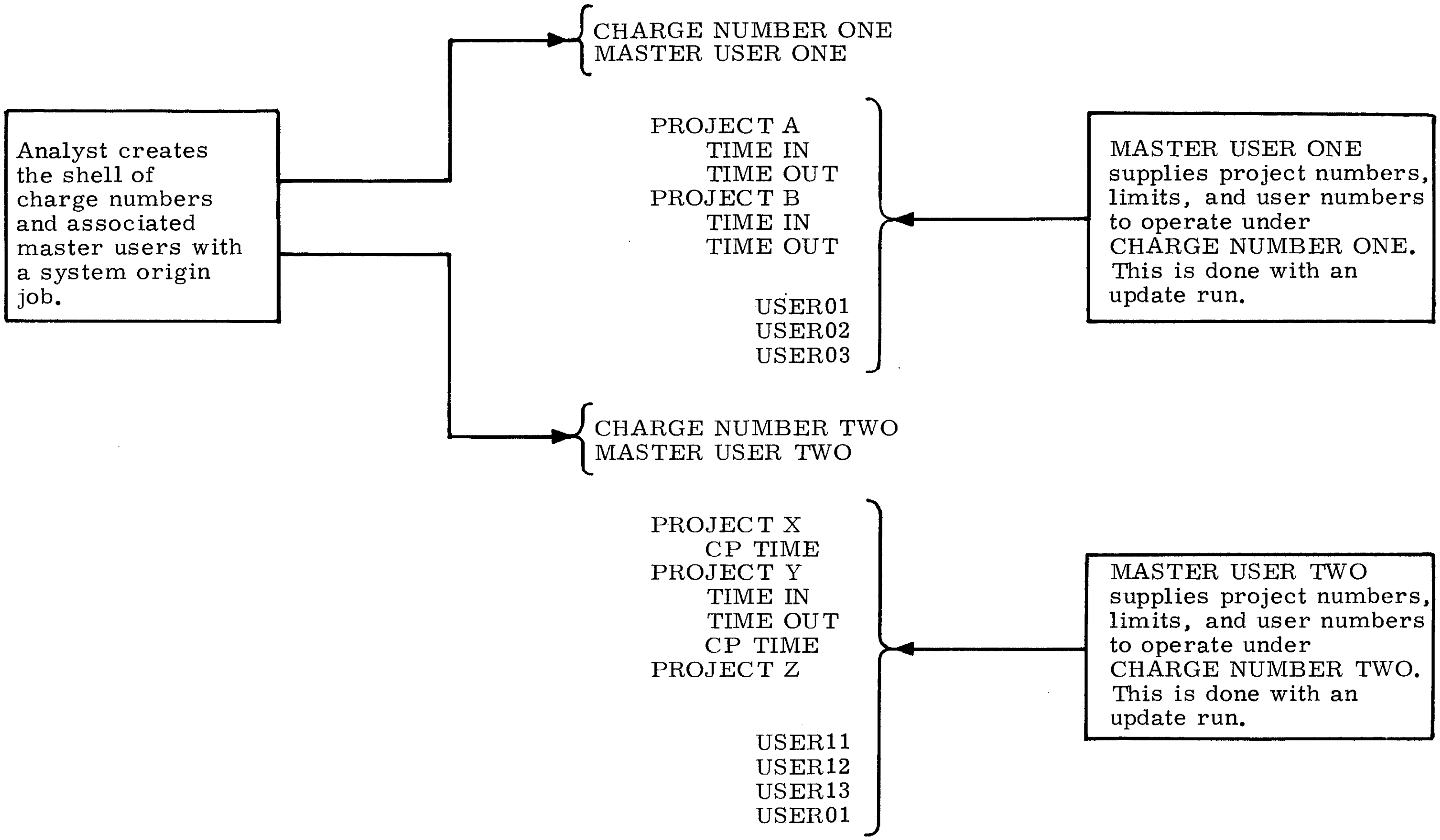

Analyst creates the shell of charge numbers and associated master users with a system origin job.
CHARGE NUMBER ONE
MASTER USER ONE
PROJECT A
TIME IN
TIME OUT
PROJECT B
TIME IN
TIME OUT
USER01
USER02
USER03
MASTER USER ONE supplies project numbers, limits, and user numbers to operate under CHARGE NUMBER ONE. This is done with an update run.
CHARGE NUMBER TWO
MASTER USER TWO
PROJECT X
CP TIME
PROJECT Y
TIME IN
TIME OUT
CP TIME
PROJECT Z
USER11
USER12
USER13
USER01
MASTER USER TWO supplies project numbers, limits, and user numbers to operate under CHARGE NUMBER TWO. This is done with an update run.

Figure 1-5. Representative structuring of a PROFILO file

1.2.1 PROFILE PROJECT PROFILE MANAGER

The project profile manager, PROFILE, uses the special system file supervisor (SFS) to create and manage the project profile file, PROFILO. Creation of PROFILO can only be done from the console via a system origin job (section 1.2.3). Access and modification of an existing PROFILO 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(p1,p2,...,pi,...,pn)
```

where the pi's are the identifications used in defining project profile operations and files.

IDENTIFICATIONS ONLY THE ANALYST CAN USE

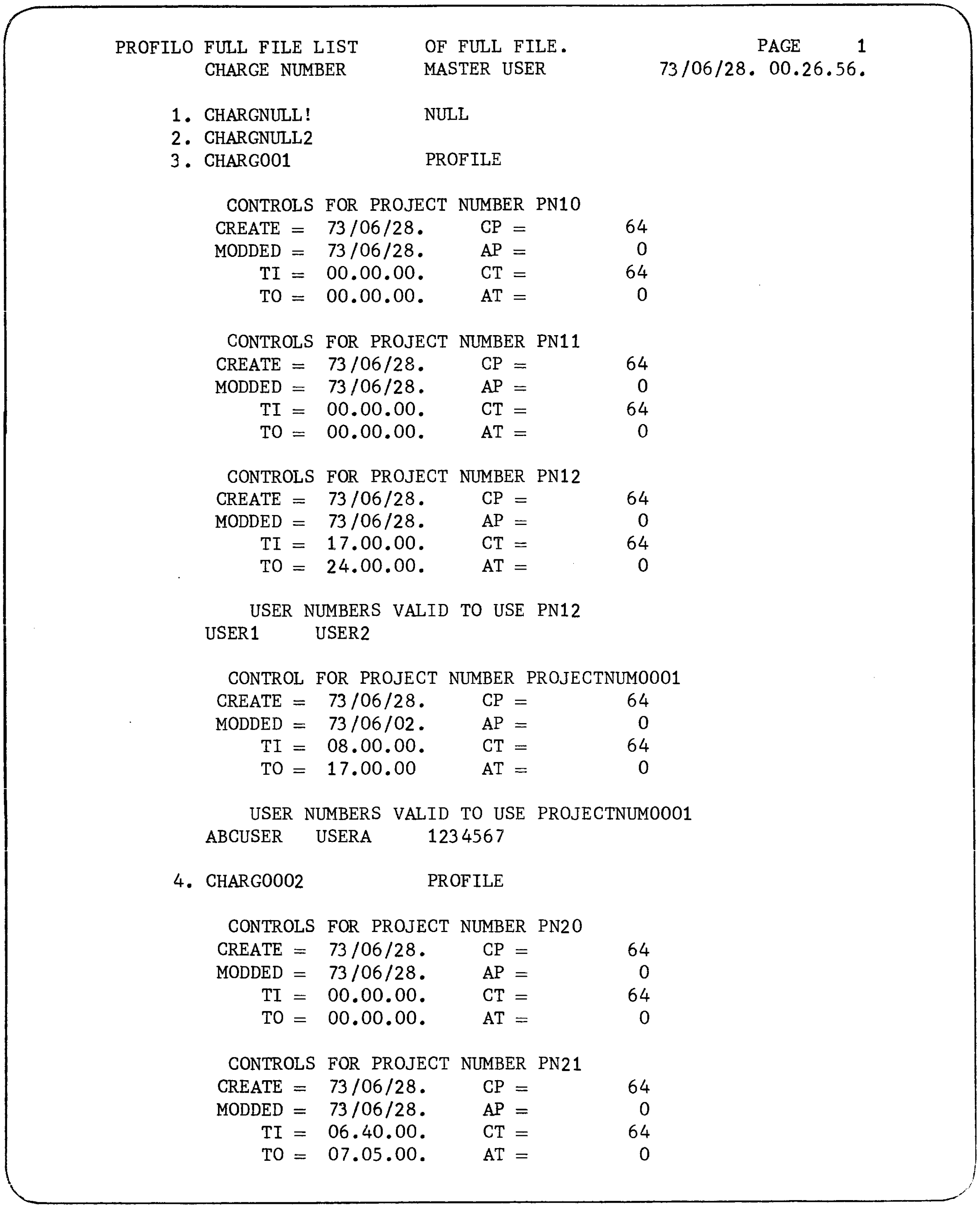
Identification	Description
OP=C	C indicates a create option. Input directives are processed so as to create a new PROFILO file. Directives are entered either through the K display or input file.
OP=K	K display option. All other options are cleared and instructions must be entered via the K display. K 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 PROFILO file, discarding any deleted entries and reconstructing the directory to reduce file access. The existing PROFILO file is replaced with this restructured file.
OP=S	S indicates a source run that returns the PROFILO 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.
S=source	source identifies the file to receive PROFILO source data for the option OP=S. Default is SOURCE.

Identification	Description
OP=L, LO=F	Produces a full listing of the whole PROFILO file. Figure IV-1-6 is an example.
OP=L, LO=C	Produces a listing of charge numbers only for the whole PROFILO file. Figure IV-1-7 is an example.
OP=L, LO=P	Produces a complete charge number and project number listing for the whole PROFILO file. Figure IV-1-8 is an example.
OP=L	Default is LO=F.

IDENTIFICATIONS THE ANALYST AND MASTER USER CAN BOTH USE

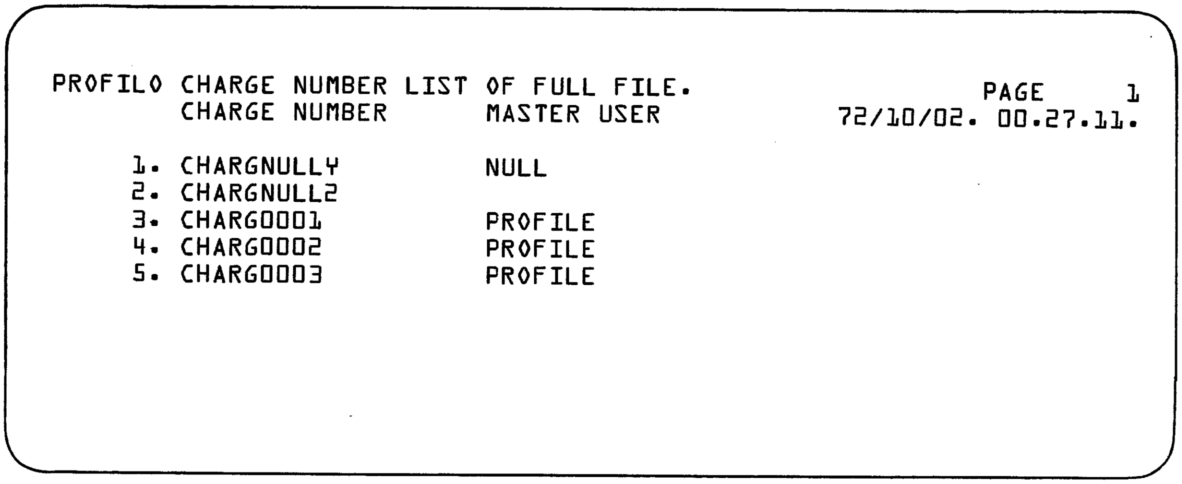
Identification	Description
I=Input	Identifies the file that contains input data for a create (OP=C) and an update (OP=U). Default is INPUT.
L=output	Identifies the file to receive output listings. Default is OUTPUT.
P=profile	Project profile file. Default is PROFILO.
OP=U	Updates the project profile file with directives supplied by the input file.
OP=T	Time-share 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.



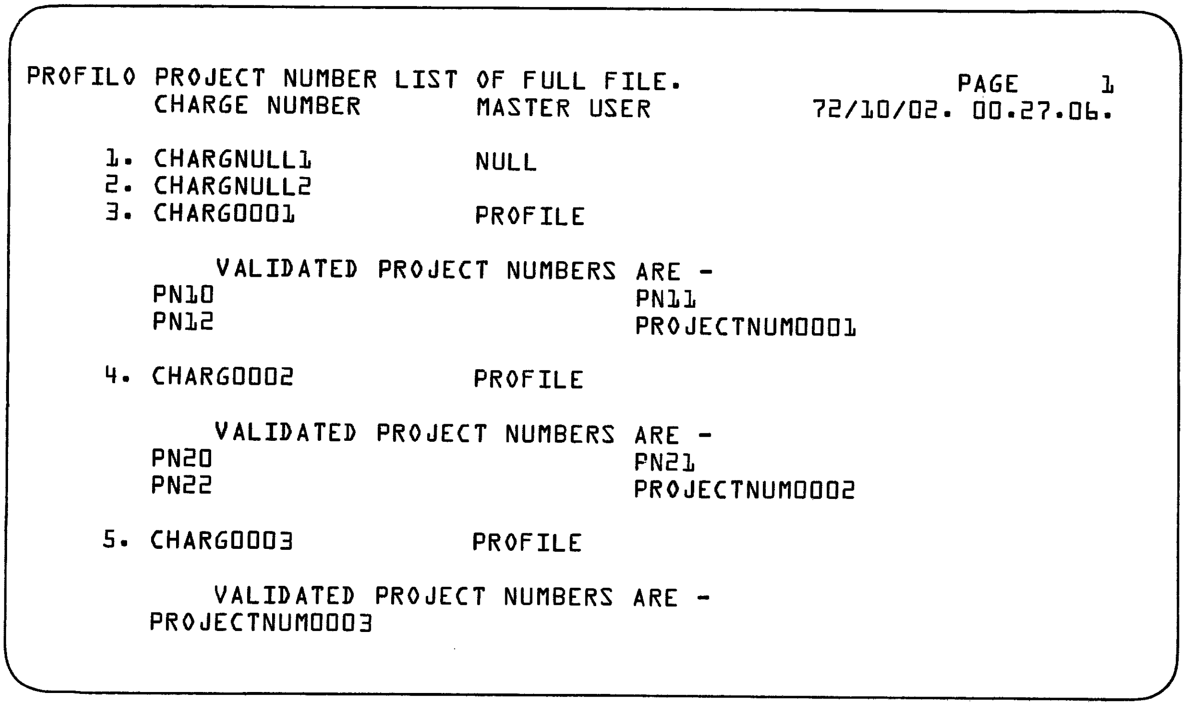
```
PROFILO FULL FILE LIST      OF FULL FILE.                PAGE    1
        CHARGE NUMBER       MASTER USER           73/06/28. 00.26.56.

     1. CHARGNULL!          NULL
     2. CHARGNULL2
     3. CHARG001            PROFILE

          CONTROLS FOR PROJECT NUMBER PN10
         CREATE =  73/06/28.    CP =          64
         MODDED =  73/06/28.    AP =           0
             TI =  00.00.00.    CT =          64
             TO =  00.00.00.    AT =           0

          CONTROLS FOR PROJECT NUMBER PN11
         CREATE =  73/06/28.    CP =          64
         MODDED =  73/06/28.    AP =           0
             TI =  00.00.00.    CT =          64
             TO =  00.00.00.    AT =           0

          CONTROLS FOR PROJECT NUMBER PN12
         CREATE =  73/06/28.    CP =          64
         MODDED =  73/06/28.    AP =           0
             TI =  17.00.00.    CT =          64
             TO =  24.00.00.    AT =           0

            USER NUMBERS VALID TO USE PN12
        USER1     USER2

          CONTROL FOR PROJECT NUMBER PROJECTNUM0001
         CREATE =  73/06/28.    CP =          64
         MODDED =  73/06/02.    AP =           0
             TI =  08.00.00.    CT =          64
             TO =  17.00.00     AT =           0

            USER NUMBERS VALID TO USE PROJECTNUM0001
        ABCUSER   USERA     1234567

     4. CHARG0002           PROFILE

          CONTROLS FOR PROJECT NUMBER PN20
         CREATE =  73/06/28.    CP =          64
         MODDED =  73/06/28.    AP =           0
             TI =  00.00.00.    CT =          64
             TO =  00.00.00.    AT =           0

          CONTROLS FOR PROJECT NUMBER PN21
         CREATE =  73/06/28.    CP =          64
         MODDED =  73/06/28.    AP =           0
             TI =  06.40.00.    CT =          64
             TO =  07.05.00.    AT =           0
```

Figure IV-1-6. Full File List (OP=L, LO=F)



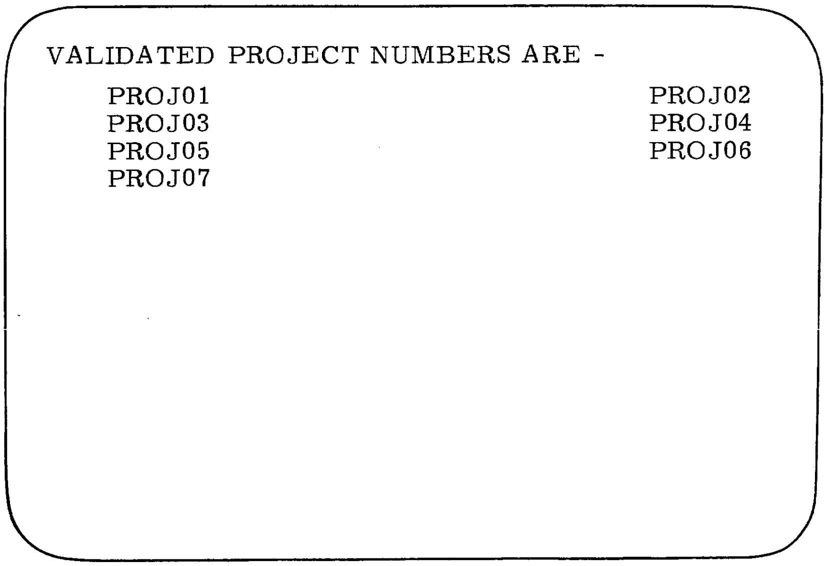
```
PROFILO CHARGE NUMBER LIST OF FULL FILE.                  PAGE    1
        CHARGE NUMBER      MASTER USER             72/10/02. 00.27.11.

     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)



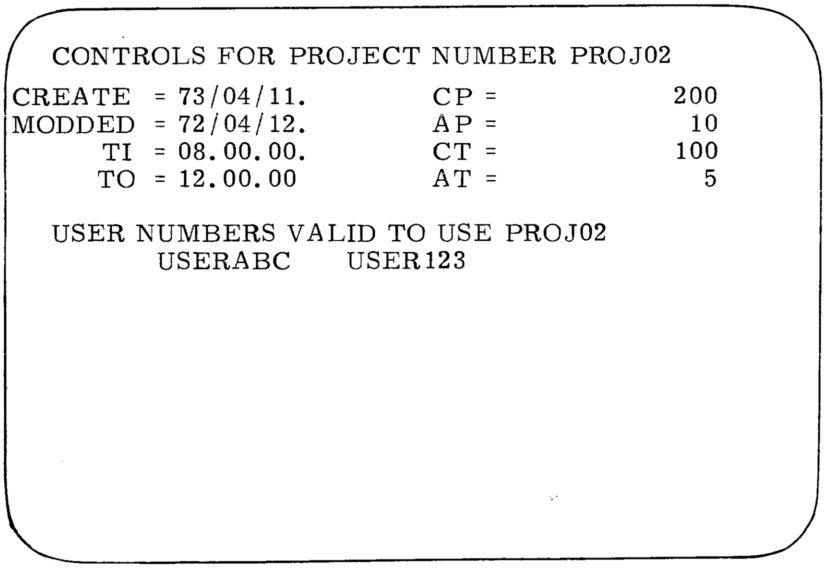
```
PROFILO PROJECT NUMBER LIST OF FULL FILE.                  PAGE    1
        CHARGE NUMBER      MASTER USER             72/10/02. 00.27.06.

     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 PROJECT 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  = 73/04/11.          CP =             200
MODDED  = 72/04/12.          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 file of everything accessible on the PROFILO file by the master user. (A sample listing is given in Figure 1-11.)
OP=L, LO=CM	Charge number list of all charge numbers accessible on the PROFILO file by the master user. (A sample listing is given in Figure 1-12.)
OP=L, LO=PM	Project number list of all project numbers accessible on the PROFILO file by the master user. (A sample listing is given in Figure 1-13.)

1.2.2 CHARGE CONTROL CARD

The system routine CHARGE provides validation of a user's charge and project number for access to defined segments of his company's 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 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.

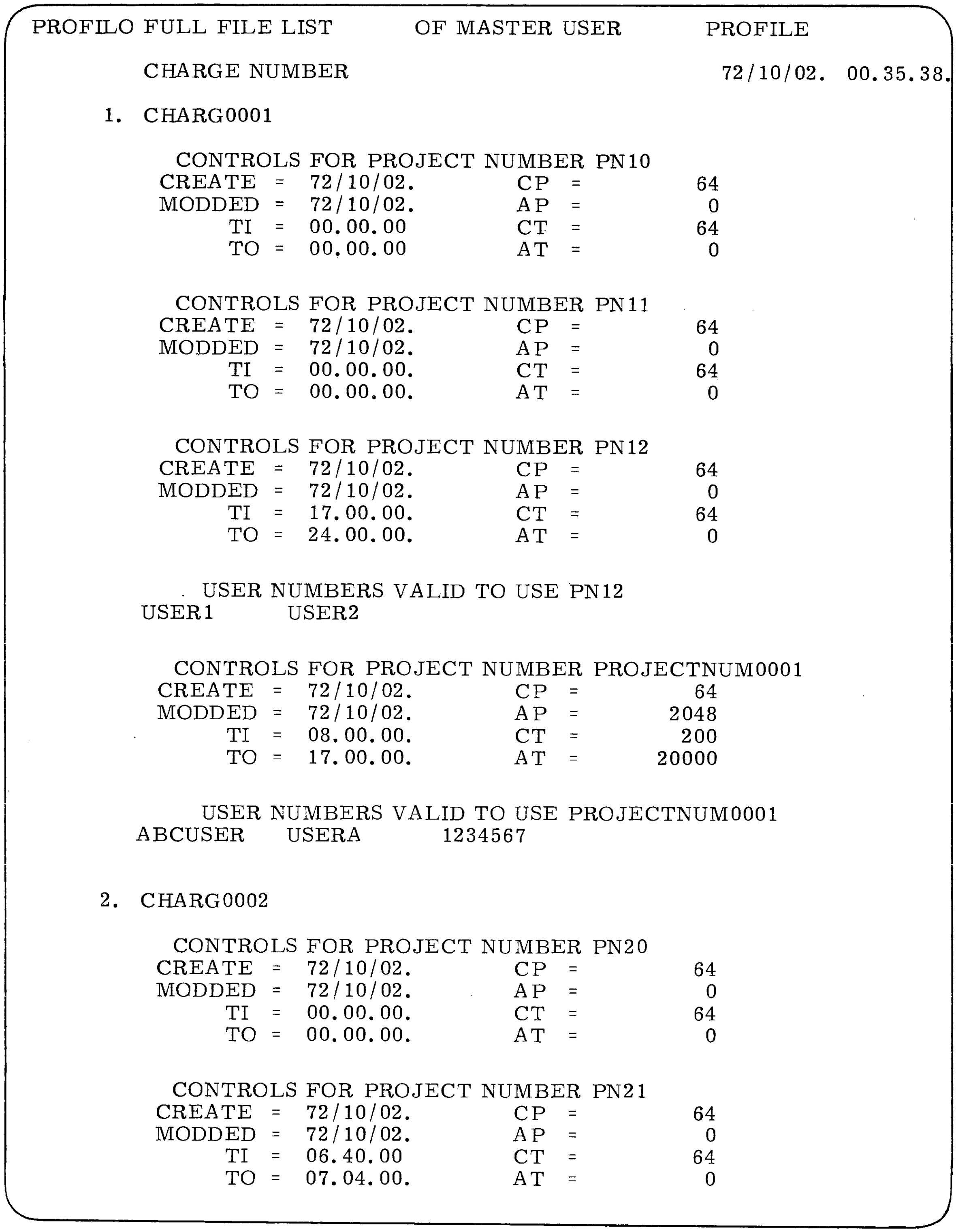
If validation is successful, the following two events occur.

1. Accounting information for all system resources used up to this point is written to the accounting dayfile. This includes:

   - Accumulated CPU time for the job
   - Central memory usage expressed in kilo-word-hours (the product of field length multiplied by time)
   - Mass storage usage expressed as kilo physical records transferred
   - Magnetic tape usage expressed as kilo physical records transferred

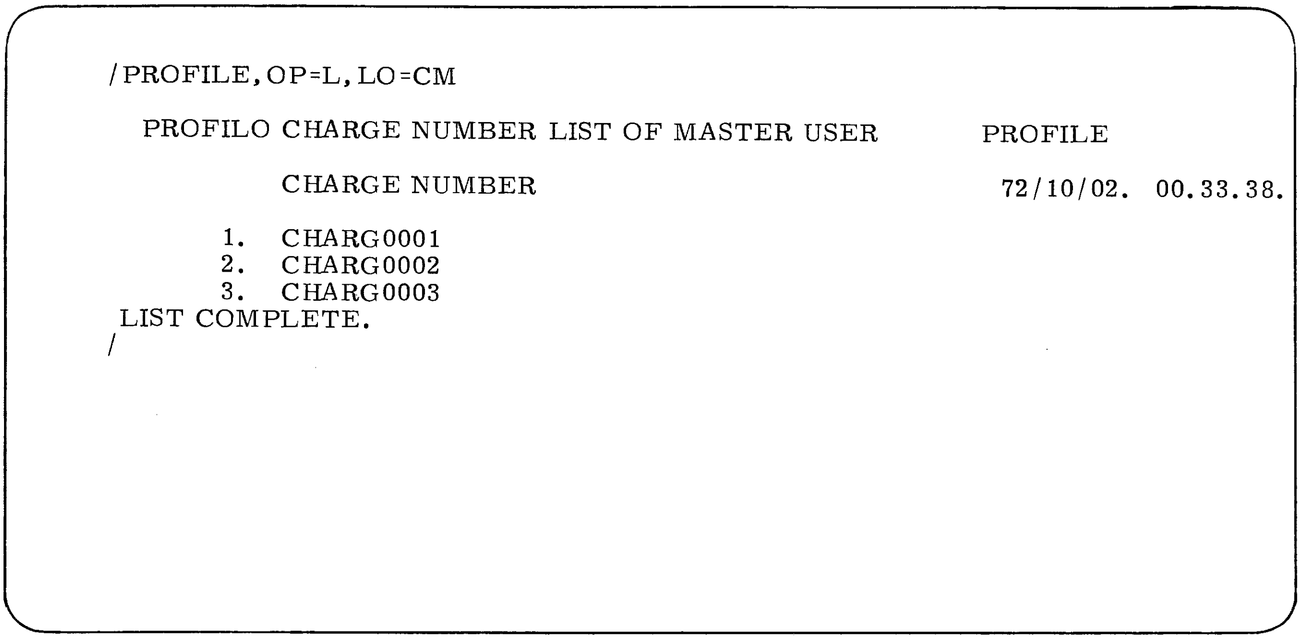
   The message format is:

```
1                   21   26         37
↓                   ↓    ↓          ↓
yy.mm.dd.  hh.mm.ss.CPC  xxxxxx.xxx SEC.
yy.mm.dd.  hh.mm.ss.CMC  xxxxxx.xxx KWH.
yy.mm.dd.  hh.mm.ss.MSC  xxxxxx.xxx KPR.
yy.mm.dd.  hh.mm.ss.MTC  xxxxxx.xxx KPR.
```



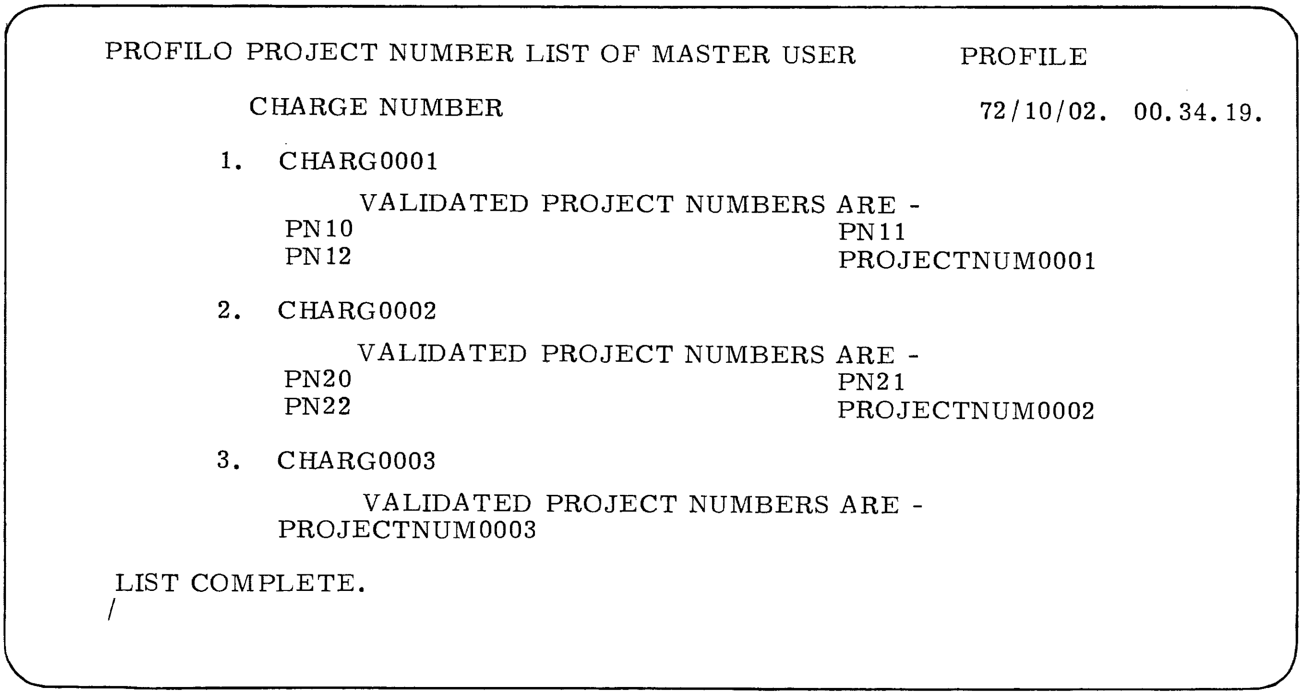
```
PROFILO FULL FILE LIST        OF MASTER USER        PROFILE

        CHARGE NUMBER                                72/10/02.  00.35.38.

     1. CHARG0001

           CONTROLS FOR PROJECT NUMBER PN10
          CREATE =  72/10/02.        CP  =          64
          MODDED =  72/10/02.        AP  =           0
              TI =  00.00.00         CT  =          64
              TO =  00.00.00         AT  =           0

           CONTROLS FOR PROJECT NUMBER PN11
          CREATE =  72/10/02.        CP  =          64
          MODDED =  72/10/02.        AP  =           0
              TI =  00.00.00.        CT  =          64
              TO =  00.00.00.        AT  =           0

           CONTROLS FOR PROJECT NUMBER PN12
          CREATE =  72/10/02.        CP  =          64
          MODDED =  72/10/02.        AP  =           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 =  72/10/02.        CP  =          64
          MODDED =  72/10/02.        AP  =        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 =  72/10/02.        CP  =          64
          MODDED =  72/10/02.        AP  =           0
              TI =  00.00.00.        CT  =          64
              TO =  00.00.00.        AT  =           0

           CONTROLS FOR PROJECT NUMBER PN21
          CREATE =  72/10/02.        CP  =          64
          MODDED =  72/10/02.        AP  =           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

  PROFILO CHARGE NUMBER LIST OF MASTER USER       PROFILE

            CHARGE NUMBER                          72/10/02.  00.33.38.

        1.  CHARG0001
        2.  CHARG0002
        3.  CHARG0003
 LIST COMPLETE.
/
```

Figure IV-1-12. Master User Charge Number List (OP=L, LO=CM)



```
PROFILO PROJECT NUMBER LIST OF MASTER USER        PROFILE

         CHARGE NUMBER                             72/10/02.  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)

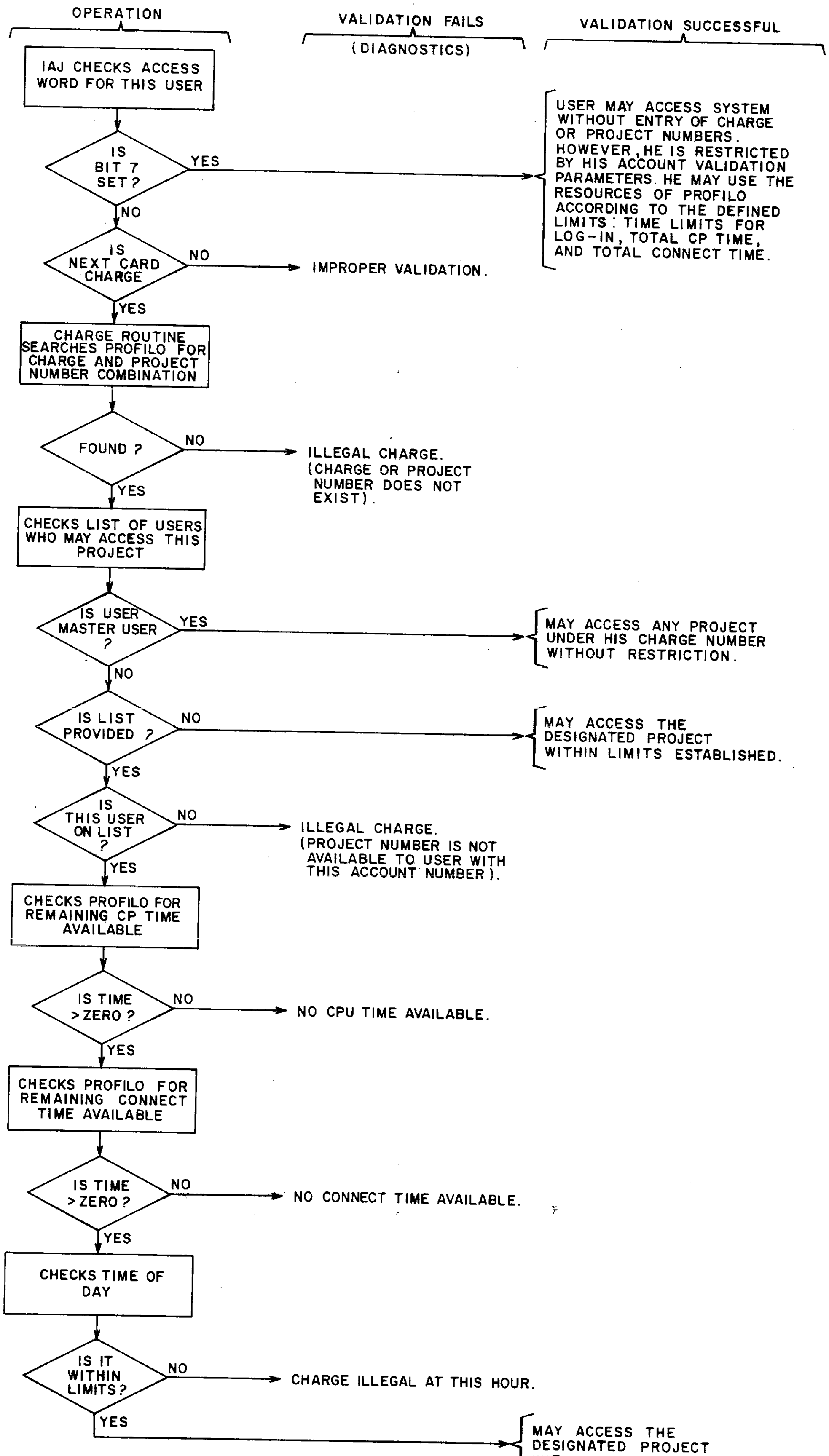

OPERATION
VALIDATION FAILS
(DIAGNOSTICS)
VALIDATION SUCCESSFUL
IAJ CHECKS ACCESS WORD FOR THIS USER
IS BIT 7 SET?
YES
USER MAY ACCESS SYSTEM WITHOUT ENTRY OF CHARGE OR PROJECT NUMBERS. HOWEVER, HE IS RESTRICTED BY HIS ACCOUNT VALIDATION PARAMETERS. HE MAY USE THE RESOURCES OF PROFILO ACCORDING TO THE DEFINED LIMITS: TIME LIMITS FOR LOG-IN, TOTAL CP TIME, AND TOTAL CONNECT TIME.
NO
IS NEXT CARD CHARGE
NO
IMPROPER VALIDATION.
YES
CHARGE ROUTINE SEARCHES PROFILO FOR CHARGE AND PROJECT NUMBER COMBINATION
FOUND ?
NO
ILLEGAL CHARGE. (CHARGE OR PROJECT NUMBER DOES NOT EXIST).
YES
CHECKS LIST OF USERS WHO MAY ACCESS THIS PROJECT
IS USER MASTER USER ?
YES
MAY ACCESS ANY PROJECT UNDER HIS CHARGE NUMBER WITHOUT RESTRICTION.
NO
IS LIST PROVIDED ?
NO
MAY ACCESS THE DESIGNATED PROJECT WITHIN LIMITS ESTABLISHED.
YES
IS THIS USER ON LIST ?
NO
ILLEGAL CHARGE. (PROJECT NUMBER IS NOT AVAILABLE TO USER WITH THIS ACCOUNT NUMBER).
YES
CHECKS PROFILO FOR REMAINING CP TIME AVAILABLE
IS TIME > ZERO ?
NO
NO CPU TIME AVAILABLE.
YES
CHECKS PROFILO FOR REMAINING CONNECT TIME AVAILABLE
IS TIME > ZERO ?
NO
NO CONNECT TIME AVAILABLE.
YES
CHECKS TIME OF DAY
IS IT WITHIN LIMITS?
NO
CHARGE ILLEGAL AT THIS HOUR.
YES
MAY ACCESS THE DESIGNATED PROJECT WITHIN LIMITS ESTABLISHED.

Figure IV-1 14. CHARGE Routine

Values indicated for resources are cumulative.

2. This charge number and this project number are written to the accounting dayfile to be used for accounting purposes for this job. The format of this entry is:

```
1                    21    27
↓                    ↓     ↓
yy. mm. dd.  hh. mm. ss. CH    chargenumbr, projectnumber
```

The CHARGE routine is called from a terminal as either a response to the teletype 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 CHARGE control statement after an ACCOUNT card.

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 card is the ACCOUNT card which gives the account number used in validating a user for system access (VALIDUS). 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 card following the ACCOUNT card must be a CHARGE card 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 card. If additional ACCOUNT cards (new users) appear in the input, each one that identifies a user requiring job profile validation must be followed by a CHARGE card with the requisite charge and project numbers.

CHARGE APPLICATION AT A TERMINAL

When any user logs in at a teletype, the access word associated with his account 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 unrestricted access to resources.

If charge and project are required, the teletype will print out at login:

```
TERMINAL        nn
RECOVER/CHARGE:
```

This user will not be allowed to enter any commands (except CHARGE, LOGIN, HELLO, 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 account number.

DATA BASE ERROR

Error in validation file structure detected. Contact an on-site analyst immediately.

CHARGE ABORTED.

External job abort.

ILLEGAL CONTROL CARD.

No terminator was found on control card.

CHARGE FILE BUSY.

File is not available for charge purposes. Wait and try again.

NO CPU TIME AVAILABLE.

Central processor time allowed under this project number has been expended.

NO CONNECT TIME AVAILABLE.

Teletype connect time allowed under this project number has been expended.

CHARGE ILLEGAL AT THIS HOUR.

This project number cannot be used at this time of day.

1.2.3 PROFILE CONSOLE INPUT

A PROFILO file can only be created by calling PROFILE from the console (system origin job). Likewise, an existing PROFILO 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 PROFILO 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 PROFILO 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 PROFILO 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.
```

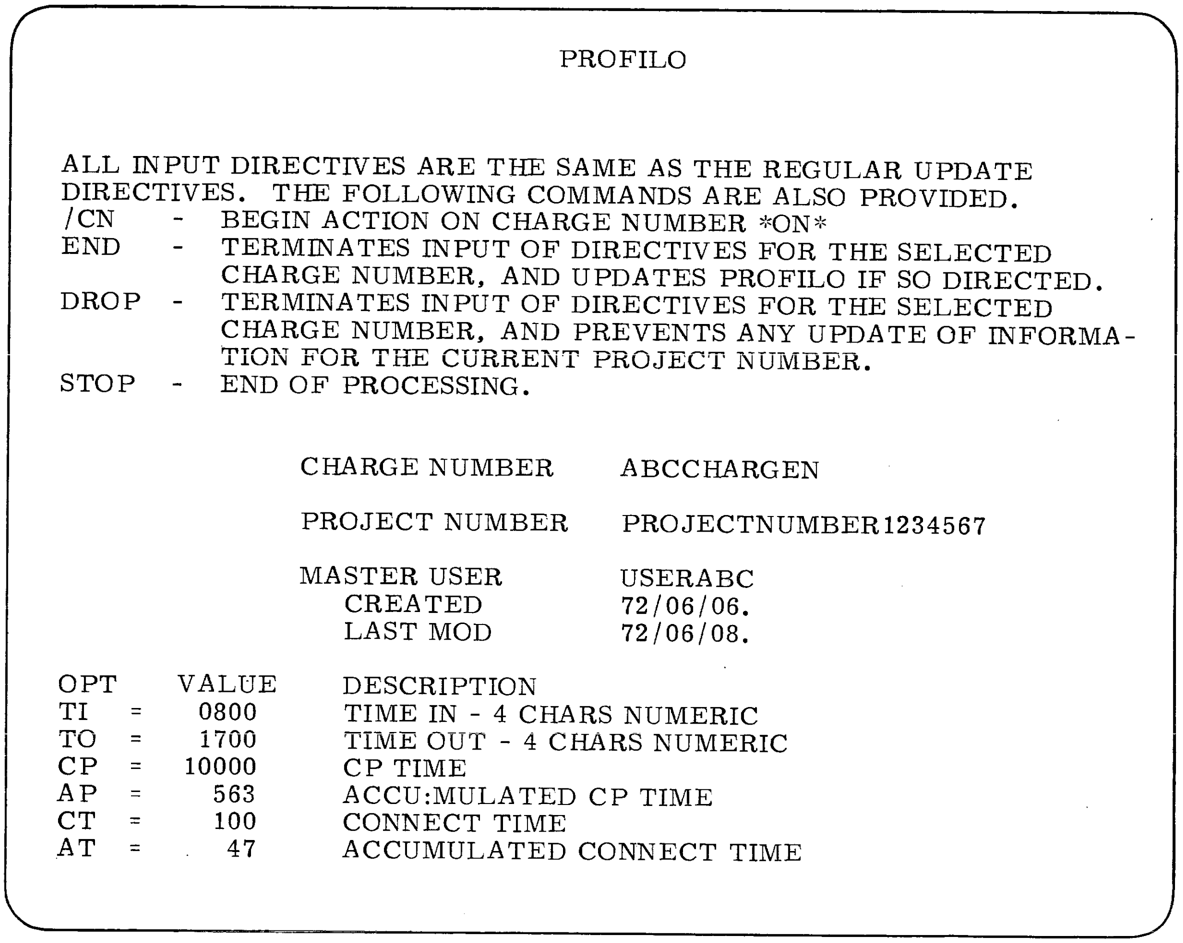
The B display will indicate the control point to which PROFILE is located. A flashing message on that line will be:

```
REQUEST K DISPLAY
```

The analyst types in

```
K,n.
```

where n is the control point number for PROFILE. This brings the K display for PROFILO 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.



```
                              PROFILO


ALL INPUT DIRECTIVES ARE THE SAME AS THE REGULAR UPDATE
DIRECTIVES.  THE FOLLOWING COMMANDS ARE ALSO PROVIDED.
/CN    -  BEGIN ACTION ON CHARGE NUMBER *ON*
END    -  TERMINATES INPUT OF DIRECTIVES FOR THE SELECTED
          CHARGE NUMBER, AND UPDATES PROFILO IF SO DIRECTED.
DROP   -  TERMINATES INPUT OF DIRECTIVES FOR THE SELECTED
          CHARGE NUMBER, AND PREVENTS ANY UPDATE OF INFORMA-
          TION FOR THE CURRENT PROJECT NUMBER.
STOP   -  END OF PROCESSING.


               CHARGE NUMBER      ABCCHARGEN

               PROJECT NUMBER     PROJECTNUMBER1234567

               MASTER USER        USERABC
                  CREATED         72/06/06.
                  LAST MOD        72/06/08.

OPT     VALUE     DESCRIPTION
TI   =   0800     TIME IN - 4 CHARS NUMERIC
TO   =   1700     TIME OUT - 4 CHARS NUMERIC
CP   =  10000     CP TIME
AP   =    563     ACCU:MULATED CP TIME
CT   =    100     CONNECT TIME
AT   =     47     ACCUMULATED CONNECT TIME
```

Figure IV-1-15. PROFILO K Display (Left Screen Only)

```
K./CHARJNUM1

K.MU=MUSE1

K.END

K./CHARJNUM2

K.MU=MUSE2

K.END

K.STOP
```

[The "/" is used when PROFILE is updating a particular charge number. Otherwise it is not used.]

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. This implies that a charge number should appear only once in an input stream.

The directives within a charge number entry are further divided into master user, 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.

Figure IV-1-16 diagrams a typical input stream.

The card image 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.

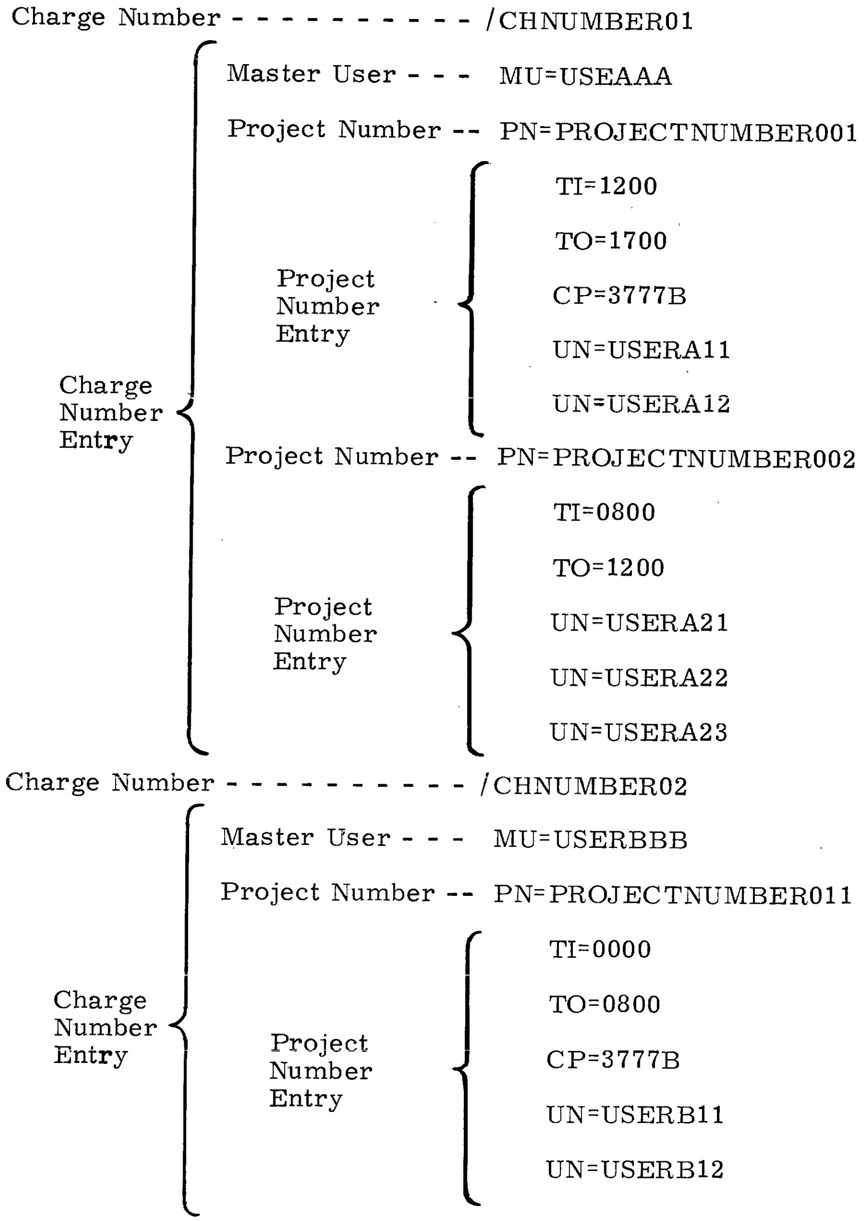
All data within a charge number entry is free format to column 72. Directives cannot be split between cards or lines. Blanks are ignored.

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 ACCOUNT card 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
                 Master User - - -  MU=USEAAA
                 Project Number --  PN=PROJECTNUMBER001
                                       TI=1200
                                       TO=1700
                    Project
                    Number             CP=3777B
                    Entry
                                       UN=USERA11
   Charge
   Number                              UN=USERA12
   Entry         Project Number --  PN=PROJECTNUMBER002
                                       TI=0800
                                       TO=1200
                    Project
                    Number             UN=USERA21
                    Entry
                                       UN=USERA22
                                       UN=USERA23
Charge Number - - - - - - - - - - /CHNUMBER02
                 Master User - - -  MU=USERBBB
                 Project Number --  PN=PROJECTNUMBER011
                                       TI=0000
   Charge
   Number                              TO=0800
   Entry            Project
                    Number             CP=3777B
                    Entry
                                       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.

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 one-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 KRONOS, 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.

CP — Total amount of central processor time in CP seconds that this project number is allowed to accumulate. Time may be one to ten numeric digits. A radix may be added. Default value is 100B.

AP — Total amount of central processor time in CP seconds that this project number has accumulated. Time may be one to ten 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. This value is not currently updated by the system.

Identifier	Description
CT	Total amount of accumulated time in seconds that users are allowed to be connected to this project. Time may be one to ten 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 one to ten 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. This value is not currently updated by the system.
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 PROFILO 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

ACCOUNT(MASTR1)

PROFILE(OP=U)

7/8/9
```

```
/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. †

```
 73/04/10.  13.56.20.
KRONOS TIME SHARING SYSTEM - VER. 2.1.
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
   CP = MAXIMUM CENTRAL PROCESSOR TIME
   AP = ACCUMULATED CENTRAL PROCESSOR TIME
   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 card 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 PROJECT NUMBER
? PROJ2  CR
```

The master user can then enter his update parameters according to the following:

```
? TI=0800  CR
? TO=1200  CR
?  CR
/
```

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).

If a project number validation is needed and has not been supplied, PROFILE outputs

ENTER PROJECT NUMBER

to the terminal and waits for the user to type in the project number for which he wants information. If a null line is entered, it indicates the end of processing for the current charge number.

If the CN appears alone, all valid project numbers for that CN are printed. PROFILE then requests the next charge number, that is, it prints out

ENTER CHARGE NUMBER

If this master user is validated for an additional charge number, he can enter it at this time (with the CHARGE directive). If not, he enters a null line as an end-of-file.

If both CN and PN are entered on the PROFILE command, all valid user numbers for that project are printed out. PROFILE then requests the next project number with

ENTER PROJECT NUMBER

The master user can then enter another project number under his charge to get a listing of its user numbers, or he can type a null line 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 PROFILO 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 card.

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.

PROFILO CREATED.

Creation run completed.

PROFILO UPDATED.

Update run completed.

FILE NOT AVAILABLE CURRENTLY.

PROFILO file busy. Try again later.

INQUIRY COMPLETE.

Inquire run completed.

REFORMAT COMPLETE.

Reformat completed.

PROFILO TO SOURCE.

Source run completed.

LIST COMPLETE.

List of PROFILO 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 PROFILO 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 - - - - -                            |
     CP=0777B              }  Project                   |
     UN=USERA              }  Number                    |    Charge
     UN=USERB      - - - - -  Entry                     |    Number
 PN=PROJECTNUMBER2 - - - - -                            }    Entry
     CP=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 one to seven 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 one to seven alphanumeric characters or an asterisk).

Action: Entire charge entry is disregarded and PROFILE skips to next charge entry.

PERMANENT FILE UTILITIES 2

Five utility processors maintain the KRONOS 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 (TTY). However, in all cases the job containing the utility control card must be system origin type or have system origin privileges.

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.

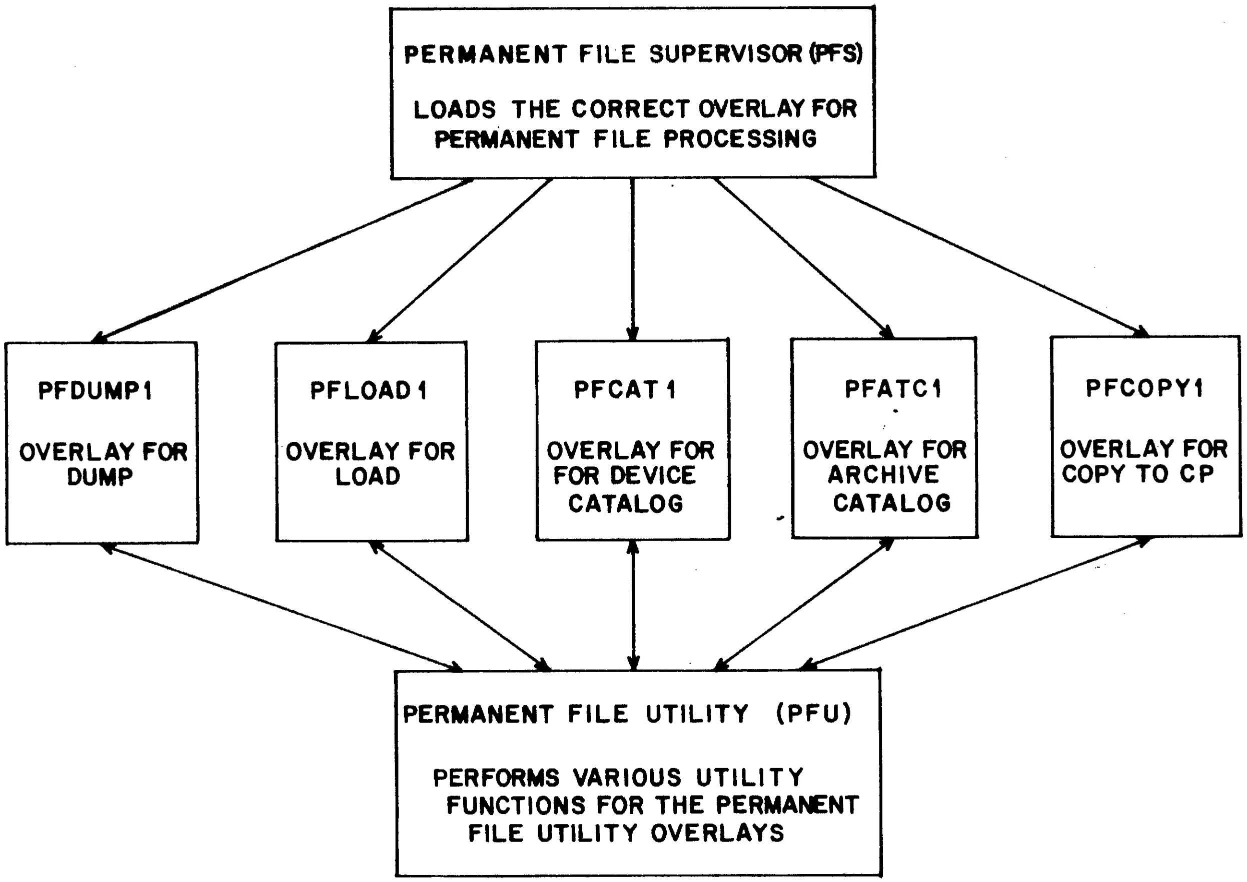

PERMANENT FILE SUPERVISOR (PFS)
LOADS THE CORRECT OVERLAY FOR PERMANENT FILE PROCESSING
PFDUMP1
OVERLAY FOR DUMP
PFLOAD1
OVERLAY FOR LOAD
PFCAT1
OVERLAY FOR FOR DEVICE CATALOG
PFATC1
OVERLAY FOR ARCHIVE CATALOG
PFCOPY1
OVERLAY FOR COPY TO CP
PERMANENT FILE UTILITY (PFU)
PERFORMS VARIOUS UTILITY FUNCTIONS FOR THE PERMANENT FILE UTILITY OVERLAYS

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 physical 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 account number created on the VALIDUX file. This index is entered through MODVAL (section 1.1.1) with the UI identifier on the account-number input directive (/accnumb,UI=xxxxxx), or MODVAL, by default, supplies the next available index.

Whenever this account submits a job, the related user index is placed in the control point area along with the account 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 account. 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:

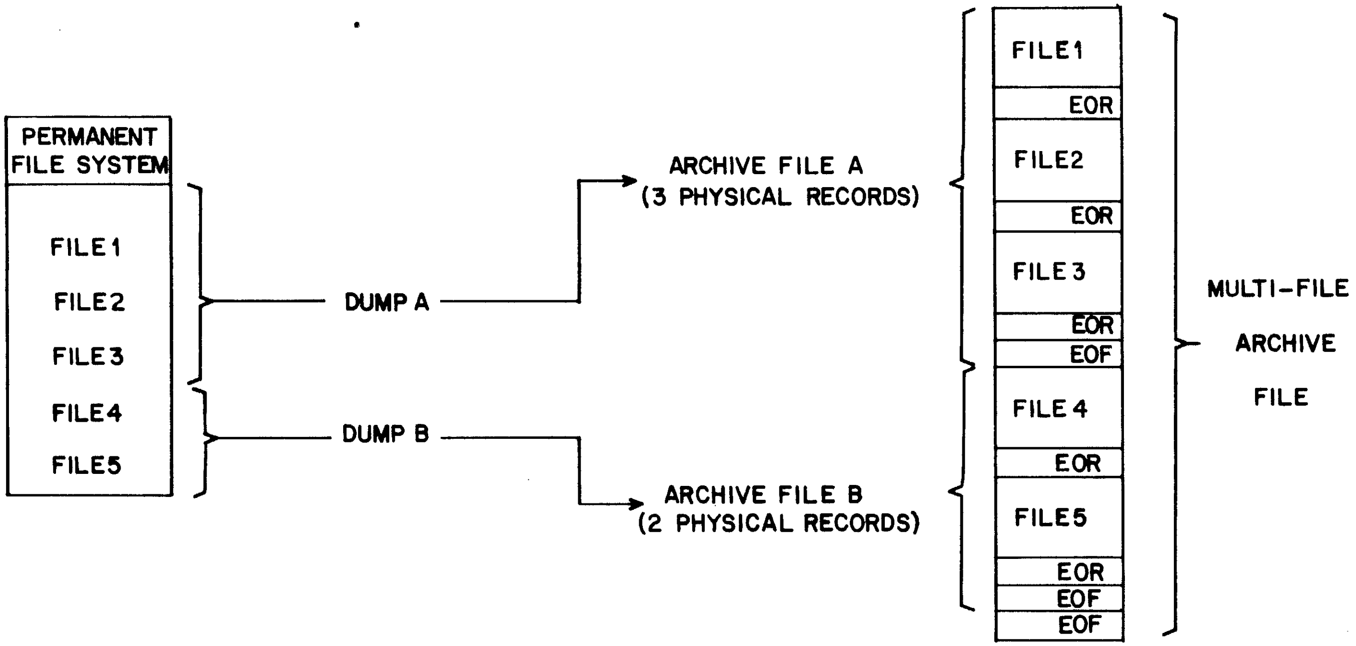


Figure IV-2-2. Example of Multifile Archive File Structure

Type of Device	Default Number of Catalog Tracks
DA 6608	10
DB 6638	10
DC 863	2
DD 854	4
DE ECS	2
DF 814	100
DH 821	100
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.

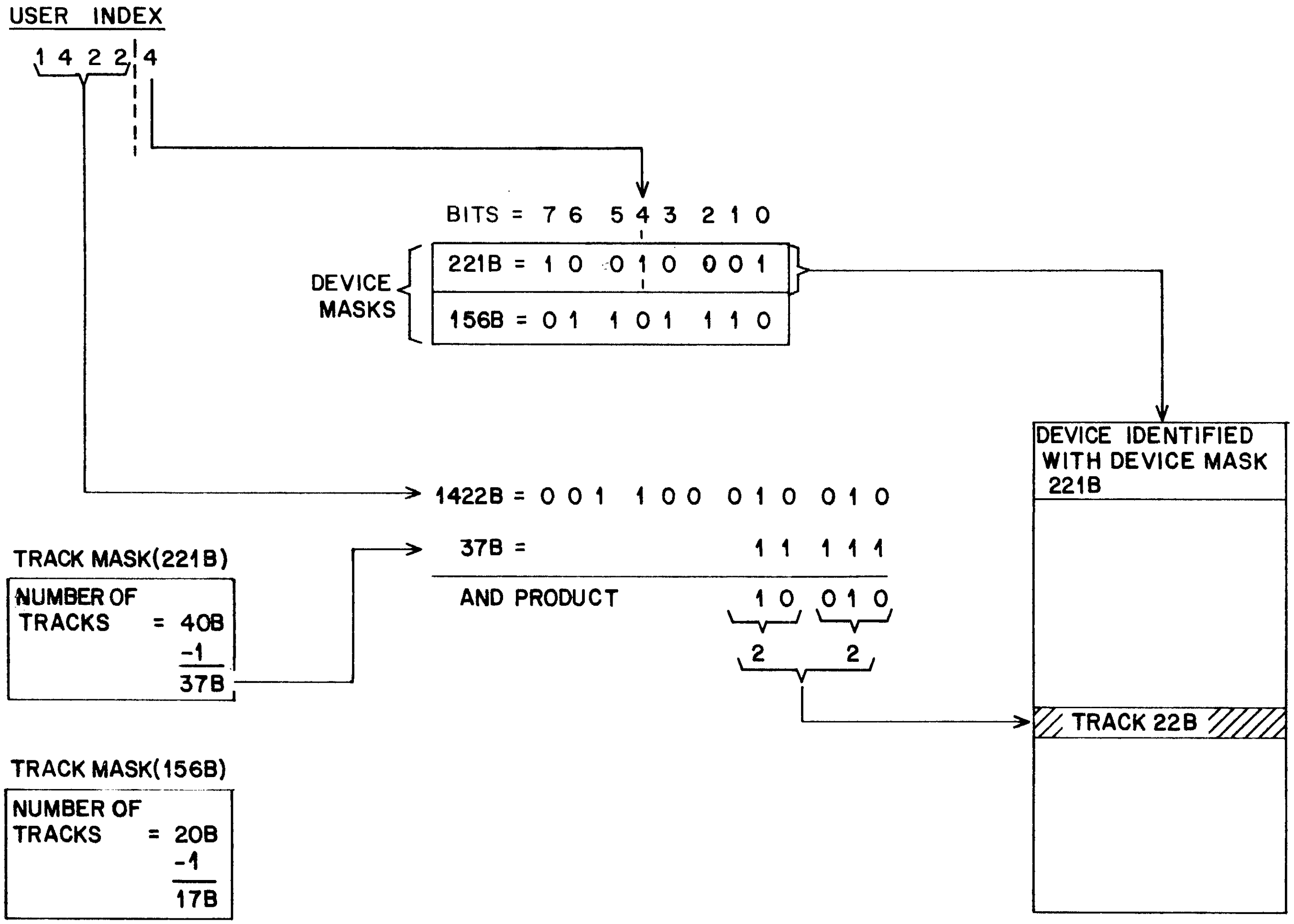


Figure IV-2-3. Example of Masking Operations With User Index

2.1.3 DEVICE 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 files (direct files are 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:

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:

$$\begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{1} & \multicolumn{1}{c|}{1} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{1} \\ \hline \end{array}$$

$$\underbrace{1\quad 0}_{2}\quad \underbrace{1\quad 1\quad 0}_{6}\quad \underbrace{0\quad 0\quad 1}_{1}$$

Within a family, the sum of all the device masks must total exactly 377B and each bit must be accounted for only once.

Example:

For a family of three devices, appropriate device masks are:

```
10 010 001  =  221B
00 100 010  =  042B
01 001 100  =  114B
-----------    ----
11 111 111  =  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:

For a family of four devices, the following masks were designated by the EQ entry at deadstart.

```
01 100 010  =  142B
00 010 000  =  020B
00 001 000  =  010B
10 000 100  =  204B
-----------    ----
11 111 110  =  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:

For the following family of four devices:

```
10 001 010   =   212B
01 000 110   =   106B
00 100 000   =   040B
00 010 001   =   021B
----------       ----
11 111 111   =   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:

```
10 110 001   =   261B
01 001 101   =   115B
00 000 001   =   001B
----------       ----
11 111 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 from the devices in one family.

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.)

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 time and date				
access count				modification time and date				
ct	mode	▨	dn	last access time and date				
▨								
file password						▨		
program control word								

filename	Permanent file name
user index	User index of file creator
file length	Length of the file in PRUs
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.
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

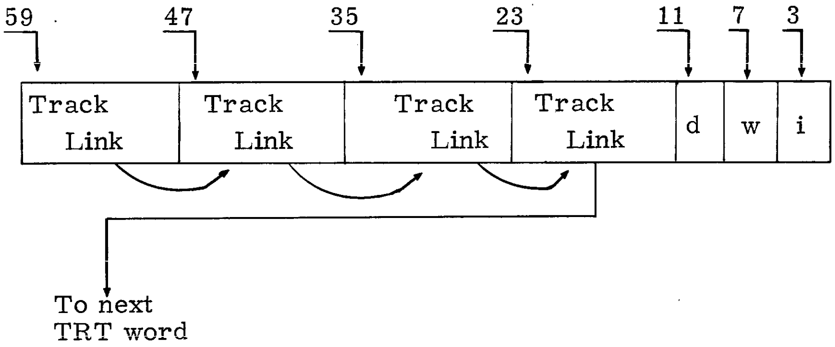
Every device in a permanent file system has a label track (usually track 0) which contains a label sector describing the device (family name, device mask, location of permit information, catalog information, and indirect files) and a number of PRUs containing the track reservation table (TRT). The TRT 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).

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



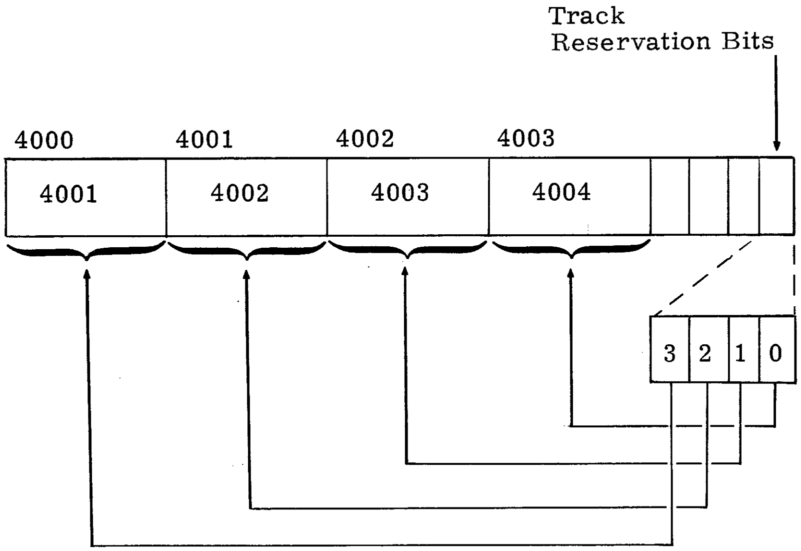
Track Link — Address of the next track that is a logical continuation of this file. The track links are shown as sequential within a word, but this is not a requirement.)

d — Bit settings for identifying the first track of a permanent file chain

w — Bit settings for establishing interlock of a track

i — Bit settings for track reservation

Figure IV-2-4. Track Reservation Table Word



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 link byte references a track that 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, or a permit chain.

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 byte references a track that 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
```

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.

2.2 CALLING THE UTILITIES

Any of the permanent file utilities can be called from console (K display), card input (batch), or terminal (TTY). 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-2-6) appears on the left screen.

3. The analyst types in

   KK. (CR)

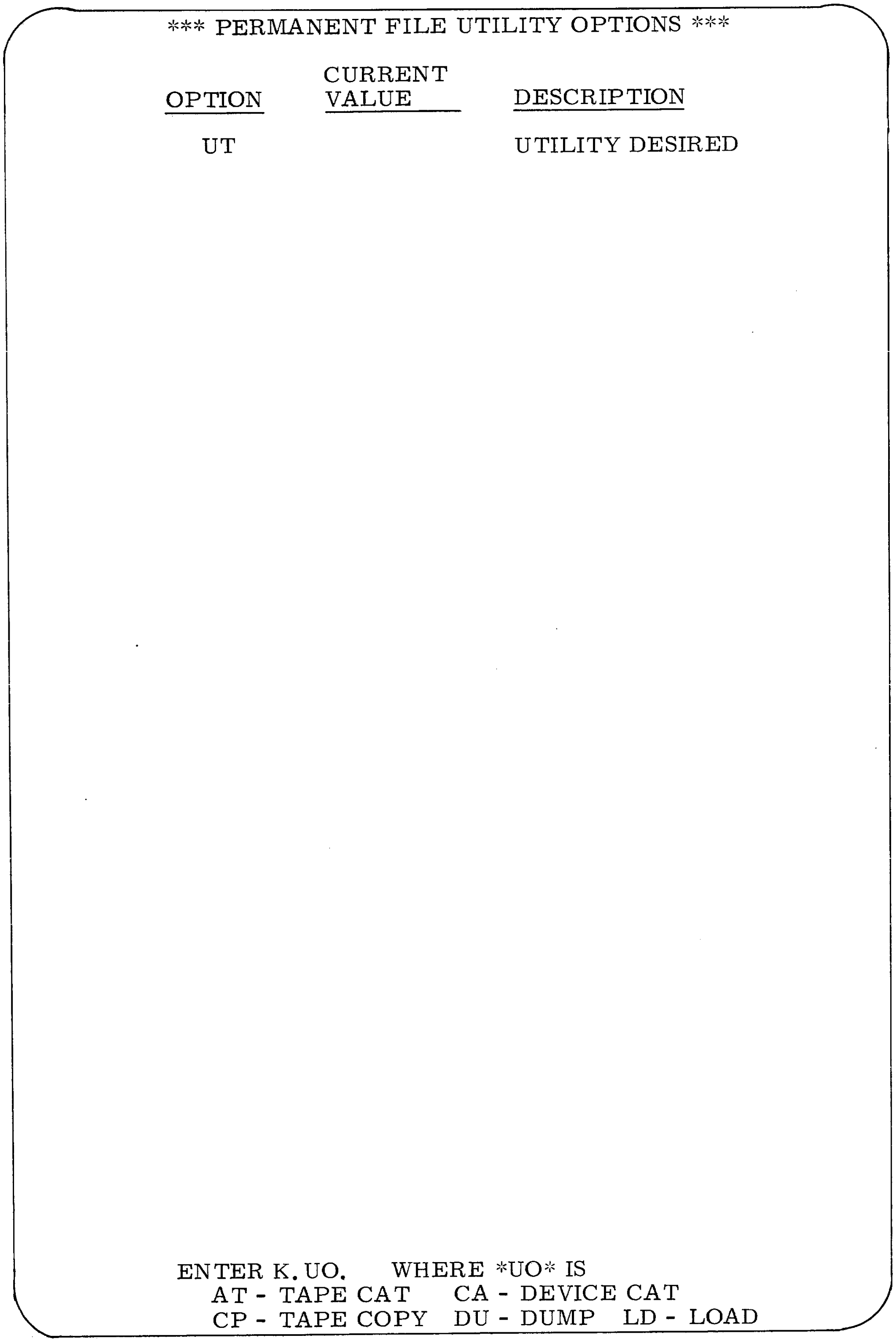
   This brings the permanent file utilities right K display (Figure IV-2-7) to the right screen. This will give a description of the permanent file utility options available.

4. The request for the desired utility is entered with

   K.uo. (CR)

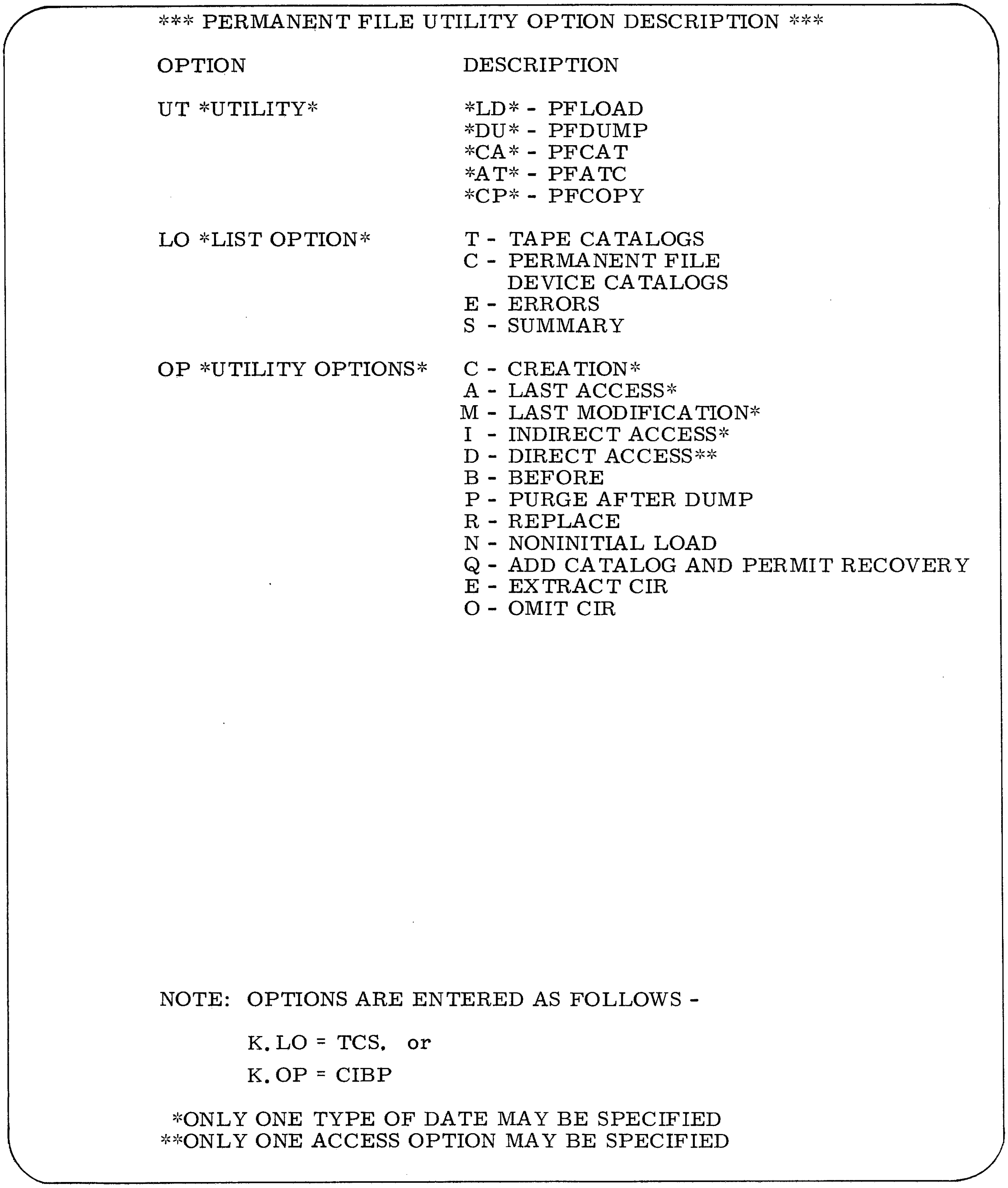
   where uo is one of the following.

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-2-6. 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-2-7. Permanent File Utilities Right K Display

The parameters available under the chosen utility appear on the left screen. Figure IV-2-8 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.

The CAMIDBP ③ appearing after the UTILITY OPTION indicates that:

C	Creation
A	Last access
M	Last modification
I	Indirect files
D	Direct files
B	Before date and time
P	Purge after dump

are the options available for the dump utility.

5. The desired parameters are entered with the command

   K.p1, p2, . . . , pn Ⓒⓡ

   where the pi are selected from the parameter list on the left screen. The selected parameters replace the default values listed on the left screen.

6. To initiate execution, the analyst types

   K.GO. Ⓒⓡ

   The lower lines of the left display ④ disappears and is replaced by:

   ```
   DEVICE                 MASK
   DATE                   TIME
   ```

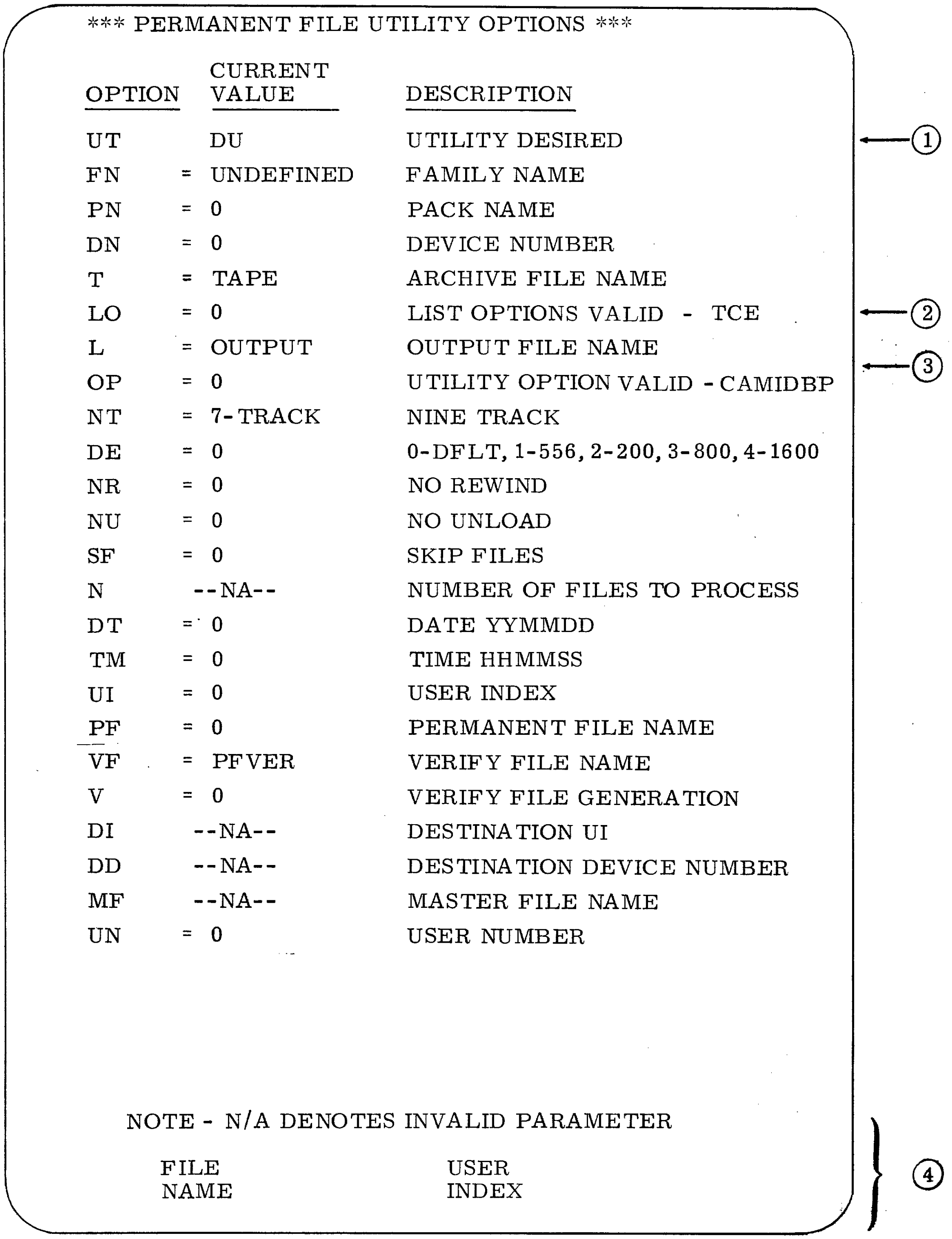
   Values appear with these identifiers as processing continues.

2.2.2 BATCH INPUT

The deck structure for card input to call a permanent file utility is:

JOB card
ACCOUNT card
PFutil(p1, p2, . . . , pn)

where util is one of the following utility options.



```
*** PERMANENT FILE UTILITY OPTIONS ***

         CURRENT
OPTION   VALUE           DESCRIPTION

UT       DU              UTILITY DESIRED
FN     = UNDEFINED       FAMILY NAME
PN     = 0               PACK NAME
DN     = 0               DEVICE NUMBER
T      = TAPE            ARCHIVE FILE NAME
LO     = 0               LIST OPTIONS VALID  -  TCE
L      = OUTPUT          OUTPUT FILE NAME
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                   USER
      NAME                   INDEX
```

Figure IV-2-8. 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 card is read, the K display appears. However, since parameters were entered on the control card, 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.

2.3 PARAMETERS FOR THE UTILITY CONTROL STATEMENTS

The parameters available for the permanent file utility control statements are listed in this section. Each entry in the list includes the name, the parametric designation, and the range of characters or digits required. Following the list is a description of each parameter. The descriptions are in the same order as the list.

2.3.1 LIST OF UTILITY PARAMETERS

Family name (FN), 1 to 7 characters

Pack name (PN), 1 to 7 characters

Device number (DN), 2 octal digits

Utility option (OP), 1 to 7 characters

Archive file name (T), 1 to 7 characters

Nine-track (NT), (NT sets the option)

Density (DE), 1 decimal digit

No rewind (NR), (NR sets the option)

No unload (NU), (NU sets the option)

Skip files (SF), 2 decimal digits

Number of files to process (N), 2 decimal digits

Date (DT), 6 decimal digits (yymmdd) where yy = year, mm = month, and dd = day

Time (TM), 6 decimal digits (hhmmss) where hh = hour, mm = minute, and ss = second

User index (UI), 6 octal digits
Destination index (DI), 6 octal digits
Permanent file name (PF), 1 to 7 characters
Verify file name (VF), 1 to 7 characters
Verify (V), (V sets option)
Output file name (L), 1 to 7 characters
List option (LO), 1 to 7 characters
Destination device (DD), 2 octal digits
Master file (MF), 1 to 7 characters
User number (UN), 1 to 7 characters

2.3.2 DESCRIPTIONS OF UTILITY PARAMETERS

FN = family name

Identifies 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

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

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 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.

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 parameter is used only with C, A, or M. DT and TM are optional. C, A, or M must be specified.

B Specifies before time in selection of files according to event.

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.

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 permanent file system, PFLOAD skips loading of the same-name file from the archive tape (refer to N option).

N Non-initial load. On a non-initial 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 reversed by 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 non-initial 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 CIR only. 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

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 = nine track

Specifies a 9-track archive tape. Default value is 7-track.

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 = no rewind

The following rewinds are automatic.

	Rewind Before Processing	Rewind After Processing
PFDUMP	X	
PFLOAD	X	X
PFATC	X	X
PFCOPY	X	X

The NR parameter cancels out all rewinds for the operation in which it is specified.

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

When PFDUMP has finished creating an archive tape, it automatically unloads that tape; that is, the whole physical tape is returned to the take-up 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 = select number of files to skip

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 = select number of files to process

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 = select date

Specifies the date to be used with C, A, M, or B option. Format is yymmdd. Value is in octal. Default is zero.

TM = select time

Specifies the time to be used with C, A, M, or B option. Format is hhmmss. Value is in octal. Default is zero.

UI = select 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 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 = select destination user index

When specified, all files being processed by PFLOAD are loaded to this user index. The parameter is assumed octal unless D radix or nonoctal digit is used. Default is zero.

PF = select permanent file name

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

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 = select verify file

Indicates to PFDUMP that it is to produce a verification file that is a duplicate of the archive file it creates. 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.

```
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). 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

This parameter specifies the 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 = select 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 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 = select 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. Default is no name. Normally, the archive files are copied as individual files retaining their permanent file name. 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 = select user number

This parameter specifies the 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.

2.4 PERMANENT FILE UTILITY ROUTINES

This section and its subsections describe the five permanent file utility routines. The parameters available to each of the routines are listed in Table IV-2-1.

TABLE IV-2-1. PERMANENT FILE UTILITY PARAMETERS

	PFLOAD	PFDUMP	PFCAT	PFATC	PFCOPY
FN = family name	FN	FN	FN		
PN = pack name	PN	PN	PN		
DN = device number	DN	DN	DN		
T = archive file name	T	T		T	T
LO = list option	LO=	LO=	LO=	LO=	LO=
T - files processed	T	T	T	T	T
C - catalog files	C	C		C	
E - errors	E	E	E	E	E
S - summary			S		
L = output file name	L	L	L	L	L
OP = utility option	OP=	OP=	OP=	OP=	OP=
C - creation	C	C	C	C	C
A - last access	A	A	A	A	A
M - last modification	M	M	M	M	M
I - indirect	I	I	I	I	I
D - direct	D	D	D	D	D

TABLE IV-2-1. PERMANENT FILE UTILITY PARAMETERS (Cont'd)

	PFLOAD	PFDUMP	PFCAT	PFATC	PFCOPY
	OP=	OP=	OP=	OP=	OP=
B - before date and time	B	B	B	B	B
P - purge after dump		P			
R - replace	R				
N - noninitial	N				
E - extract CIR	E				
O - omit CIR	O				
Q - catalog and permit records					Q
NT = nine track	NT	NT		NT	NT
DE = density	DE	DE		DE	DE
NR = no rewind	NR	NR		NR	NR
NU = no unload		NU			
SF = number of files to skip	SF	SF		SF	SF
N = number of files to process	N			N	N
DT = date	DT	DT	DT	DT	DT
TM = time	TM	TM	TM	TM	TM
UI = user index	UI	UI	UI	UI	UI
PF = permanent file name	PF	PF	PF	PF	PF
DI = destination user index	DI				
VF = verify file name		VF			
V = verify		V			
DD = destination device number	DD				
MF = master file name					MF
UN = user number	UN	UN	UN	UN	UN

2.4.1 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 2.4.2) or its permanent file copies can be selectively accessed by other utilities for cataloging and copying.

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 2.4.2.

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-2-2 shows the defining characteristics of the three kinds of dump.

TABLE IV-2-2. PFDUMP TYPES

Selective Dump	Partial Dump	Full Dump
OP=M	OP=C A I D P O, DN=$^{0}_{\text{master device}}$	OP=0, DN=$^{0}_{\text{master device}}$

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.

59	47	35	17	0
Permanent File Name			User Index	
Device Number	Access Count	Access Date and Time		

The first word of the entry describes the file uniquely in the system. In the second word, the device number is used only to identify direct access files. The access count is the number of times this file was accessed as of the access date and time.

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.

2.4.2 PERMANENT FILE LOAD UTILITY TABLE (PFLOAD)

Archive files produced by the PFDUMP utility (section 2.4.1) 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.

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 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.

Example:

In this example, permanent files created and modified on three devices are dumped nine times. The action runs from May 1, 1973 (73.5.1) to May 10, 1973 (73.5.10). 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.

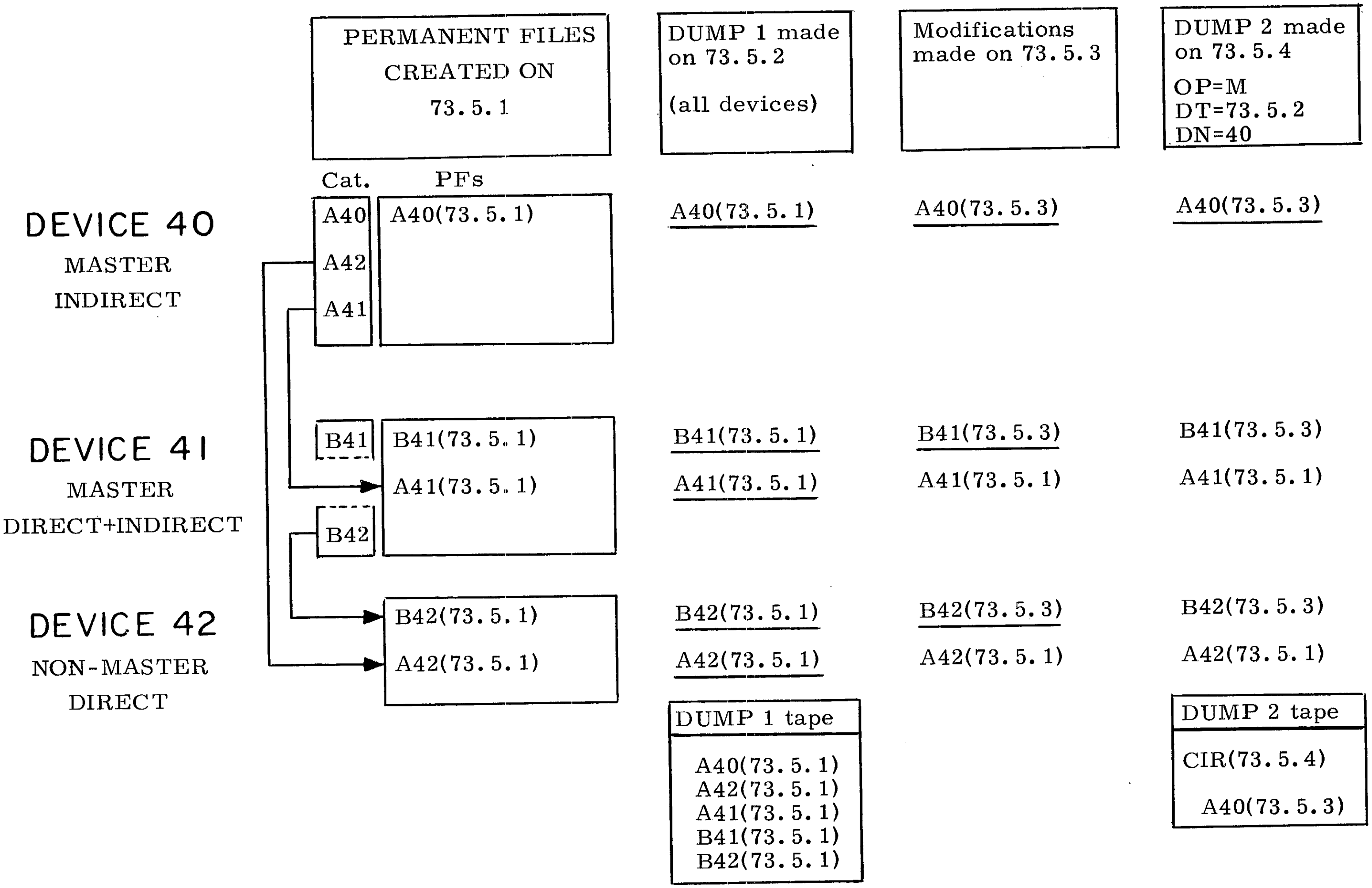
DEVICE 41 A master device containing indirect and direct access files. Direct access files cataloged on this device may be written on 42 or this device.

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.

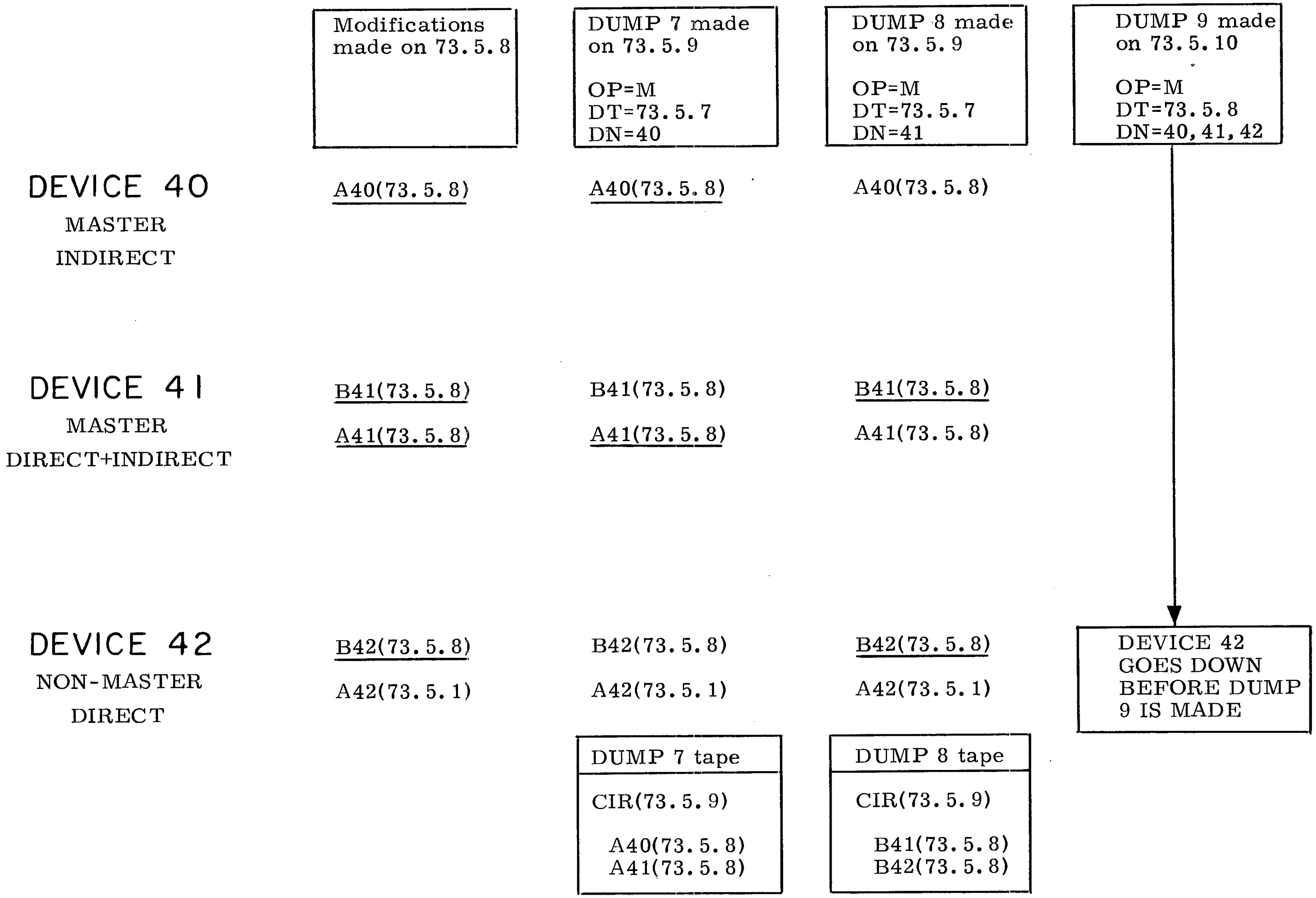

PERMANENT FILES CREATED ON 73.5.1
DUMP 1 made on 73.5.2 (all devices)
Modifications made on 73.5.3
DUMP 2 made on 73.5.4 OP=M DT=73.5.2 DN=40
Cat.
PFs
DEVICE 40
MASTER
INDIRECT
A40
A42
A41
A40(73.5.1)
A40(73.5.1)
A40(73.5.3)
A40(73.5.3)
DEVICE 41
MASTER
DIRECT+INDIRECT
B41
B42
B41(73.5.1)
A41(73.5.1)
B41(73.5.1)
A41(73.5.1)
B41(73.5.3)
A41(73.5.1)
B41(73.5.3)
A41(73.5.1)
DEVICE 42
NON-MASTER
DIRECT
B42(73.5.1)
A42(73.5.1)
B42(73.5.1)
A42(73.5.1)
B42(73.5.3)
A42(73.5.1)
B42(73.5.3)
A42(73.5.1)
DUMP 1 tape
A40(73.5.1)
A42(73.5.1)
A41(73.5.1)
B41(73.5.1)
B42(73.5.1)
DUMP 2 tape
CIR(73.5.4)
A40(73.5.3)

	DUMP 3 made on 73.5.4 OP=M DT=73.5.2 DN=41	DUMP 4 made on 73.5.4 OP=M DT=73.5.2 DN=42	Modifications made on 73.5.5	DUMP 5 made on 73.5.6 OP=M DT=73.5.4 (all devices)	DUMP 6 made on 73.5.7 OP=M DT=73.5.4 DN=42
DEVICE 40 MASTER INDIRECT	A40(73.5.3)	A40(73.5.3)	A40(73.5.5)	A40(73.5.5)	A40(73.5.5)

DEVICE 41
MASTER
DIRECT+INDIRECT
	B41(73.5.3)
	A41(73.5.1)
		B41(73.5.3)
		A41(73.5.1)
			B41(73.5.5)
			A41(73.5.1)
				B41(73.5.5)
				A41(73.5.1)
					B41(73.5.5)
					A41(73.5.1)

DEVICE 42
NON-MASTER
DIRECT
	B42(73.5.3)
	A42(73.5.1)
		B42(73.5.3)
		A42(73.5.1)
			B42(73.5.5)
			A42(73.5.1)
				B42(73.5.5)
				A42(73.5.1)
					B42(73.5.5)
					A42(73.5.1)

	DUMP 3 tape
	CIR(73.5.4)
	B41(73.5.3)
	B42(73.5.3)
		DUMP 4 tape
		CIR(73.5.4)
		B42(73.5.3)
				DUMP 5 tape
				CIR(73.5.6)
				A40(73.5.5)
				B41(73.5.5)
				B42(73.5.5)
					DUMP 6 tape
					CIR(73.5.7)
					B42(73.5.5)



	Modifications made on 73.5.8	DUMP 7 made on 73.5.9
		OP=M
		DT=73.5.7
		DN=40
			DUMP 8 made on 73.5.9
			OP=M
			DT=73.5.7
			DN=41
				DUMP 9 made on 73.5.10
				OP=M
				DT=73.5.8
				DN=40,41,42

DEVICE 40 MASTER INDIRECT	A40(73.5.8)	A40(73.5.8)	A40(73.5.8)	

DEVICE 41
MASTER
DIRECT+INDIRECT
	B41(73.5.8)
	A41(73.5.8)
		B41(73.5.8)
		A41(73.5.8)
			B41(73.5.8)
			A41(73.5.8)	

DEVICE 42
NON-MASTER
DIRECT
	B42(73.5.8)
	A42(73.5.1)
		B42(73.5.8)
		A42(73.5.1)
			B42(73.5.8)
			A42(73.5.1)
				DEVICE 42 GOES DOWN BEFORE DUMP 9 IS MADE

		DUMP 7 tape
		CIR(73.5.9)
		A40(73.5.8)
		A41(73.5.8)
			DUMP 8 tape
			CIR(73.5.9)
			B41(73.5.8)
			B42(73.5.8)	

Actions during the 10 days in May in which the dumps and modifications were made runs as follows:

Date — Action

73.5.1 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.

73.5.2 DUMP 1. A full permanent file system dump. No CIR is written on the archive file.

73.5.3 Three files (A40, B41, and B42) are modified on this date. This is indicated by the modification date in parentheses following the file name.

73.5.4 DUMP 2. This is a selective dump of device 40 that specifies permanent files modified are 73.5.3. 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 73.5.3, qualifies for this dump. It is written on an archive tape after a CIR for this date (73.5.4). The CIR gives a current description of all five files in the permanent file system.

73.5.4 DUMP 3. This is a selective dump of device 41 that specifies all files modified after 73.5.2. The catalog for this device contains entries for files B41 and B42. Both files were modified on 73.5.3 and qualify for this dump. They are written on an archive file after a CIR for this date.

73.5.4 DUMP 4. This is a selective dump of device 42 that specifies all files modified after 73.5.2. The PFDUMP utility scans the catalogs on devices 40 and 41 to locate the files on 42. One file, B42 (75.5.3), falls in this category.

73.5.5 Three files are modifed on this date.

73.5.6 DUMP 5. This is a selective dump of each of the three devices. All files modified after 73.5.4 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.

73.5.7 DUMP 6. This selctive dump of device 42 specifies all files modified after 73.5.4. Since 42 is nonmaster, the dump routine scans the catalogs of the other two devices.

Date	Action
73.5.8	Four files are modified on this date.
73.5.9	DUMP 7. A selective dump of device 40 is made with DT=73.5.7. Two files qualify.
73.5.9	DUMP 8. A selective dump of device 41 is made with DT=73.5.7. 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 73.5.7 and are dumped.
73.5.10	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(73.5.9)

B41(73.5.8)
B42(73.5.8)

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(73.5.9)

A40(73.5.8)
A41(73.5.8)

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(73.5.7)
B42(73.5.5)
	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(73.5.6)
A40(73.5.5)
B41(73.5.5)
B42(73.5.5)
	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(73.5.4)
B42(73.5.3)
	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(73.5.1)	This tape, produced by a full dump, contains all the files that were on the permanent file system when the dump was made (73.5.1). These files are checked against CIR (73.3.9). 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 73.5.9 when the last selective dump was made.

2.4.3 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 format of the directory depends on the parameters selected (refer to Table IV-2-1, section 2.4).

Figure IV-2-9 gives the beginning of a sample directory which was produced with the command

```
PFCAT(DN=1,L0=T,UN=ROBERTR)
```

2.4.4 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. The format of the directory depends on the parameters selected (refer to Table IV-2-1, section 2.4).

Figure IV-2-10 gives the beginning of a sample directory which was produced with the command

```
PFATC(L0=T,UI=1)
```

2.4.5 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 parameters available for this utility are given in Table IV-2-1, section 2.4.

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.

```
        73/06/13.              KRONOS CATALOG FILE                      08.25.57.
DIRECTORY OF PERMANENT FILE DEVICE       1                        PAGE          3
CATALOG OF USER INDEX         144 ROBERTR
   FILE NAME ACCESS FILE-TYPE LENGTH OR CREATION LAST ACCESS LAST MOD
    PASSWORD MD/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
   1 PROFLO   DIR.  PRIVATE          5   * 73/04/28.  73/06/13.   73/04/28.
                 29          WRITE         09.48.23.  00.26.26.   09.48.23.
   2 PROFILO  DIR.  PRIVATE         55   * 73/05/11.  73/05/11.   73/05/11.
                  1          WRITE         07.19.21.  07.42.07.   07.19.21.
                                       .
                                       .
                                       .
```

Figure IV-2-9. Sample Directory Produced by PFCAT

```
P.F. ARCHIVE CATALOG OF - PFDUMP                                PAGE        1
        REEL -     1      EQ. MASK - 377                 73/06/13. 01.30.12.
FILE NAME ACCESS FILE-TYPE LENGTH OR CREATION LAST ACCESS LAST MOD
  PASSWORD MD/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
3 CATLG4     IND.  PRIVATE          1   73/03/17.   73/05/01.   72/05/01.
               4      144  WRITE        06.33.05.   09.32.38.   09.32.38.
4 REC13      IND.  PRIVATE          1   73/06/08.   73/06/08.   73/06/08.
               0      144  WRITE        06.14.07.   07.21.39.   07.21.39.
```

Figure IV-2-10. Sample Directory Produced by PFATC

2.5.2 PERMANENT FILE UTILITY PROCESSOR (PFU)

DAYFILE MESSAGES

PFU - PARAMETER ERROR.

Data in PFU call in error.

DUPLICATE FILE NAME.

File already used at control point.

REQUESTED FILE NOT AVAILABLE.

FNT Entry for file requested not found in FNT.

EQXX TRACK LIMIT.

No remaining tracks available on EQxx.

BUFFER CONTROL WORD ERROR.

Word count of sector to be read from central memory exceeds word count limit of a sector (100B).

DEVICE NOT INITIALIZED.

PFLOAD attempting to load to uninitialized master device.

NO DEVICE SPECIFIED.

Device number 0.

FET POINTERS OUT OF BOUNDS.

Out pointer is greater than limit pointer.

ALTERNATE DEVICE NOT FOUND.

PFU is unable to locate an alternate device in the system.

CATALOG TRACK NOT FOUND.

In attempting to interlock a catalog track, PFU could not find track.

OPERATOR DROP.

If PFU is unable to clear the utility interlock, the operator may drop the job.

NOT SYSTEM JOB.

Calling program is not system origin or does not have system origin privileges with debug set on.

2.5.3 PERMANENT FILE DUMP OVERLAY

INFORMATIVE OPERATOR MESSAGES

P. F. DEVICE (DEVICNM) DUMPED.

DUMPING (FILENAME) (USER INDEX)

PFDUMP YY/MM/DD. HH. MM. SS.

PFDUMP DEVICE (DEVICENM) FAMILY (FAMILYNAM).

PFDUMP DEVICE (DEVICENM) PACK (PACKNAM).

PFDUMP DEVICE MASK XXX.

DUMPING - DIRECT ACCESS FILES ONLY.

DUMPING - INDIRECT ACCESS FILES ONLY.

FILENAME XXXXXXX USER INDEX XXXXXX.

ALL FILES FOR USER INDEX XXXXXX

ACCESSED AFTER YY/MM/DD. HH. MM. SS.

CREATED AFTER YY/MM/DD. HH. MM. SS.

MODIFIED AFTER YY/MM/DD. HH. MM. SS.

NO FILES PROCESSED.

CHECK DAYFILE FOR ERRORS.

WAIT FOR CATALOG INTERLOCK.

GENERATING CATALOG IMAGE.

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 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 skipped.

STATUS ERR (LFN) (U.I.).

Status error on DAF (lfn). File 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.

NO USER INDEXES ON TAPE MATCH DEVICE MASK.

ILLEGAL CATALOG TRACK COUNT.

DESTINATION DEVICE NOT FOUND.

USER INDEX NOT ON DEVICE.

DEVICE SPECIFIED NOT FOUND.

NO PF DEVICE IN EST.

ERROR IN CATALOG IMAGE ON PFDUMP TAPE.

PARITY ERROR IN CATALOG IMAGE RECORD.

ANY LOADING TO THIS POINT IS
INCOMPLETE A REINITIALIZE
AND RELOAD IS REQUIRED.

PFLOAD ABORTED.

INFORMATIVE OPERATOR MESSAGES

PFLOAD DEVICE (NN) FAMILY (FAMILYNAM)

PFLOAD DEVICE (NN) PACK (PACKNAM)

PACKNAME (PACKNAM) LOADED.

P. F. DEVICE (DEVICNM) LOADED.

LOADING FROM (ARMASK) TO (DVMASK).

LOADING (FILENAM) (USERIN).

READING (FILENAM) (USERIN).

ALL FILES FOR USER INDEX (USERIN).

FILENAME (FILENAM) USER INDEX (USERIN).

CREATED AFTER YY/MM/DD. HH. MM. SS.

MODIFIED AFTER YY/MM/DD. HH. MM. SS.

ACCESSED AFTER YY/MM/DD. HH. MM. SS.

LOADING - INDIRECT ACCESS FILES ONLY.

LOADING - DIRECT ACCESS FILES ONLY.

..WAIT FOR TOTAL INTERLOCK..

NO FILES PROCESSED.

CHECK DAYFILE FOR ERRORS.

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.

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 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 the KRONOS Time-Sharing System, 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 (TELEX) at control point 1. Under TELEX direction, communication can be made with other control points. The transaction terminal, under TELEX direction, extends communication to the Transaction Executive (TRANEX) at control point 2. Processing under TRANEX makes use of the Transaction Subsystem, an extension of the KRONOS 2.1 operating system.

From a time-sharing terminal, the user enters application programs and data to be processed in conjunction with whatever permanent files he may have established. From a transaction terminal, this basic activity is greatly enhanced by the added capabilities of data management, a library of tasks, and a preestablished data base. The data management structures use data, both from the data base and from an input transaction. The tasks are application programs employed by a particular user throughout many transactions. Tasks are called by the transaction executive to subcontrol points within the field length of the transaction executive (control point 2). A data base is a user-specified superstructure of data on mass storage. This data base is accessed by the data manager when referenced by a task or an input program. Figure IV-3-1 gives an overview of the Transaction Subsystem. The Transaction Subsystem Reference Manual gives a detailed treatment.

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. TELEX and TRANEX use these tables to put the network on the air and exercise its capabilities.

3.2 DIRECTIVE FORMATS

A directive card may be up to 90 columns but only columns 1 through 72 interpreted. Any card containing an asterisk in column 1 is a comment card and is ignored.

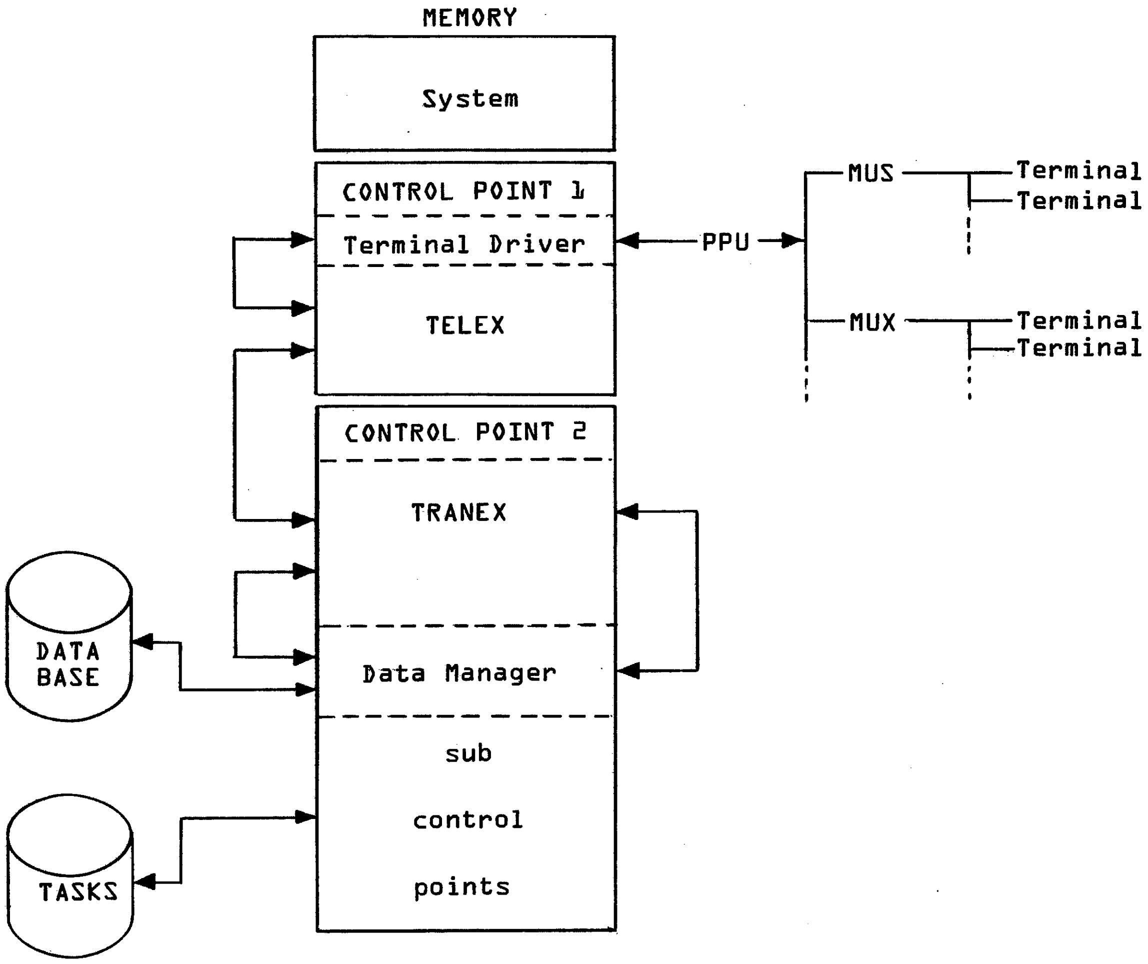


Figure IV-3-1. Organization of the Transaction Subsystem

3.2.1 MULTIPLEXER DEFINITION DIRECTIVE

The format of the multiplexer definition directive is:

```
//6671,CH=n1,EQ=n2.

          or

//6676,CH=n1,EQ=n2.
```

depending on whether a 6671 or 6676 multiplexer is used. The CH equation gives the channel to which the multiplexer is attached, and the EQ equation gives the equipment number of the multiplexer. The channel range (n1) is 0 to 13B and 20B to 33B. The equipment number range (n2) is 0 to 7.

3.2.2 TERMINAL DEFINITION DIRECTIVE

The format of the terminal definition directive is:

```
/termnam,PN=n1,pi=ni,...
```

termnam — Mandatory, unique alphanumeric name (1 to 7 characters) that identifies the terminal

PN — Mandatory port number for which n1 has the decimal ranges 1 through 16 (6671 multiplexer) or 1 through 64 (6676 multiplexer)

The pi=ni have the following possible values.

Keyword	Description
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) or PN+1 to 63 (6676 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.

Keyword — Description

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).

A KRONOS Transaction Subsystem (KTS) terminal is defined with the previous arguments as well as the following which are unique to the Transaction Subsystem.

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.

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 16777215 (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 that was created with the common deck COMCRTN. 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 sytem file will be either SIMFILE or NETWORK depending on its use. Typical coding would be:

```
X.DIS

ACCOUNT,ANLST.

SUI=377777.

DEFINE,NETWORK/CT=L.

COPY(INPUT,NETWORK)
```

The control card used to call the VALNET program has the following format.

```
VALNET(p1,p2,p3)
```

pi

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 NETWORK.

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	
Card 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 TERMINAL DESCRIPTION ERROR.
```

List lines are formatted for 72 columns except for input lines that exceed this length.

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 card 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

Deadstart procedures can dump all PPUs, central memory, and (optionally) extended core storage to a printer or to magnetic tape. The dump to magnetic tape is via the Express Deadstart dump (EDD). This tape can be read and listed at a future date by the express deadstart dump interpreter (DSDI). Information on the tape is sequenced according to Figure IV-4-1. The listings are formatted for a 136-column printer.

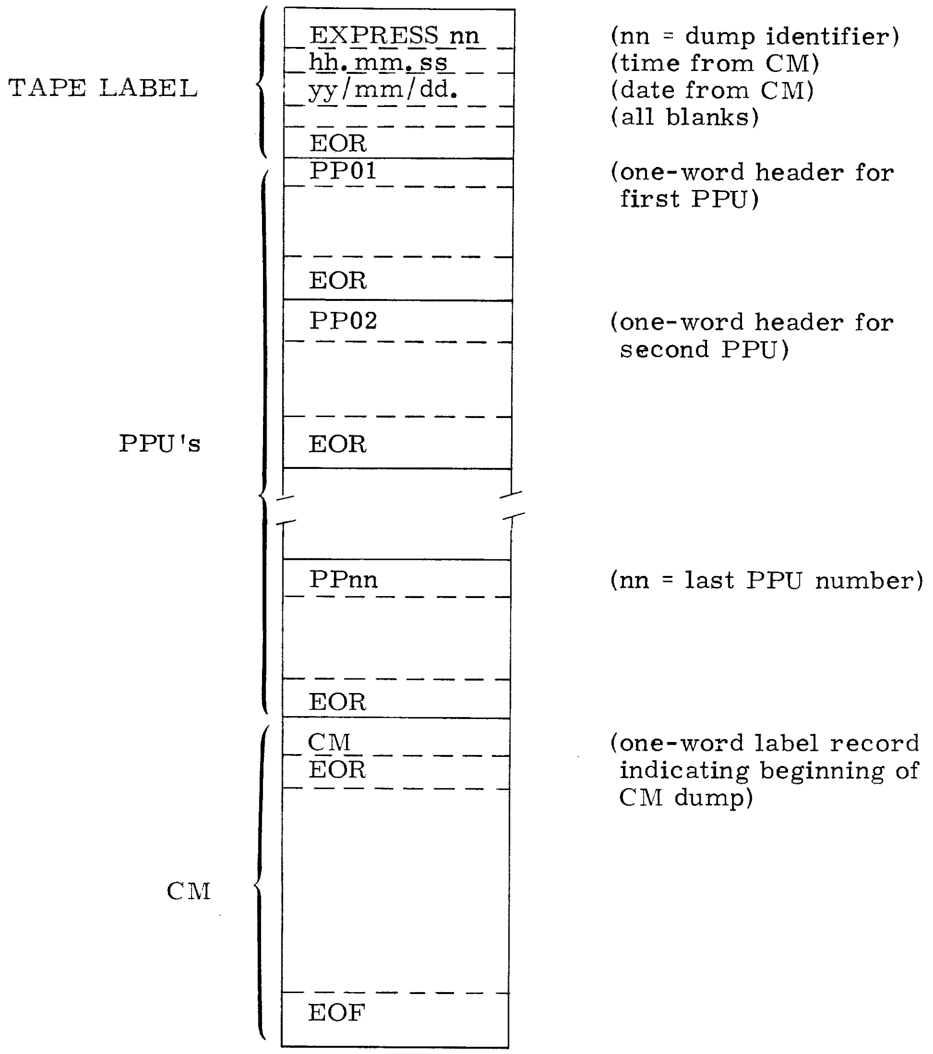
4.1 CALLING THE EXPRESS DEADSTART DUMP INTERPRETER

Processing of the express deadstart dump tape is initiated with the DSDI control card. The format of this card is:

```
DSDI(p1,p2,...,pn)
```

where each pi is either a keyword or a keyword equated to a value. The following is a listing of the keywords and values available.

Keyword	Definition
Q or none	Dumps all PPUs and central memory up through address 20000B. The PPUs will be in word format, and central memory will be in byte format.
F	Dumps all PPUs, central memory, and extended core storage. The PPUs will be in word format; central memory and extended core will be in byte format.
P	Dump the 10 lower PPUs (1 through 11B). Printout is in word format.
P=N	Dump PPU number N. Printout is in word format. This designation can be repeated within a single call. Example: DSDI(P=0, P=4) will produce a printout of PP00 and PP04.
U	Dump the 10 upper PPUs (20B through 31B). Printout is in word format.
C	Dump all of central memory in byte format.
C=L	Dump central memory from address 0 to address L in byte format.



NOTE

If ECS is present, it follows CM.

Figure IV-4-1. Dump Tape Format

Keyword Definition

C=F-L Dump central memory from address F to address L in byte format.

To dump central memory in display code rather than byte format, the previous three calls become:

CD

CD=L

CD=F-L

E Dump all of extended core in byte format.

E=L Dump extended core from address 0 to address L in byte format.

E=F-L Dump extended core from address F to address L in byte format.

To dump extended core in display code rather than byte format, the previous three calls become:

ED

ED=L

ED=F-L

EF Ensure that end of file is reached after dump.

PR List PPU registers in word format.

CM=N List control point N memory (that segment of CM containing the data from control point N). The listing is in byte format.

CMD=N List control point N memory (that segment of CM containing the data from control point N). Listing is in display code.

All numeric values can be specified as octal or decimal with a B or D radix immediately following the number. If no radix is given, the number is octal.

Sample listings are given in Figure IV-4-2.

4.2 DIAGNOSTICS

4.2.1 DAYFILE MESSAGES

DSDI ARGUMENT ERROR

An unknown keyword was encountered.

DSDI NUMBERIC CONSTANT ERROR

A numeric value is not properly formed.

DSDI(P=4)

```
PP04    EXPRESS 00 06.03.00.  73/05/26          73/05/26. 06.22.54.

    00XX    0X    1X    2X    3X    4X    5X    6X    7X     01XX    0X    1X    2X    3X    4X    5X    6X    7X
      00  0003  0032  0030  6600  ----  3434  ----  0001       00  1477  3474  0141  2023  0100  5400  1401  5000
      01  1500  0002  7760  7600  0007  2030  ----  0100       01  1701  3051  0200  0620  0121  0324  0200  0633
      02  7304  7700  ----  ----  6100  ----  ----  1000       02  0576  1377  0424  0364  1014  2001  0336  5400
```

DSDI(C=100-200)

```
  CM DUMP FROM      100 TO    200  EXPRESS 00 06.04.00.  73/05/26.          73/05/26.  06.22.10.

100  0000 0000 0000 0000 0000   0000 0000 0000 0001 0000   0000 0000 0000 0000 0000   0100 0100 0100 0100 0100
104  0100 0100 0100 0100 0100   0100 0100 0100 0100 0100   0004 1651 0001 0204 7647   0000 0000 0000 0000 0000
110  0000 0000 0000 0000 0001   0000 0000 4513 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
114  0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
     DUPLICATED LINES.
```

DSDI(CMD=1)

```
CP  1 DUMP FROM     0 TO    6000   EXPRESS 00 06.03.00. 73/05/26.           73/05/26. 06.23.33.

   0 00000000000000000000 05160400000400000001 37770000000000000000 37770000000000000000        END D  A4;      4;
   4 37770000000000000000 37770000000000000000 37770000000000000000 37770000000000000000 4;      4;      4;      4;
     DUPLICATED LINES.
```

Figure IV-4-2. Sample Listings Under DSDI

DSDI ARGUMENT CANT BE EQUATED

The Q, F, or U argument is equated to a value.

DUMP LABEL ERROR.

First record on dump file is not four words long.

DUMP FILE EMPTY.

First read on dump file produced an EOF and/or EOI.

EXPRESS DUMP COMPLETE.

Dump completed normally.

INCOMPLETE DUMP.

Indicates any one of the following conditions:

CM not present but dump requested.
ECS not present but dump requested.
EOF and/or EOI encountered during read of PPUs.
FWA of CM or ECS dump not present.

DUMP FORMAT ERROR.

Incorrect PPU number appears on dump file.

DUMP FWA .GT. LWA.

CM or ECS dump bounds are illegal.

4.2.2 LISTING ERROR MESSAGES

CONTROL POINT MEMORY DUMP IMPOSSIBLE

The low core field NCPL is unreasonable.

PP COMMUNICATIONS DUMP IMPOSSIBLE

The low core field PPUL is unreasonable.

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 KRONOS 2.1 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 5.4.

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. When constructing a session file, terminate each entry with a teletypewriter carriage return, which is a V in display code (66 code). Note the following requirements.

1. Begin each session with the entry ANSWERBAC; this is a terminal type identifier to TELEX (the Time-Sharing Subsystem Executive).

2. Enter each session entry on a separate card.

3. Terminate each card entry with an 11-0 punch (display code 66).

4. After ANSWERBAC, enter a user number entry and a password entry.

5. Specify the time-sharing terminal commands and data.

6. End each session with the command BYE so that the login of the next session is performed correctly.

7. Enter an EOR after each session except for the last; enter an EOF, instead of an EOR, after the last session.

The following is an example of a session file.

5.2 SESSION FILE PROCESSING

Use the stimulator software to process the time-sharing session file. The procedure is:

1. There must be a Time-Sharing Subsystem Stimulator entry in the EST as described in part II, section 4.3.7.

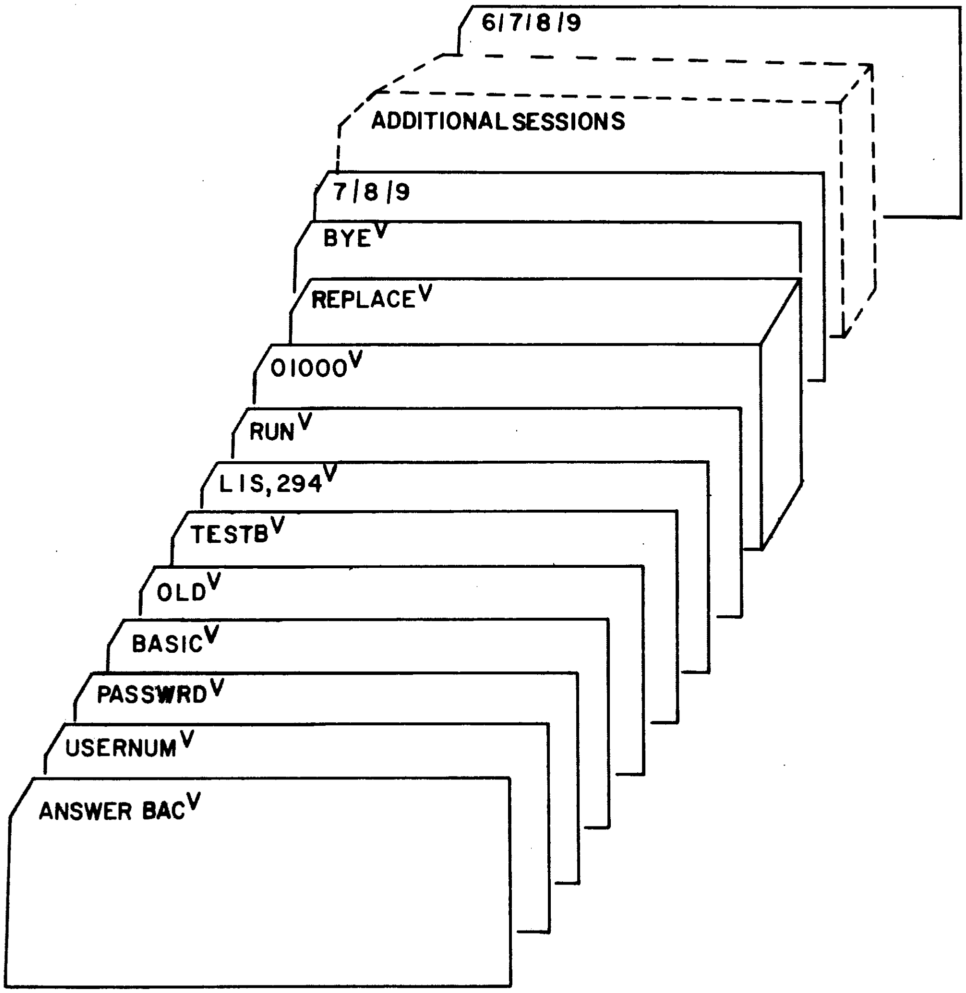


Figure IV-5-1. Sample Session File

V represents an 11-0 punch (display code 66).

2. Use DSD to bring the stimulator software to the control point 17.

   Type STIMULATOR
   Press CR

3. The stimulator requests the K display at control point 17.

4. Type K,17.
   Press CR

5. The following information appears on the left display.

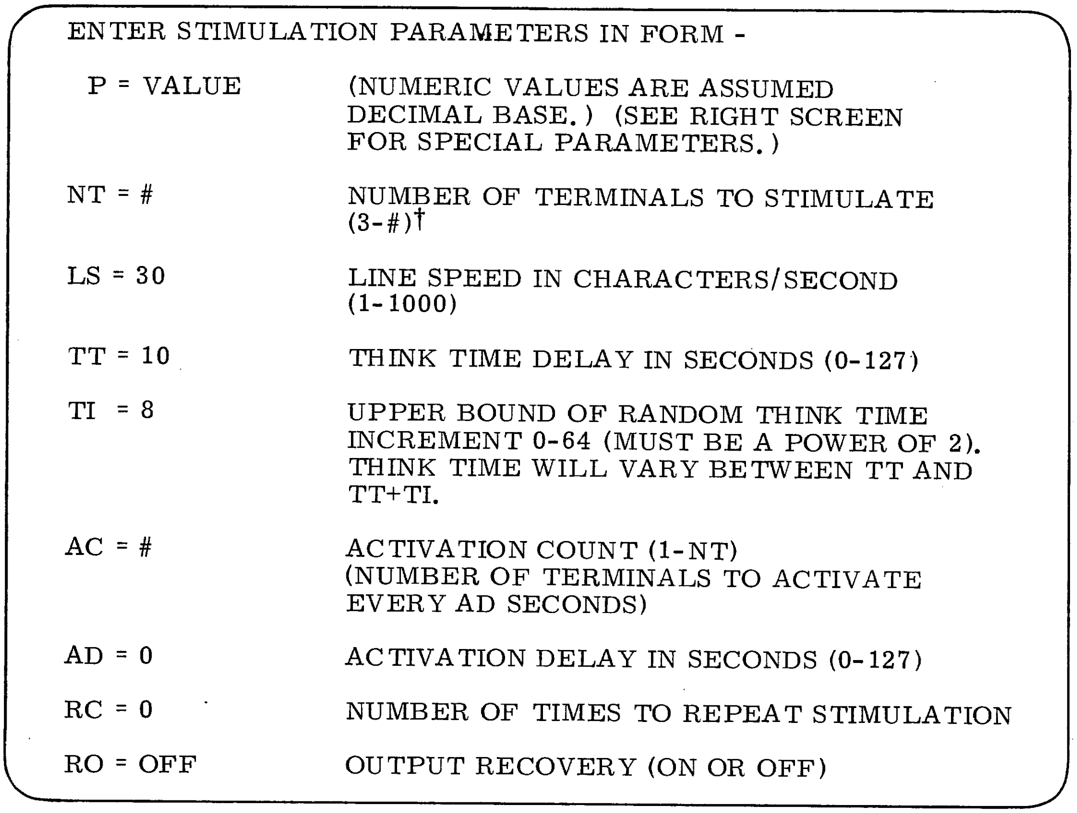
```
ENTER STIMULATION FILE PARAMETERS IN FORM -

   P = VALUE

   F =          SESSION FILE NAME

   UI = OB      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.
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.
K.GO	None	Indicates to the stimulator that all the parameters have been entered.

7. The following information appears on the left display.

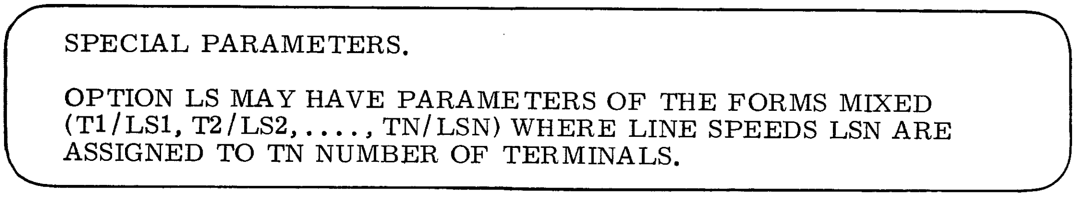


```
ENTER STIMULATION PARAMETERS IN FORM -

  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)
```

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 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=x	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 7777_8.
K.LS=y	30 characters/second	y is the line speed at which the operator wishes to run the terminals; 1 through 1000 characters per second is the range.
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:

59	47	11 0
terminal number		ASCII character

		When RO=OFF, the output is not recovered.
K.GO		Indicates to the stimulator that the parameters are entered.

The special parameter format is:

K.MIXED T1/x, T2/y, ... , Tn/n	Each terminal (T1, T2, Tn) is assigned the designated terminal speed (x, y, n); 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 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.
MEMORY OVERFLOW	F	There is not enough field length for the managed tables for STIMULA.	
MEMORY REQUEST ERROR	F	STIMULA 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 at control point.	Bring Time-Sharing Subsystem to control point before running the stimulator.

INDEX

COMMENT SHEET

MANUAL TITLE CONTROL DATA® KRONOS 2.1 Installation Handbook

PUBLICATION NO. 60407500 REVISION A

FROM: NAME:

BUSINESS ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

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

FOLD ON DOTTED LINES AND STAPLE

CUT ALONG LINE

PRINTED IN U.S.A

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

CUT ALONG LINE

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Technical Publications Department
4201 North Lexington Ave.
Arden Hills, Minnesota 55112

MD 220

FOLD

FOLD

CONTROL DATA
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.