CONTROL DATA
CORPORATION

CONTROL DATA®
CYBER 70/MODEL 76 COMPUTER SYSTEM
7600 COMPUTER SYSTEM

SCOPE 2
INSTALLATION HANDBOOK

REVISION RECORD

REVISION	DESCRIPTION
A	
12-72	Original printing.

Publication No.
60344000

© 1972
by Control Data Corporation
Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Technical Publications Dept.
4201 North Lexington Avenue
Arden Hills, Minnesota 55112

or use Comment Sheet in the back of this manual.

PREFACE

This manual provides information needed to install and modify the SCOPE 2.0 operating system. It is assumed that the analyst has at least six months of experience with a 7600 or CYBER 70 computer.

The SCOPE 2.0 Installation Handbook has the following format.

Part I	List of all materials (such as tapes and cards) released with SCOPE 2.0, its product set members, and stations. A flow chart and general description of the process of installing and modifying the SCOPE 2.0 operating system that includes the purchased product set and stations.
Part II	The procedures for deadstarting standard stations and SCOPE 2.0.
Part III	The standard installation options, procedures to modify, and detailed description of jobs necessary to modify each of the stations, product set members, and SCOPE 2.0.
Part IV	Information related to installation, such as installation cautions, installation messages, and memory requirements.

SCOPE 2.0 is intended to be installed and configured only as described in this installation handbook. Therefore, Control Data is only responsible for the proper functioning of features or parameters described in this manual.

CONTENTS

PART II

DEADSTART OF UNCONFIGURED STATIONS AND SYSTEM

PART III

INSTALLATION OF CONFIGURED SYSTEM AND PRODUCT SET

PART IV

ADDITIONAL INFORMATION

FIGURES

LIST OF PRODUCTS 1

1.1 RELEASED PRODUCTS FOR SCOPE 2.0

The following products are released with the SCOPE 2.0 operating system.

SCOPE 2.0 Maintenance Package

System Maintenance Monitor 3.0 (SMMB)

SCOPE 2.0

- ANALYZE
- LIBEDT
- Loader
- Record Manager
- UPDATE 2.0

COMPASS 2.0

Diagnostic Control Program (DCP)

One or more of the following stations is required.

CONTROL DATA 7611-1 I/O Station Version 2

CONTROL DATA 7611-2 Magnetic Tape Station Version 2

CONTROL DATA 6000 CYBER 70 Series SCOPE 3.4 Station Version 2

1.2 OPTIONAL PRODUCTS

The following products may also be purchased separately to be used with SCOPE 2.0.

FORTRAN Extended 2.0

FORTRAN Run 2.0

COBOL 1.0

Sort/Merge 1.0

RELEASE MATERIALS 2

2.1 SCOPE 2.0 MAINTENANCE PACKAGE

Tapes

MODTAPE — 7-track tape file with the following characteristics: labeled, 556 bpi, odd parity, BT=I, RT=W, and MRL=5120. MODTAPE*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 1 for more detail.

Partition 1	MODPL	Modifications to released system in an OLDPL format (*=/).
Partition 2	SYSDECKS	Installation, modification, and utility decks for SCOPE 2.0 in an OLDPL format (*=$).

2.2 SYSTEM MAINTENANCE MONITOR 3.0 (SMMB)

Materials other than the SMMB deck are available to the customer engineer through the SMM Software Availability Bulletin (SAB).

Cards

SCPSMM — SMMB binary deck configured for SCOPE 2.0.

2.3 7611-1 I/O STATION

Tapes

STALIB — 7-track tape file with the following characteristics: unlabeled, 800 bpi, odd parity, BT=C, RT=S, and MBL=5120; contains all 7611-1 system routines in binary format and is used to deadstart the station.

STAOPL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120; STAREL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 3 for more detail.

Partition 1	STAOPL	Source code in UPDATE format for STALIB.

Cards

STATDS — One deadstart binary card that initializes the station disk pack from the STALIB tape.

STADDS — One deadstart binary card that initializes the station from its disk pack.

2.4 7611–2 MAGNETIC TAPE STATION

Tapes

MTSLIB — 7-track tape file with the following characteristics: unlabeled, 556 bpi, odd parity, BT=I, RT=W, and MBL=5120; contains the binary deadstart for the station.

MTSOPL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120; MTSREL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 4 for more detail.

Partition 1	MTSOPL	Source code in UPDATE format for MTSLIB tape.

Cards

MTSTDS — A binary card deck of the program to be loaded into the FLPP to initialize the 7611-2 station disk pack from the MTSLIB tape.

MTSDDS — A binary card deck of the program to be loaded into the FLPP to initialize the 7611-2 station from the 7611-2 station disk pack. This program is also on the SCPLIB tape under program name FDS and is loaded by the SCOPE 2.0 system when the system is installed.

2.5 6000 OR CYBER 70 SERIES SCOPE 3.4 STATION

The SCOPE 3.4 station release materials are described in the 6000 SCOPE 3.4 Installation Handbook, Pub. No. 60307400.

2.6 SCOPE 2.0

Tapes

SCPLIB — 7-track tape file with the following characteristics: unlabeled, 556 bpi, odd parity, BT=C, RT=W, and MBL=5120; contains the SCOPE 2.0 operating system and is used to deadstart SCOPE 2.0. This tape contains the binary of UPDATE, COMPASS, and diagnostic control package.

SCPOPL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. SCPOPL2.0 is the file identifier in the HDR1 label. Refer to part III, section 6 for more detail.

Partition 1	SCPOPL	Source code in UPDATE format for the SCPLIB tape. The source for UPDATE is in this partition, but COMPASS and the diagnostic control package have separate OPLs.
Partition 2	DBUGBIN	Binary code for the DBUGLIB system library.

Cards

SCPSID — SCOPE 2.0 deadstart deck. Only the following binary decks of DS1 and the FLPP drivers are included in the release materials. The other cards are installation dependent and are described in part II, section 6.2.

The following SCPSID binary decks are released.

DS1	Deadstart binary deck
CPL	FLPP binary deck for 6000 SCOPE 3.4 station
DSC	FLPP binary deck for 7611-2 magnetic tape station
DSK	FLPP binary deck for system mass storage device
MMD	FLPP binary deck for on-line tape unit

2.7 COMPASS 2.0

Tape

CM2REL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. CM2REL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 7 for more detail.

Partition 1	CMPOPL	Source code in UPDATE format for CMPBIN.
Partition 2	CMPBIN	Absolute binary image of COMPASS assembler.

2.8 FORTRAN EXTENDED 2.0

Tape

FN2REL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. FN2REL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 8 for more detail.

Partition 1	FTNOPL	Source code in UPDATE format for FTNBIN.
Partition 2	FTNBIN	Absolute overlay of FORTRAN Extended.

2.9 FORTRAN RUN 2.0

Tape

RN2REL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. RN2REL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 9 for more detail.

Partition 1	RUNOPL	Source code in UPDATE format for RUNBIN.
Partition 2	RUNBIN	Absolute overlay of the FORTRAN Run compiler.

2.10 FORTRAN OBJECT TIME ROUTINES

Tape

FCLOPL — 7-track tape file (SCOPE 3.4 tape) with the following characteristics: labeled, 556 bpi, odd parity, BT=C, RT=S, and MBL=5120. FTNLIBS*3P4 is the file identifier in the HDR1 label. Refer to part III, section 10 for more detail.

Partition 1	FCLOPL	Source code in UPDATE format for the system libraries FORTRAN and RUNLIB.

2.11 COBOL 1.0

Tape

COBREL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. COBREL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 11 for more detail.

Partition 1	COBOPL	Source code in UPDATE format for COBREL and COBBIN.
Partition 2	COBBIN	Absolute overlay of the COBOL compiler.
Partition 3	COBREL	COBOL relocatable binary subroutines.

2.12 SORT/MERGE 1.0

Tape

SRTREL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. SRTREL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 12 for more detail.

Partition 1	SRTOPL	Source code in UPDATE format for SRTBIN, SRTREL, and SRTMAC.
Partition 2	SRTBIN	Absolute overlay of Sort/Merge.
Partition 3	SRTREL	Sort/Merge relocatable binary subroutines.
Partition 4	SRTMAC	Sort/Merge macros in XTEXT format.

2.13 DIAGNOSTIC CONTROL PROGRAM

Tape

DIAREL — 7-track tape file with the following characteristics: labeled, 800 bpi, odd parity, BT=I, RT=W, and MBL=5120. DIAREL*SCP2.0 is the file identifier in the HDR1 label. Refer to part III, section 13, for more detail.

Partition 1	DIAOPL	Source code in UPDATE format for DCP and diagnostic programs CT73 and MEMC.
Partition 2	DIABIN	Absolute binaries of decks DCP, CT73, and MEMC.

SUMMARY OF PROCEDURES TO INSTALL OR MODIFY A CONFIGURED SYSTEM AND PRODUCT SET 3

The installation and modification of the SCOPE 2.0 operating system involves five general steps. These steps are summarized in this section and are presented in detail in part III, section 1. The sequence is also reflected in the flowchart in Figure I-3.2.

	General Procedure	Summary, Part I	Detail, Part III
1.	Establish base system materials from the materials issued at release time.	section 3.1	section 1.2
2.	Build a production release system that contains the customer's product set either as released or modified according to installation parameters.	section 3.2	section 1.3
3.	Apply customer modifications, if any, to materials created in step 2.	section 3.3	section 1.4
4.	Create back-up copies of production materials that contain the product set and customer modifications.	section 3.4	section 1.5
5.	Generate maintenance information.	section 3.5	section 1.6

3.1 ESTABLISH BASE SYSTEM MATERIALS

To establish base system materials the following tapes and decks must be available and at least PRDLIB must be cataloged as a permanent file.

The heart of the procedures to install or modify the system is the set of decks called SYSDECKS in partition 2 of MODTAPE (part III, section 1). These decks require that tapes, decks, and permanent files with special names, labels, permissions, and IDs be used to install a customer's system. If these materials do not exist, the person installing the system must either create them or modify the contents of SYSDECK.

3.1.1 TAPES NECESSARY FOR A BASE SYSTEM

Release materials for operating system and products as described in part I, section 2.

A deadstart tape (SCPLIB) containing at least UPDATE and COMPASS.

A LOADPF/DUMPF tape containing permanent files (cataloged with ID=PRDLIB) for product set libraries, product set core image binaries, and operating system core image binaries, (part III, section 1.2.2).

3.1.2 DECKS NECESSARY FOR A BASE SYSTEM

SMMB deck to install SMMB 3.0 at the MCU (part II, section 2)

SCPSID, the deadstart deck for SCOPE 2.0 (part II, section 6.2)

Deadstart cards for all stations at the customer site (part II, sections 3.1, 4.1, and 5.1)

Any required decks from SYSDECK (part III, section 1.2.3)

3.1.3 PROCEDURES TO BUILD BASE MATERIALS

Following is an overview of procedures to create a base system.

1. Construct the SCPSID deck (part II, section 6.2).
2. Deadstart SMMB, available stations, and SCOPE 2.0 as they are released (part II, sections 2.2, 3.2, 4.2, 5.2, and 6.1).
3. Use MODCAT to catalog MODPL and SYSDECK, and list SYSDECKS MEMO at the same time (part III, section 1.2.3).
4. Use PUNCHDKS to punch SYSDECK decks to be used to install or modify the system (part III, section 1.2.3).
5. For an initial installation, run the OPRDLIB job to catalog COMPASS and DCP from the SCPLIB tape as a dummy PRDLIB PF.
6. For subsequent installations, run the LDIDPRD job to execute a LOADPF on the ID=PRDLIB DUMPF tape (part III, section 1.6). Or, if in a production environment (running jobs), verify that files are cataloged using AUDIT (part III, section 1.2.4).

3.1.4 CAUTION IF BUILDING A SYSTEM DURING PRODUCTION

The procedures in this handbook assume that the installation or modification does not occur during production time. Some SYSDECK jobs create files that are used by other jobs. These permanent files may conflict with production permanent files. For example, the FTN object library permanent file causes problems if a SYSLIBE job that attaches the highest cycle is run.

The detailed procedures in part III, section 1 contain batch notes that indicate possible problems and ways to avoid these problems. The customer must evaluate each of the solutions in terms of his special circumstances.

3.2 BUILD A PRODUCTION RELEASE SYSTEM

A production release system contains the operating system, as released or with modified installation parameters (part III, section 6.2). It also contains products the customer has purchased, either as they are released or modified according to product configuration parameters in part III.

The basic steps in building the production release system are:

1. Run the xxxMOD or xxxGEN job (xxx is the product abreviation) to update, assemble, and catalog the binaries for the products purchased by the customer and for which modifications exist on the MODPL (part III, sections 1.3.1 and 1.3.2). Refer to part I, section 3.6 for product dependencies.

   Or, run the xxxINS job to copy the product's binary from the release tape and catalog the binary (part III, sections 1.3.1 and 1.3.2).

2. For subsequent installations, run the FTNLIB7 job to update the old FCLOPL tape and to create a new FCLOPL tape (part III, section 1.3.3).

3. Run the NPRDLIB job to update the PRDLIB file. Binaries for products not modified in current PSR summary are retained from the previous system (part III, section 1.3.2).

4. Run the FLIBGEN and RLIBGEN jobs to update, assemble, and catalog the binary for the FTN and RUN object libraries (part III, section 1.3.3).

5. Run the SCPMOD or SCPGEN job to modify SCOPE 2.0, create a library of SCOPE 2.0 code, add the product set from PRDLIB, and create a new deadstart tape (part III, section 1.3.4).

6. Run the DBUGINS job to catalog the DBUGLIB permanent file (part III, section 1.3.4).

3.3 ADD CUSTOMER MODIFICATIONS

The procedures in SYSDECK assume that customer modifications are contained on a permanent file called USERMODPL, which is an OLDPL (with the master control character set to /) with one deck for each released product's program library (that is, one deck for the COMPASS program library, one deck for the FTN program library, etc.). SYSDECK contains a set of xxxUSR decks that update the release materials created by the xxxMOD decks. FTNLIBX adds customer modifications to the FORTRAN object libraries. The procedure to include customer modifications is similar to the procedure to apply PSR modifications.

3.4 CREATE BACK-UP MATERIALS

When the final tapes and decks are created, make copies of tapes and decks so that back-up materials are available. There are several SYSDECK jobs to copy the tapes and decks (part III, section 1.5).

3.5 GENERATE MAINTENANCE INFORMATION

There are several jobs on SYSDECK to produce maintenance information, such as listings of the system and various cross references of overlays, macros, and symbols (part III, section 1.6).

3.6 TABLE OF INSTALLATION DEPENDENCIES

The following table lists the elements, other than COMPASS and UPDATE, that are necessary to build the product (column 1) and also those elements necessary to use the product once it is built (column 2). The order of installation jobs is discussed in detail in part III, section 1.

In the following table, DBUGLIB is the object time library used by the TRAP utility; COBLIB is the object time library used by COBOL; FORTRAN and RUNLIB are the object libraries used by FTN and RUN respectively. SRTMACS is an XTEXT file containing COMPASS Sort macros.

Product	Required Elements To:	
	Build (xxxGEN or xxxMOD)	USE (User Execution)
SCOPE 2.0		
ANALYZE	FTN Compiler FORTRAN Library	
LIBEDT	FTN Compiler FORTRAN Library	
TRAP		DBUGLIB
COMPASS 2.0		
SORTMRG 1.0		SRTMACS SRTLIB Library
COBOL 1.0	SRTLIB Library	SRTLIB Library COBLIB Library
FORTRAN Extended 2.0		FORTRAN Library
FORTRAN Run 2.0		RUNLIB Library
FORTRAN Library	FTN Compiler	
RUNLIB		

Figure I-3.1. Table of Installation Dependencies

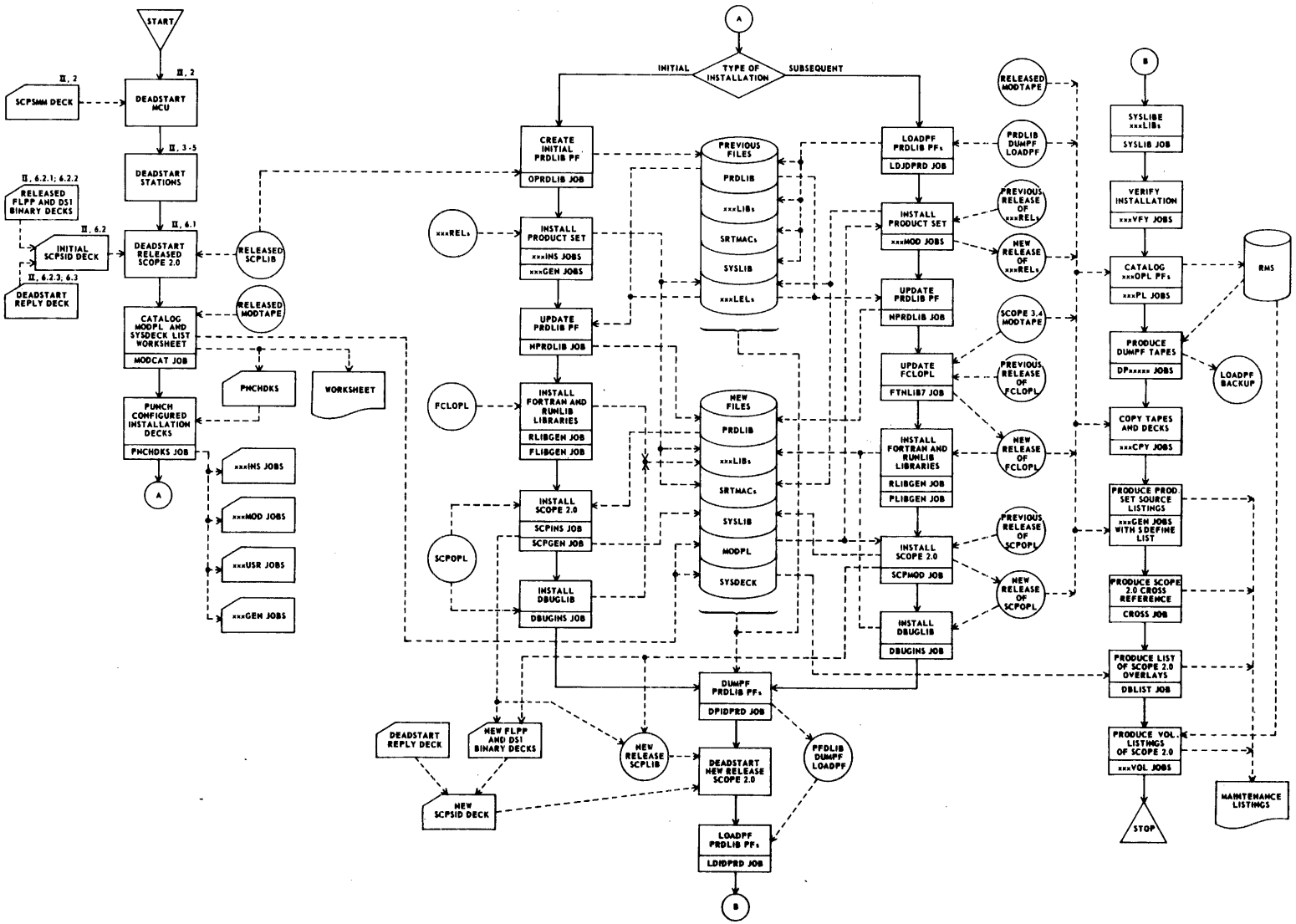


Figure I-3.2. Flowchart of Installation

ADDITIONAL MANUALS 4

Refer to the following publications for hardware, operating system, and product set information. Those with an asterisk (*) are combined SCOPE 2.0 and SCOPE 3.4 manuals.

Control Data Publications	Pub. No.
SCOPE 2.0 Manuals	
SCOPE 2.0 Reference Manual	60342600
SCOPE 2.0 User's Guide	60372600
*Loader Reference Manual	60344200
*Record Manager Reference Manual	60307300
SCOPE 2.0 Diagnostic Handbook	60344100
On-line Diagnostics Reference Manual	60405900
Hardware Manual	
CONTROL DATA® CYBER 70/Model 76 Reference Manual	60367200
Station Manuals	
7611-1 I/O Station Version 2 Operator's Reference Manual	60343600
7611-2 Magnetic Tape Station Version 2 Operator's Reference Manual	60343700
6000 and CYBER 70 Series SCOPE 3.4 Station Version 2 Operator's/ Reference Manual	60343800
6000 SCOPE 3.4 Installation Handbook	60307400
SCOPE 2.0 Support Manuals	
System Maintenance Monitor Version 3 Reference Manual	60312400
System Maintenance Monitor Version 3 Operator's Guide	60373800
Product Set Manuals	
*COBOL Reference Manual	60384200
*COMPASS Reference Manual	60279900
FORTRAN Run Reference Manual Version 2	60360700
*FORTRAN Extended Reference Manual	60305600
*Sort/Merge Reference Manual	60343900
*UPDATE Reference Manual	60342500

SCOPE 2.0 MAINTENANCE PACKAGE 1

Since the SCOPE 2.0 maintenance package is only used to modify the system and not to deadstart, it is described only in part III, section 1.

2.1 REQUIREMENTS

The SMMB initialization card deck is released in absolute (ABS) format. The first card is a 77 (prefix) card. The last card is a lace card that contains binary data.

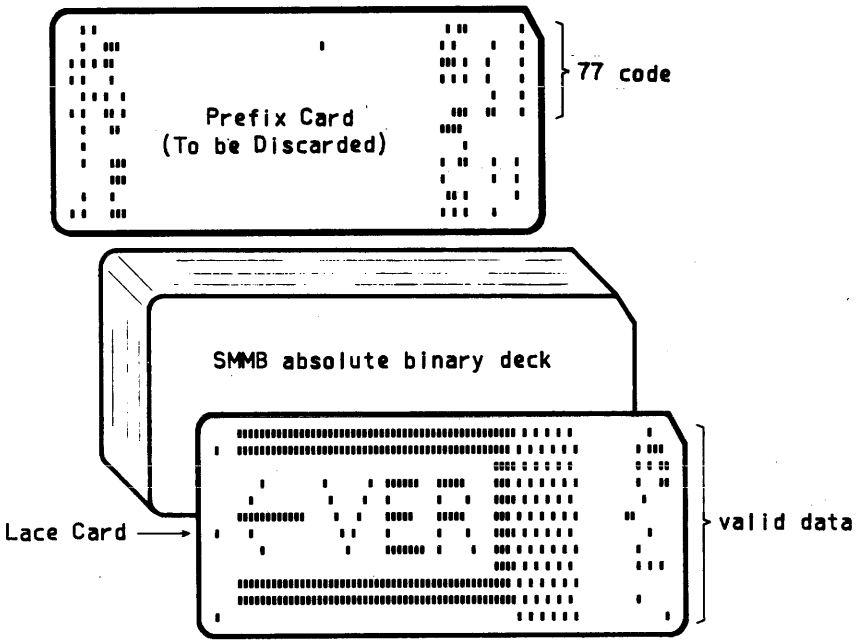


Figure II-2.1. SMMB Deck as Viewed from Back to Show Card Formats

It is the customer engineer's responsibility to install and maintain SMM 3.0. SMMB installation information is included in this manual because SMM is a requirement to deadstart and run SCOPE 2.0. Refer to the System Maintenance Monitor Version 3.0 Operator's Guide for SMM operating procedures and interface specifications.

2.2 DEADSTART PROCEDURE

Perform the following operations to deadstart the MCU and load the MCU operating system monitor program from the card reader.

1. Load the SMMB card deck as formatted in section 2.1 into the MCU card reader.
2. Press the deadstart button (ON-LINE/OFF-LINE button) on the card reader.
3. Press MOTOR POWER switch on the MCU card reader.

4. Press AUTO/MAN switch so that MAN lights.

5. Press the READY switch on the card reader operator panel.

The MCU reads in the card deck and initiates execution of the MCU monitor program in the MCU.

A successful deadstart of the MCU is indicated when the following information appears on the console: COPYRIGHT CONTROL DATA CORPORATION 1971 on the right screen and SMMB3000* on the lower left screen.

NOTE

If SMMB is restarted after SCOPE 2.0 is installed, it is necessary to reestablish communication with the CPU by typing in or reading in by card:

CN 10730,10460,5344,10540.

Part IV section 3.2.3 describes each of these parameters. Refer to the SMM Operator's Guide for a detailed description of the CN command and the SMM MCU/CPU communication block.

7611–1 I/O STATION 3

3.1 REQUIREMENTS

SMMB must be installed, as in part II, section 2. The installation of the 7611-1 I/O Station involves:

1. Transferring the 7611-1 I/O Station system from the STALIB tape to the station disk pack (section 3.2.1)

2. Deadstarting the 7611-1 I/O Station from disk (section 3.2.2)

The following materials are needed to perform this installation.

STATDS	Card to transfer the 7611-1 I/O Station system from tape to disk
STADDS	Card to initialize the 7611-1 I/O Station from disk
STALIB	Tape containing the 7611-1 I/O Station library
STACLR	Card to clear memory and channel

Additional instructions included in this section are:

Procedures to read the SCOPE library (SCPLIB) tape from the 7611-1 I/O Station (section 3.2.3)

Procedures to log in the station (section 3.2.4)

3.2 PROCEDURES

3.2.1 TAPE DEADSTART

1. Turn on all I/O station peripheral equipment. For turn-on procedures, refer to the reference/CE manual for each device. (It isn't necessary to turn on the punch to deadstart the station.)

2. Place the 7611-1 disk pack to be used for the new system on the I/O station disk drive; ready the unit.

3. Mount the 7611-1 I/O Station library tape (STALIB) on tape unit 0; ready the unit at 800 bpi.

4. Place STATDS (tape deadstart card) in the 7611-1 I/O Station card reader.

5. Press MOTOR POWER switch on the card reader operator panel.

6. Press deadstart button (ON-LINE/OFF-LINE button) on the inside panel of the card reader.

7. Press AUTO/MAN swtich so that MAN lights.

8. Press READY switch on the operator panel.

9. After the card is read†, press the END OF FILE switch twice.

PPUs 0, 3, 4, and 5 are deadstarted. (The disk pack is initialized which requires about three minutes. To avoid addressing the pack, type TERM 4 and press CR.) PPU 5 reads the tape and displays every program name. The system builds the station disk pack from the binaries on the STALIB tape.

The initialization operation is complete when the system rewinds the tape and turns off the selection unit light. Deadstart the disk using the procedure in section 3.2.2.

3.2.2 DISK DEADSTART

1. Turn on all I/O station peripheral equipment. For turn on procedures, refer to the reference/CE manual for that device.

2. Place the disk pack (created in section 3.2.1) on the I/O station disk drive; ready the unit.

3. Place STADDS (disk deadstart card) in the card reader.

4. Press MOTOR POWER switch on the card reader operator panel.

5. Press the deadstart button (ON-LINE/OFF-LINE button) on the inside panel of the card reader.

6. Press the AUTO/MAN switch so that MAN lights.

7. Press the READY switch to read the card.

8. After the STADDS card is read†, press END OF FILE switch twice; the system deadstarts PPUs 0, 3, and 4 and brings up the INITL. display.

9. Deadstart the remaining PPUs using the AUTO command.

   Type AUTO.
   Press CR

   The deadstart operation is complete when the SFNT display appears on the console.

10. Turn off any printer that isn't useable.

3.2.3 DEADSTART SCOPE 2.0 FROM THE 7611-1 I/O STATION

To deadstart the SCOPE 2.0 system from the 7611-1 station, follow the procedures in part II, section 6.1, noting these exceptions:

1. Specify the 7611-1 I/O Station as the system source device (part II, section 6.3.4).

---

† If the STATDS or STADDS card does not perform its function, it may be necessary to clear memory and channel. Read in the STACLR card using the same procedure as to read in the STATDS or STADDS cards.

2. The following messages appear in the PPU5 message buffer area on the station console.

   a. DEADSTART 7000 SCOPE 2.0 indicates that the SCPLIB tape is being read.

   b. DEADSTART COMPLETE WAITING FOR LOGIN appears while the tape is rewinding and until the deadstart process is complete.

   c. CPU REQUESTS LOGIN appears when the deadstart process is complete.

3.2.4 LOG IN

When SCOPE 2.0 is installed, log the station in with the following procedure.

1. Press the / key to select CENTRAL mode; **CENTRAL** is displayed at the bottom of the display area.

2. To allow communication between the I/O station and central:

   Type LOGIN,ggg.

ggg	Optional three-alphanumeric-character station identifier; if omitted, the default station parameter IOS is used.

   Press CR

Refer to part II, section 6.1 for the procedures to ready the station for communication with central, to designate a SYSTEM OPERATOR station, and to assign the number of jobs to multiprogram in central.

7611–2 MAGNETIC TAPE STATION 4

4.1 REQUIREMENTS

The procedures in section 4.2 describe the installation of the 7611-2 Magnetic Tape Station.

SMMB must be installed as in part II, section 2.

Materials to install the station are:

MTSTDS	Deck to initialize the station disk pack from tape; it is also on the SCPLIB tape as TDS
MTSLIB	Tape containing the station system library
MTSDDS	Card deck that initializes the station from disk; it is also on the SCPLIB tape as FDS to initialize the station at SCOPE 2.0 installation time.

Refer to the 7611-2 Magnetic Tape Station Version 2 Operator's Reference Manual for operating procedures.

4.2 PROCEDURES

Use the procedure in section 4.2.1, tape deadstart, for the initial deadstart of the station. These procedures load the station system from the MTSLIB tape to a disk pack and then load the system from the disk pack into the 7611-2.

The disk pack does not need to be loaded with the station system each time the station is deadstarted. After the system has been placed on the 7611-2 disk pack, use the procedure in section 4.2.2 to deadstart the station from disk either at SCOPE 2.0 installation time or after SCOPE 2.0 is already installed. However, if the system has been lost (because of disk errors, for example), the disk pack must be reloaded.

4.2.1 TAPE DEADSTART

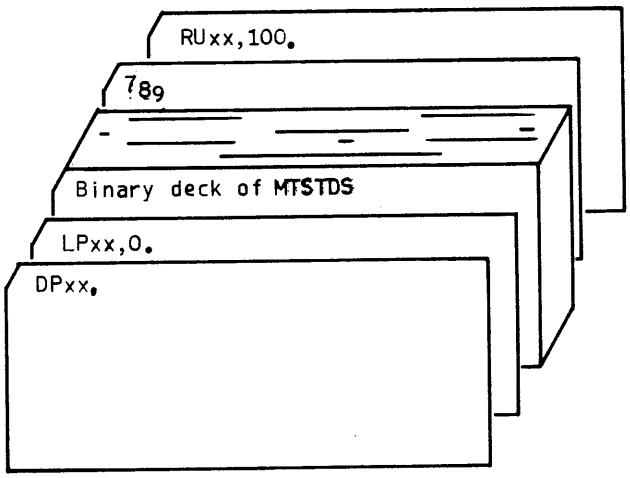
1. Turn on the peripheral equipment. For turn-on procedures, refer to the reference/CE manual for the device.

2. Master clear the CC522 station console (maintenance panel) and select the left display page; turn up intensity on display.

3. Place a disk pack on the 857 disk drive.

4. Mount the MTSLIB tape on unit 0 at 800 bpi; ready the unit.

5. Load the MTSTDS deck into the MCU card reader; the format of the deck is as follows.



xx.	FLPP connected to the 7611-2 Magnetic Tape Station
DPxx.	Deadstarts the designated FLPP
LPxx,0.	Loads the binary cards in MTSTDS in the FLPP specified starting at location 0
RUxx,100.	Executes the program in the designated FLPP starting at address 100

6. Press the AUTO/MAN switch so that AUTO lights.

7. Press the MOTOR POWER switch so that the motor runs.

8. Press READY switch (operator panel).

   The system reads the MTSTDS deck and loads the station system on the disk pack.

   The following message appears on the station display while the disk is addressed.

   INITIALIZING DISK

   Then

   CREATING LIBRARY

   appears; the tape is read and written in a scratch portion of the station disk.

   The following message appears as the routines are placed in the proper portion on the disk (xxx = routine name).

   CREATING LIBRARY xxx

9. The installation of the station is completed during the installation of SCOPE 2.0.

   When installing SCOPE 2.0, insert a CHQ card describing the station in the deadstart reply deck (section 6.3.3).

When the deadstart deck SCPSID is read, the station receives a bootstrap called FDS that reads the library from its disk pack. FDS deadstarts the multiplexer PPU (XPP) in the station and sends it a deadstart program. The XPP deadstarts the remaining station PPUs. SCOPE 2.0 must be installed. The station is now capable of performing blank labeling functions and diagnostic operations.

4.2.2 DISK DEADSTART

If SCOPE 2.0 has been installed and the station system is on a disk pack, use the following procedure to deadstart the station (unless the system library is lost because of disk errors).

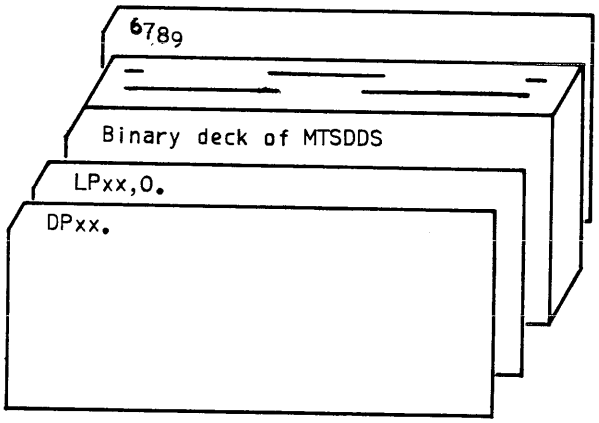
If SCOPE 2.0 and the station are to be installed at the same time, use only steps 1 through 3 and then proceed with the SCOPE 2.0 installation procedures (part II, section 6.1).

1. Turn on the peripheral equipment. For turn-on procedures, refer to the reference/CE manual for the device.

2. Master clear the CC522 station console (maintenance panel) and select the left display page; turn up the density on the display.

3. Place the disk pack containing the station system on the 857 disk drive.

4. Load the MTSDDS deck into the MCU card reader in the following format.



xx.	FLPP connected to the station
DPxx.	Deadstarts the designated FLPP
LPxx,0.	Loads the binary cards in MTSDDS into the FLPP specified starting at location 0

5. Press the AUTO/MAN switch so that AUTO lights.

6. Press the MOTOR POWER switch so that the motor runs.

7. Press the READY switch on the operator panel.

   The system reads the FDS program contained in MTSDDS and reads the library from its disk pack. FDS deadstarts the multiplexer PPU (XPP) in the station and sends it a deadstart program. The XPP deadstarts the rest of the station PPUs. The station is now capable of performing blank labeling functions and diagnostic operations.

4.2.3 DEADSTART SCOPE 2.0 FROM 7611-2 MAGNETIC TAPE STATION

To deadstart the SCOPE 2.0 system from the 7611-2 magnetic tape station, follow the procedures in part II, section 6.1 noting the following:

1. Specify the 7611-2 Magnetic Tape Station as the system source device (part II, section 6.3.4).

2. n. LOG IN THE STATION appears on the station display console when the deadstart process is complete.

4.2.4 LOG IN

1. When SCOPE 2.0 is ready to communicate with the station, it sends an initiate message and the station displays the following message on the console display.

   n.LOG IN THE STATION

2. Type n.LOGIN ggg
   Press CR

n	Same number as in preceding message
ggg	Optional three-alphanumeric-character station identifier; if omitted, the default station identifier MTS is used.

   The station is operational when the data appears in the upper right-hand corner of the unit status display.

6000 OR CYBER 70 SERIES SCOPE 3.4 STATION 5

5.1 REQUIREMENTS

Refer to the SCOPE 3.4 Installation Handbook for installation of the 6000 SCOPE 3.4 Station and to the 6000 and CYBER 70 Series SCOPE 3.4 Station Version 2 Operator's Reference Manual for operating procedures.

5.2 PROCEDURES

5.2.1 INSTALLATION AND LOG IN

1. Install the 6000 SCOPE 3.4 operating system using the installation instructions in the 6000 SCOPE 3.4 Installation Handbook. To indicate that the 6000 computer system is to be used as a station, insert *DEFINE,STATION in the UPDATE record of the SCOPE1 deck. (Remove the card from the deck when a configured program library is created.)

2. To bring the station to control point after SCOPE 3.4 is deadstarted and is executing:

   Type n.STATggg
   Press CR

n	Control point number assigned to 6000 station
ggg	Optional three-alphanumeric-character station identifier; first character must be alphabetic. If parameter is omitted, the default station identifier CCP is used.

3. If SCOPE 2.0 is already deadstarted and executing, log in occurs automatically; if the SCOPE 2.0 deadstart tape SCPLIB is to be read from the 6000 station, log in of the 6000 station occurs when SCOPE 2.0 deadstart initialization or recovery is complete; if SCOPE 2.0 is not deadstarted and executing, the following message appears at the 6000 station control point.

   WAITING FOR 7000

5.2.2 DEADSTART SCOPE 2.0 FROM 6000 3.4 STATION

To read the SCPLIB file from the 6000 station, use the following procedure.

1. Complete the preceding LOG IN procedure (section 5.2.1).

2. Use the SCOPE 2.0 installation procedure (part II, section 6.1) assigning the 6000 station channel as the channel through which the system library is to be sent (source command in part II, section 6.3.4).

3. After receiving the deadstart signal from the 7000, the 6000 station searches for a permanent file named DEADST (a default file name set by an assembly option) to transfer it to the SCOPE 2.0 system. If that permanent file is found, the SCPLIB file is copied from the 6000 station to the SCOPE 2.0 system disk. (Procedures to catalog SCPLIB on permanent file DEADST are in the Operator's Reference Manual for the SCOPE 3.4 Station.)

4. If the default file DEADST cannot be found as a permanent file, the following message appears on the SCOPE 3.4 system B display.

   REQUEST DEADSTART FILE, LAST ASSIGN NOT FOUND

5. If the deadstart file is on tape:

   Type n.ASSIGN uu.
   Press CR

n	Control point number assigned to the 6000 Station
uu	6000 equipment status table (EST) ordinal of the tape unit uu

   Type DSFILE,uu,dd.†
   Press CR

uu	Deadstart file is on tape at EST ordinal uu
dd	LO, HI, HY for 7-track tape density; if omitted, default is HI for 7-track. This parameter is ignored for 9-track tapes.

6. If the deadstart file is another permanent file (other than DEADST):

   Type DSFILE,pfn.†
   Press CR

pfn	Deadstart file is a permanent file with file name pfn.

7. The SCPLIB is copied from the 6000 station to the SCOPE 2.0 system disk. When SCOPE 2.0 is deadstarted, it sends a station initiate message to all stations known in the system.

   Continue with SCOPE 2.0 installation in part II, section 6.1.

†The DSFILE command can be entered after the n.STATggg (section 5.2.1, step 2) but before the SCOPE 2.0 deadstart procedure (section 5.2.2). In this case the REQUEST DEADSTART FILE message does not occur.

Deadstart is the process that prepares a computer system for running jobs. With SCOPE 2.0 there are two kinds of deadstart.

Deadstart initialization	The DS1 module is loaded into SCM from the MCU card reader; the system library to be initialized is either on a tape or is one of five cycles of the system library permanent file resident on the system disk. Procedure is in section 6.1.
Deadstart recovery	The DS1 module is loaded into SCM from the system disk; the system library to be recovered is either on a tape or is one of five cycles of the system library permanent file resident on the system disk. Procedure is in section 6.4.

The functions of the deadstart program during the deadstart process are described in part IV, section 3.6.7. The following figure identifies the information that can be recovered from the previous deadstart.

Deadstart Option	Deadstart Initialization		Deadstart Recovery	
	System Library from Tape	System Library from Disk	System Library from Tape	System Library from Disk
EST, LIB, SYS, FLS, FLL, and CHQ recovered?	no	no	yes	yes
Permanent files recovered?	optional	automatic	optional	automatic
Flaw table (in volume label group) recovered?	automatic if permanent file recovery selected	automatic	automatic	automatic
T.MAXS, T.MAXL, and T.MAXBUF recovered?	no	no	yes	yes
SFT I/O files recovered?	no	no	no	optional
SIF LCM buffers recovered?	no	no	no	automatic if SFT I/O files recovered

Figure II-6.1. Diagram of Deadstart Options

6.1 DEADSTART INITIALIZATION PROCEDURE

This section contains a brief description of the deadstart initialization procedure with references to other sections for detailed information.

NOTE

Read all information contained in part II, sections 6.1 through 6.4 before attempting to use this procedure. It is necessary to know the hardware assignments (such as channel and FLPP assignments) before beginning the installation.

1. Construct the SCPSID deck to reflect the site's configuration. — part II, section 6.2

2. Deadstart the MCU. — part II, section 2

3. Deadstart the stations that are to be used in the deadstart process. — part II, sections 3.2, 4.2, and 5.2

4. If the system library is on tape, mount the SCPLIB tape on either a 7611-1 station, 7611-2 station, 6000 station, or on-line tape unit; ready the unit. The default values for the 7611-1 and 7611-2 station and on-line tapes are: unit 0, 556 bpi, 7-track tape. A 6000 station tape unit must be assigned to the station control point. — part II, sections 3.2.3, 4.2.3, or 5.2.2

5. Place the SCPSID deck in the MCU card reader.

   a. Press the AUTO/MAN switch so that AUTO lights.

   b. Press the MOTOR POWER switch so that the motor runs.

   c. Press the READY switch on the operator panel.

   The cards are read; if the card reader does not begin to read cards, begin again with step 2 or refer to the EP command in part IV, section 3.2.4.

6. Reply to the deadstart requests at the MCU display console. (The sequence of requests and replies are in the flowchart in section 6.5.1). — part II, section 6.3

7. Log in the stations. — part II, sections 3.2.4, 4.2.4, and 5.2.1

8. Ready the 7611-1 and 6000 stations for communication with central.

   Type ONSTAT.
   Press CR

9. Designate one SYSTEM OPERATOR station.

   Type SETOP.
   Press CR

   Type ONOP.
   Press CR

10. At the SYSTEM OPERATOR station, assign the number of jobs to be executed in multiprogramming mode.

    Type JCB,n
    Press CR

11. If a permanent file dump tape of the system libraries exists, run the LDIDPRD job. — part III, sections 1.2.2 and 1.2.4

12. Run the SYSLIB job to specify the object libraries to the operating system. — part IV, section 3.6.3

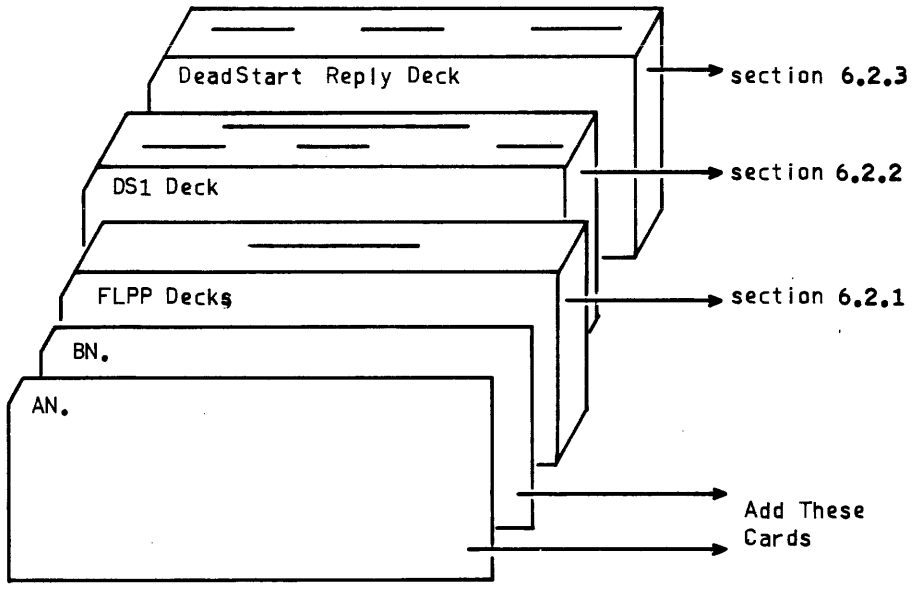
6.2 FORMAT OF SCPSID DEADSTART DECK

The purpose of this section is to describe the structure of the deadstart deck. (Its function is described in part IV, section 3.6.7.) The deck contents and the SMM commands necessary in the deadstart deck are described in sections 6.2.1 through 6.2.4. Refer to the System Maintenance Monitor Version 3 Reference Manual for a more detailed description of each SMM command.

NOTE

Keypunch errors in the deadstart deck may cause deadstart to abort.

SCPSID contains the following decks and control cards.

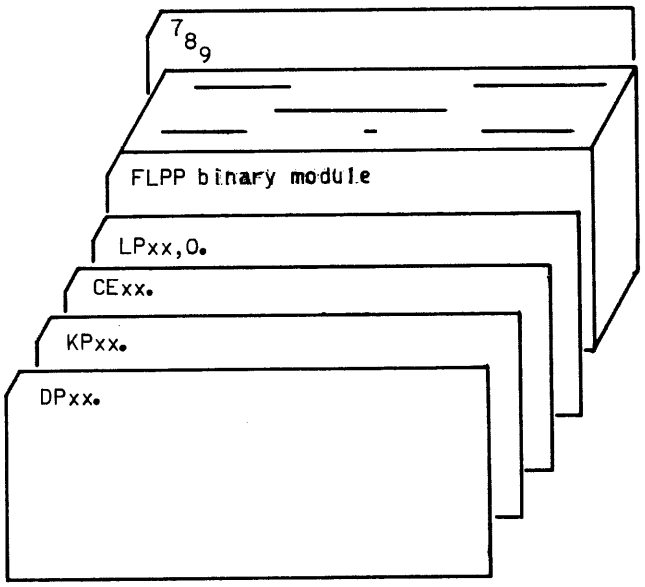


6.2.1 FLPP DECKS

Place a FLPP card deck in the SCPSID deck to initialize each FLPP in the site's hardware configuration.

Deck Format for FLPPs Used to Deadstart SCOPE 2.0

Construct a deck similar to the following for each FLPP to be used for the deadstart process.



DPxx.	Loads resident program and deadstarts the designated PPU
KPxx.	Clears the entire FLPP memory, except for resident
CExx.	Clears parity errors for designated FLPP
LPxx,0.	Loads binary program to the FLPP specified starting at location 0; all load addresses given to SMM for all FLPPs are 0.
FLPP binary module	FLPP binary programs used to deadstart SCOPE 2.0; deck names are:

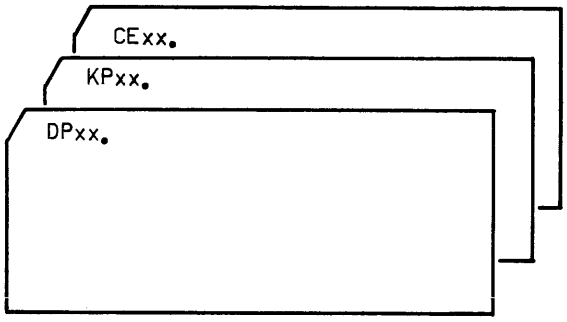
CPL	6000 station deck
DSC	7611-2 Magnetic Tape Station deck
DSK	System mass storage device decks
MMD	On-line tape unit device (MTD is COMPASS ident)

The 7611-1 I/O station does not have an FLPP.

$MPxx_1,xx_2$.	If there is more than one FLPP driver for a given device in the system, an FLPP deck is necessary for each driver. Since the binary deck for each is identical, the MP (move) command can be used to move the contents of $FLPPxx_1$ to $FLPPxx_2$.

Deck Format for Other FLPPs

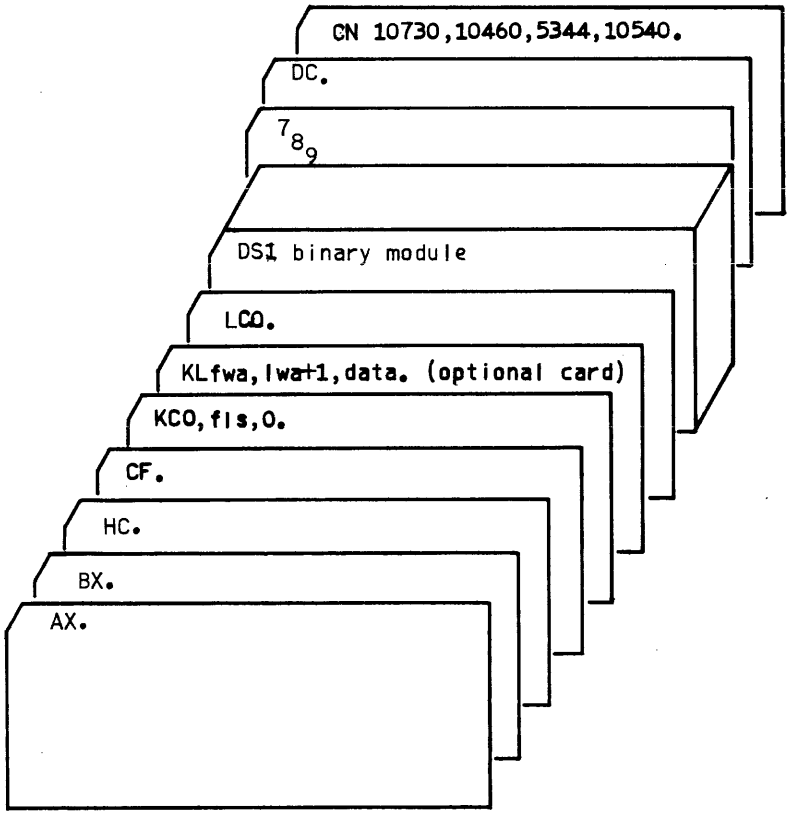
Construct a deck in the following format for each FLPP not part of the deadstart process.



DPxx.	Loads resident program and deadstarts the designated PPU
KPxx.	Clears the entire PPU memory, except for resident
CExx.	Clears parity errors for designated PPU

6.2.2 DS1 DECK

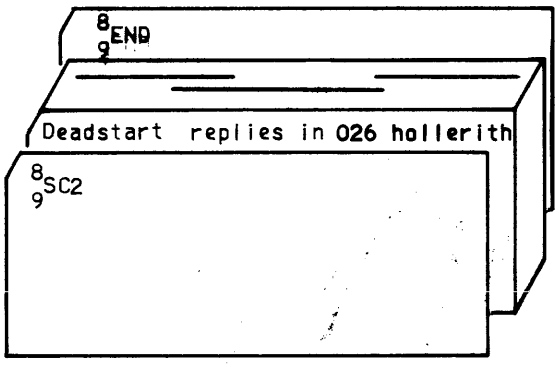
The DS1 deck must consist of the following cards.



AX.	Commands SMM to bring up an X format display of SCM on the A screen
BX.	Commands SMM to bring up an X format display of SCM on the B screen
	Forcing of an SCM display during the initial (noninterruptable) phase of DS1 prevents SMM from displaying LCM. Avoid displaying LCM while the CPU is executing in noninterruptable mode, because the MCU stops communicating with the CPU after a period of time.
HC.	Applies the deadstart signal to the CPU until an RD or DC command is given or until a reference to LCM is required
CE.	Clears parity error indicators for SCM and LCM
	The CF. and KC0,fls,0. commands are needed after running off-line diagnostics in the CPU to prevent the DS1 binary deck from writing over the MCU/CPU communication area in SCM.
	The CF. command directs the MCU to stop monitoring the MCU/CPU communication area and to stop writing parity status into the communication area.
KC0,fls,0.	Sets SCM from 0 to fls-1 to zeros. fls is length of SCM: 100000_8 for half-sized SCM or 200000_8 for full-sized SCM.
KL fwa,lwa+1, data.	Sets LCM from fwa to lwa to value of data; optional
LC0.	Loads binary module DS1 to SCM starting at location 0
DS1 binary module	The module must end with a 7/8/9 card
DC.	Deadstarts the CPU; the CPU exchange jumps to location 0
CN 10730, 10460, 5344, 10540.	Directs the MCU to monitor the MCU/CPU call block for CPU calls and to store FLPP and SCM/LCM status in the communication area. The parameters specify the locations of the communication area, the EEA exchange package, and addresses required by the SMM SY command. Whenever ORE or ORL changes are made, check the CN parameters for accuracy. Refer to part IV, section 3.2.3 for a detailed description of the CN parameters.

6.2.3 DEADSTART REPLY DECK

The deadstart reply deck is constructed in the following way.



8/9 SC2	The 8/9 punch in column 1 indicates to SMM that this is a reserve control card. SC2 is an identifier to DS1 to signal the beginning of the SCOPE 2.0 deadstart reply deck. This card is necessary even if there are no requests entered by card (when all are entered through the MCU console).
Deadstart Replies	Optional; if replies are to be included in the SCPSID deck: Replies must be in the same order as presented in section 6.3. BCD text on the card begins in column 1. Up to 40 hollerith characters are allowed per card. Terminate the text with an 11/12 punch to indicate end of reply. (This corresponds to the carriage return when submitting requests at the MCU console.) Comments may be added after the 11/12 punch.
8/9 END	The 8/9 punch indicates to DS1 that this is a reserve control card. END indicates the last card in the deadstart reply deck. It is necessary even if no deadstart request replies are in the deadstart deck.

6.2.4 SAMPLE SCPSID DECK STRUCTURE

Following is the SCPSID deck as it would be constructed for the typical configuration described in sections 6.3.1 and 6.3.3. There are two examples of deadstart reply decks, the first for a deadstart initialization without permanent file recovery and the second for a deadstart recovery with permanent file recovery.

Only FLPP decks necessary to deadstart the system are required in the SCPSID. FLPPs are reloaded automatically by deadstart. In the following example all possible decks are included.

! indicates an 11/12 punch; * indicates optional cards.

FLPP and DS1 Decks

```
AN.
BN.
DP2.
KP2.
CE2.
LP2,0.
DSK  (Disk FLPP binary driver)
7/8/9
DP3.
KP3.
CE3.
LP3,0.                                  }
DSK  (Disk FLPP binary driver)          } or MP2,3.
7/8/9                                   }
DP4.
KP4.
CE4.
DP5.
KP5.
CE5.
DP6.
KP6.
CE6.
DP7.
KP7.
CE7.
DP10.
KP10.
CE10.
LP10,0.
CPL  (6000 FLPP binary driver)
7/8/9
DP11.
KP11.
CE11.
LP11,0.
MMD  (On-line tape FLPP binary driver)
7/8/9
DP12.
KP12.
CE12.
LP12,0.
DSC  (7611-2 FLPP binary driver)
7/8/9
DP14.
KP14.
CE14.
LP10,0.                                 }
CPL  (6000 FLPP binary driver)          }  or MP10,14.
7/8/9                                   }
DP15.
KP15.
CE15.
AX.
BX.
HC.
```

```
CE.
CF.
KC0, fls, 0
*KL.
LC0.
DS1 binary module
7/8/9
DC.
CN 10730, 10460, 5344, 10540.
```

Deadstart Reply Decks

The following example is a deadstart reply deck for deadstart initialization without permanent file recovery and the source of the system library on tape.

```
8/9SC2
FLS=200000,FLL=1764000!
LIB=205!
EST=1,DT=AF/PF/SY,CH=2/3,UN=0!
EST=2,DT=AF,CH=3/2,UN=1!
EST=3,DT=AF,CH=4/5,UN=0!
EST=4,DT=AF,CH=6/7,UN=1!
EST=40,DT=MT,CH=11,UN=0!
EST=41,DT=MT,CH=11,UN=1!
EST=42,DT=MT,CH=11,UN=2!
!              END OF EST ENTRIES
CHQ=2,DT=AF/2P,FC=01043276!
CHQ=3,DT=AF/2P,FC=01047632!
CHQ=4,DT=AF/4P,FC=01043200!
CHQ=5,DT=AF/4P,FC=01040032!
CHQ=6,DT=AF/4P,FC=01043200!
CHQ=7,DT=AF/4P,FC=01040032!
CHQ=10,DT=6ST,FC=01200000!
CHQ=11,DT=MT,FC=01003200!
CHQ=12,DT=MTS,FC=01320000!
CHQ=13,DT=6ST,FC=01200000,PP=14!
CHQ=16,DT=7ST,FC=0!
CHQ=17,DT=7ST,FC=0!
!              END OF CHQ ENTRIES
T12!           GIVE SYSTEM SOURCE REPLY
N!             NO PERMANENT FILE RECOVERY
20,5,1!        FLAW ON TRACK 20 OF C2, U0 (SYSTEM RESIDENT DEVICE)
!              END OF FLAWS ON C2, U0, TRACK 20
1200!          OTHER FLAWS ON C2, U0
35,0,12!       OTHER FLAWS ON C2, U0
37,1,1!        OTHER FLAWS ON C2, U0
!              END OF FLAWS ON C2,U0
AUTO!          GO REPLY
*STORE 340,L,T.MAXBUF!          Maximum number of buffers that can be allocated to a job.
*STORE,0,L,T.SPF!               SFT disk write is inhibited
*STORE 160000,L,T.MAXS!         Optional for full size machine; maximum user FLS
*STORE 1400000,L,T.MAXL!        Optional for full size machine; maximum user FLL
GCM!
!              NO FLAWS ON TRACK 20 of C3, U1
!              NO OTHER FLAWS ON C3, U1
!              NO FLAWS ON TRACK 20 OF C4, U0
!              NO OTHER FLAWS ON C4, U0
```

```
20,8,1!          FLAWS ON TRACK 20 OF C6, U1
!                NO OTHER FLAWS ON TRACK 20 OF C6, U1
145,12!          OTHER FLAWS ON C6, U1
!                END OF FLAWS ON C6, U1
8/9 END
```

The following example is a deadstart reply deck for a deadstart recovery with permanent file recovery. The system library is on tape. The next command to appear after using this deck is the request for the disk address of the volume label group.

```
8/9SC2
FLS=200000,FLL=1764000!
LIB=205!
EST=1,DT=AF/PF/SY,CH=2/3,UN=0!
EST=2,DT=AF,CH=3/2,UN=1!
EST=3,DT=AF,CH=4/5,UN=0!
EST=4,DT=AF,CH=6/7,UN=1!
EST=40,DT=MT,CH=11,UN=0!
EST=41,DT=MT,CH=11,UN=1!
EST=42,DT=MT,CH=11,UN=2!
!                END OF EST ENTRIES
CHQ=2,DT=AF/2P,FC=01043276!
CHQ=3,DT=AF/2P,FC=01047632!
CHQ=4,DT=AF/4P,FC=01043200!
CHQ=5,DT=AF/4P,FC=01040032!
CHQ=6,DT=AF/4P,FC=01043200!
CHQ=7,DT=AF/4P,FC=01040032!
CHQ=10,DT=6ST,FC=01200000!
CHQ=11,DT=MT,FC=01003200!
CHQ=12,DT=MTS,FC=01320000!
CHQ=13,DT=6ST,FC=01200000,PP=14!
CHQ=16,DT=7ST,FC=0!
CHQ=17,DT=7ST,FC=0!
!                CHQ CONFIRMATION
T12!             GIVE SYSTEM SOURCE REPLY
Y!               PERMANENT FILE RECOVERY
8/9 END
```

6.3 DEADSTART REQUESTS AND REPLIES FOR DEADSTART INITILIZATION

Information defining the site's hardware configuration can be partially submitted in the SCPSID deck. All information not provided in this deck is requested at deadstart time at the MCU console with the request messages noted in this section. The first request displayed at the MCU console when the SCPSID deck is read is the one that follows the last reply supplied in the deadstart deck.

The following possibilities are available to the person installing the system.

- Answering all requests at the MCU console by only submitting the 8/9SC2 and the 8/9END cards in the SCPSID deck to represent the deadstart reply deck.
- Submitting some replies in the SCPSID deck with the following restrictions.

  If permanent files are not to be recovered from the previous deadstart, replies in sections 6.3.1 through 6.3.11 (up to but not including the volume label group) can be entered by card.

If permanent files are to be recovered from the previous deadstart, only replies in sections 6.3.1 through 6.3.5 can be submitted by card.

If any reply is entered incorrectly, an ERROR message appears. Refer to part IV, section 2.6 for corrective action.

6.3.1 EQUIPMENT STATUS TABLE (EST)

The EST contains the on-line equipment settings. They are set as each reply is received through the card reader or the MCU console.

Display Format

```
NO.    TYPE    STATUS    CHANNELS    UNIT

LIBRARY BUFFERS                      220
SYSTEM TABLES                         40
FLS                                 100000
FLL                                0764000
```

NO.	EST entry ordinal
TYPE	Device type
STATUS	Not used
CHANNELS	FLPP primary and secondary channels
UNIT	Unit number
LIBRARY BUFFERS	Number of 1000_8 word library buffers; to modify released value of 220 refer to section 6.3.2
SYSTEM TABLES	Number of 1000_8 word system tables; to modify released value of 40 refer to section 6.3.2
FLS	Field length for SCM; to modify released value of 100000_8 refer to section 6.3.2
FLL	Field length for LCM; to modify released value of 0764000_8 refer to section 6.3.2

Reply Format

A single carriage return when information is typed in (or 11/12 punch in column 1 when the information is on card) confirms that the EST is accurate and/or that EST input is completed.

To add to or modify the EST, submit the following information by card in the deadstart reply deck or at the MCU console when the EST display appears. The on-line equipment is set in the EST as each reply is received and the display is updated.

During a deadstart initialization with permanent file recovery only changes to the magnetic tape entries (MT and NT) can be made. Do not make any additions, deletions, or modifications to the system mass storage entries. They must be the same as in the last successful deadstart or recovery.

```
EST=ord,DT=dt,CH=cc1/cc2,UN=unit
```

ord — EST entry ordinal; 1 through NE.EST-1 (NE.EST=100_8 in released system)

dt — Device type and device characteristics when necessary. AF indicates that the equipment is a mass storage disk; SY indicates that the system library is read onto that file during deadstart; PF indicates that the equipment is for permanent file directory (PFD) and permanent file catalog (PFC).

AF	Mass storage device
AF/PF	Mass storage device; available for PFD and PFC.
AF/PF/SY	Mass storage device; available for PFD, PFC, and system library file.
AF/SY	Mass storage device; system library file residence
MT	7-track magnetic tape
NT	9-track magnetic tape

cc_1/cc_2 — FLPP primary (cc_1) and secondary (cc_2) channels when necessary; 2 through 17_8

unit — Physical unit number; 0 through 77_8

To delete EST entries type or submit by card the following (ord is the ordinal of the equipment to be deleted):

```
EST=ord
```

Example:

The sample EST entries for the typical configuration in figure II-6.2 are:

```
EST=1,DT=AF/PF/SY,CH=2/3,UN=0!
EST=2,DT=AF,CH=3/2,UN=1!
EST=3,DT=AF,CH=4/5,UN=0!
EST=4,DT=AF,CH=6/7,UN=1!
EST=40,DT=MT,CH=11,UN=0!
EST=41,DT=MT,CH=11,UN=1!
EST=42,DT=MT,CH=11,UN=2!
```

6.3.2 MEMORY SIZE

As an option at anytime during the building of the EST, the number of library buffers, the number of system tables, and the size of SCM and/or LCM buffer areas can be allocated. When submitting this information as part of the SCPSID deck, these cards should precede the EST cards. This information is displayed at the bottom of the EST display as shown in part II, section 6.3.1.

Reply Format for Library Buffers

Library buffers are allocated in LCM and are partially filled during deadstart with system overlays and directories. During system running, additional library buffers are used when the SYSLIBE function adds an LCM resident library. If the number of library buffers allocated is overflowed during the deadstart operations, the fatal error message LIB BUFFERS EXCEEDED is displayed. In this case, deadstart again with a larger number of library buffers specified.

Type or submit by card in the deadstart reply deck the following information.

```
LIB=xxx
```

xxx 3-digit number of library buffers; each buffer is 1000_8 in length; default is in IP.LIB

Reply Format for System Table Definition

System tables are used by the system as a data area for FATs, PREs, and other temporary tables.

Type or submit by card in the deadstart reply deck the following information.

```
SYS=xxx
```

xxx 1 to 3 digit number of system tables; each table is 1000_8 words in length; default is in IP.SYS

Reply Format for FLS and FLL

Type or submit by card in the deadstart reply deck the following information.

```
FLS=xxxxxx,FLL=yyyyyyy
```

xxxxxx 1 to 6 digit field length for SCM; default is 100000B (IP.SCMSI in part III, section 6.2.12)

yyyyyyy 1 to 7 digit file length for LCM; default is 764000B (IP.LCMSI in part III, section 6.2.12)

6.3.3 CHANNEL QUEUE TABLE (CHQ)

The channel queue table contains the characteristics of each hardware I/O channel.

Display Format

```
CHANNELS
NO.    TYPE    SC    BL    QL    FC
00     REJ     00    000   000   00000000
01     REJ     00    000   000   00000000
 .      .       .     .     .      .
 .      .       .     .     .      .
 .      .       .     .     .      .
```

NO. CHQ entry ordinal number

TYPE Device type and characteristics; REJ refers to an unused channel

SC Maximum number of subchannels for a station channel; cannot be modified during deadstart

BL Hardware I/O buffer length; cannot be modified during deadstart

QL Maximum number of requests for a station channel; cannot be modified during deadstart

FC Channel function code for FLPP initialization

Reply Format

To add to or amend the CHQ, type at the MCU console or submit on cards in the deadstart reply deck the following information. As each reply is processed, the CHQ display is updated.

During a deadstart initialization with permanent file recovery, do not make any changes that would affect the EST and CHQ mass storage device assignments.

```
CHQ=ord,DT=dt,FC=channel,PP=flpp
```

ord — CHQ entry ordinal; must be the I/O multiplexer channel (MUX) number to be described (2 through 17_8)

dt — Device type and characteristics when necessary

AF/2P	Mass storage device, 2 PPU configuration
AF/4P	Mass storage device, 4 PPU configuration
MT	7-track magnetic tape
NT	9-track magnetic tape
MT/2P	7-track magnetic tape 2 PPU configuration
NT/2P	9-track magnetic tape 2 PPU configuration
6ST	6000 or CYBER 70 Station
7ST	7611-1 I/O Station
MTS	7611-2 Magnetic Tape Station

channel — Channel configuration for FLPP initialization. The first four characters are data in direct address 6; the last four characters are data in direct address 7.

For system mass storage file driver (2 PPU configuration)

DIRECT ADDRESS 6

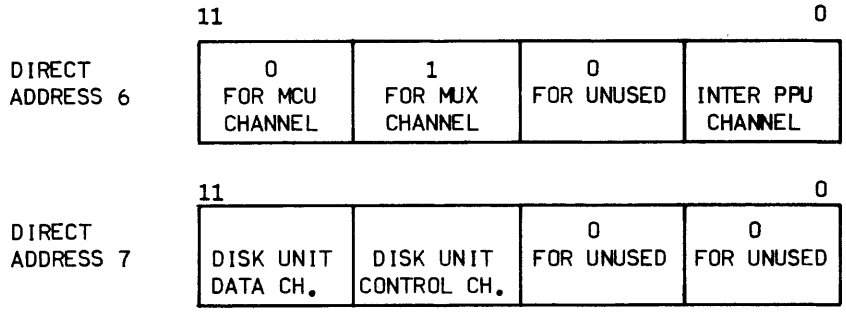
11			0
0 FOR MCU CHANNEL	1 FOR MUX CHANNEL	0 FOR UNUSED	INTER PPU CHANNEL

DIRECT ADDRESS 7

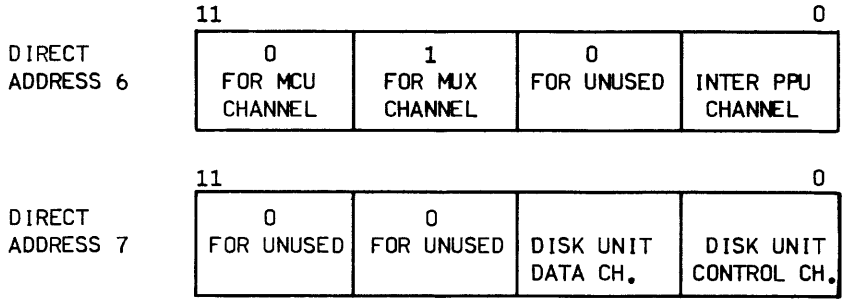
11			0
PRIMARY UNIT DATA CHANNEL	PRIMARY UNIT CONTROL CHANNEL	SECONDARY UNIT DATA CHANNEL	SECONDARY UNIT CONTROL CHANNEL

For system mass storage file driver (4 PPU configuration)

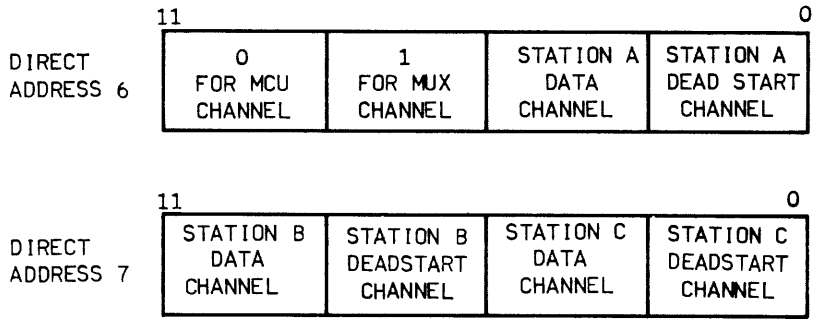
Lower numbered PPU



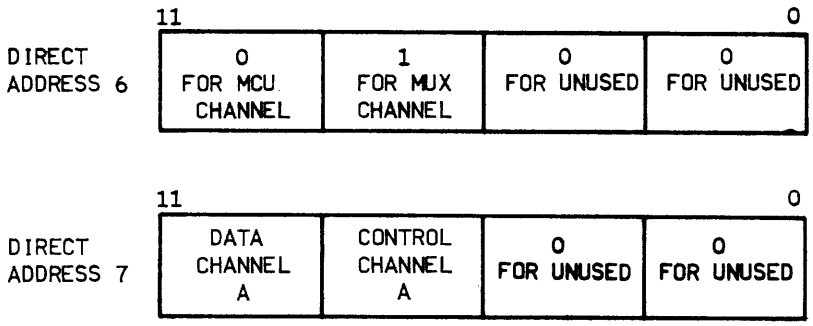
Higher numbered PPU



For station driver



For on-line tape driver



flpp Optional FLPP number; must be entered only when the FLPP number differs from the number of the CPU channel to which the FLPP is attached. For example, if CPU MUX channel 13 is connected to FLPP14, the parameters would be PP=14 on the CHQ=13,... reply.

REJ as type on the display indicates unused channel. Change a channel to unused channel by typing or submitting by card:

```
CHQ=ord
```

A single carriage return when information is typed in (or an 11/12 punch in column one when the information is on cards) indicates that the CHQ is confirmed and/or that CHQ input is completed.

Before confirming the CHQ, the operator may display the EST. To do so, type EST and press CR. Return to the CHQ display by typing CHQ and pressing CR.

Example:

The CHQ entries for the typical configuration in Figure II-6.2 are:

```
CHQ=2,DT=AF/2P,FC=01043276!
CHQ=3,DT=AF/2P,FC=01047632!
CHQ=4,DT=AF/4P,FC=01043200!
CHQ=5,DT=AF/4P,FC=01040032!
CHQ=6,DT=AF/4P,FC=01043200!
CHQ=7,DT=AF/4P,FC=01040032!
CHQ=10,DT=6ST,FC=01200000!
CHQ=11,DT=MT,FC=01003200!
CHQ=12,DT=MTS,FC=01320000!
CHQ=13,DT=6ST,FC=01200000,PP=14!
CHQ=16,DT=7ST,FC=0!
CHQ=17,DT=7ST,FC=0!
```

The EST entries for this typical configuration are in section 6.3.2.

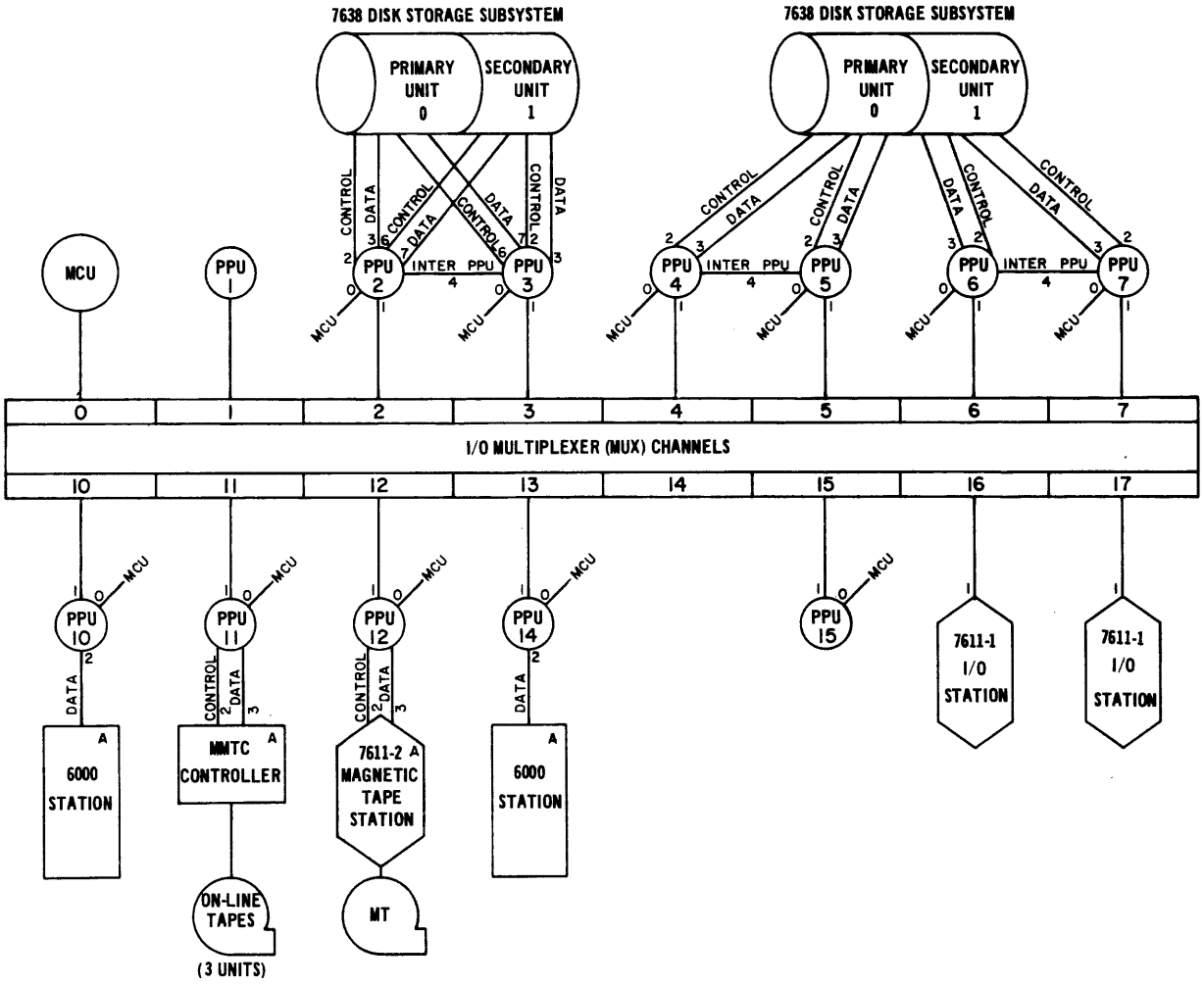


Figure II-6.2. Diagram of Typical Configuration

6.3.4 GIVE SYSTEM SOURCE

This message requests the CPU channel through which the system library will be sent.

Display Format

```
GIVE SYSTEM SOURCE
```

Reply Format

If a new system is to be loaded from an on-line or station tape, type at the MCU or submit by card in the deadstart reply deck the following information. Defaults do not pertain to 6000 station.

```
Tchannel,unit,density,type
```

channel — 2-digit CPU channel number through which the system library is to be sent. The channel must contain either an on-line tape driver or a station from which to read the tape (SCPLIB).

unit — Optional 1-digit physical unit number. If a 7611-1, 7611-2, or on-line tape unit is used to read the tape, the default is unit 0. If it is to be read from the 6000 station, the tape unit must be assigned to the station control point.

density — Optional 2-digit deadstart tape density: LO, HI, HY, or PE; HI (556 bpi) is default value when parameter is omitted.

type — Optional 1-digit tape type: 7 or 9; 7 is default value when parameter is omitted.

If the system library source is an existing cycle of the system file on disk, type at the MCU or submit by card in the deadstart reply deck the following:

```
D
```

6.3.5 PERMANENT FILE RECOVERY Y/N

This is displayed only when the system library source is tape; it is displayed in order to determine if the system should attempt the recovery of permanent files. Permanent file recovery is automatically attempted when the system library source is disk.

Display Format

```
PERMANENT FILE RECOVERY Y/N
```

Reply Format

To attempt permanent file recovery, type or submit by card:

```
Y
```

To specify that permanent files are not to be recovered, type or submit by card:

```
N
```

6.3.6 ENTER DISK ADDRESS

If permanent files are recovered, the following message is displayed requesting the track and sector address of the volume label group on the system mass storage device. Also, if permanent files are recovered, type all remaining replies, including this one.

Display Format

```
ENTER DISK ADDRESS
OF VOLUME LABEL GROUP
TTTTSS TTTT=TRACK  SS=SECTOR
```

Reply Format

```
track sector
```

The track and sector address is that of the volume label group as presented in the previous deadstart (section 6.3.13). This address is 2000 (track 20, sector 00) if no flaws exist on track 20.

6.3.7 ENTER FLAW

Requests for flaws on mass storage devices can occur at different times during deadstart, depending upon whether permanent files are recovered. Refer to the flowchart for the precise sequence of requests. The following description applies to flaw requests for both deadstart initialization and recovery.

If permanent files are recovered:

Flaws are requested on devices according to the following order.

System resident mass storage device

Mass storage devices containing permanent files in order of EST ordinals

All other mass storage devices in EST ordinal order

A permanent file recovery recovers the track flaw table in the volume label group; therefore, all flaws from the previous deadstart are recovered. The request for track 20 flaws does not appear.

1. The first request occurs before loading the system library. It requests any flaws not specified during the previous deadstart on the mass storage device that contains the system library.
2. The second request occurs after loading the system library. It requests any flaws not specified during the previous deadstart on any other mass storage devices that contain permanent files.
3. The third request displayed is for flaws not specified on other mass storage devices in the previous deadstart.

If permanent files are not recovered:

1. The first request occurs before loading the system library. It requests all flaws that exist on track 20 of the system resident mass storage device. (With deadstart recovery, track 20 flaws are not requested.) When this reply is entered, then another request is displayed for all other flaws on the system resident device.

2. After the system library is loaded, flaw requests are displayed for all other mass storage devices in the order in which they appear in the EST. First the request for flaws on track 20 of a device is displayed (only if deadstart initialization), then the request for flaws on all other tracks on that device.

As each reply is received, the device allocation map for each unit is updated until a carriage return (or an 11/12 punch in column 1 of a card) is received. If entering flaws through the console, enter a carriage return after the last track 20 flaw and again after the last flaw on the remaining tracks.

A disk I/O error message (described in part IV, section 2.6.1) that may occur is:

DISK I/O ERROR
Cchannel Uunit TRACKtrack SECTORsector

Display Format

```
ENTER FLAW CchannelUunit TT,SS,NN (TRACK 20)
```

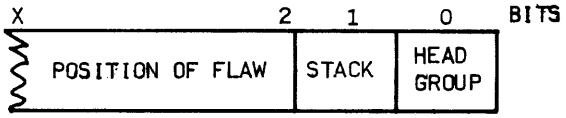
channel Channel number; 1 through 17_8

unit Unit number; 0 or 1

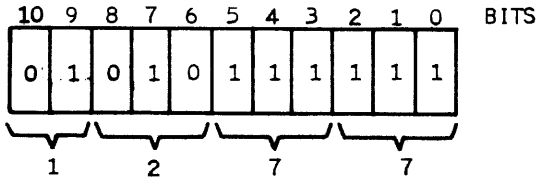
Reply Format

```
track,sector,no.
```

track 1 to 2 digit track number of flaw (20 in first reply); calculate the track number in the following way:



For example: position 257_8 of stack 1 in head group 1 is equivalent to track number 1277_8.



sector 1 to 2 digit initial sector number (0 to 47_8) of flaw; if omitted, sector 0 is assumed

no. 1 to 2 digit total number of consecutive sectors (1 to 50_8) that are flawed; if omitted, the remainder of track is assumed to be flawed.

Examples:

20,10,1 Only sector 10 is flawed on track 20

20,10 All sectors 10 and above are flawed on track 20

20 All of track 20 is flawed

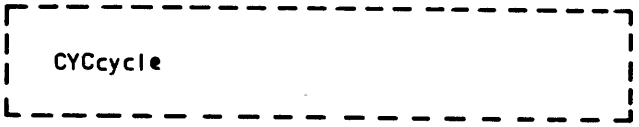
6.3.8 SYSTEM PFN AND CYCLE REQUEST

If permanent files are to be recovered, the following message is displayed. If permanent files are not recovered, the first system deadstarted is entered into cycle 1 of the permanent file named 7000 SCOPE VERSION 2.0.

Display Format

Reply Format

To specify a cycle of the current pfn, type:



cycle One of the five cycles of the current pfn that is to be recovered or entered into. The cycle number must be a value from 1 to 5 of the permanent file 7000 SCOPE VERSION 2.0. Deadstart from a user cataloged file is not allowed.

If the system library is from tape, the deadstart program catalogues the system library under the specified cycle. If the system library is from disk, the deadstart program recovers the specified cycle of the system.

To recover the current cycle of the system pfn, press CR. If the system library is from disk, the deadstart program uses the same cycle of the system permanent file as in the previous deadstart. If the system library is from tape, the deadstart program replaces the existing system library with one from tape so it has the same cycle of the system permanent files as in the previous deadstart.

6.3.9 ENTER FLAW (FOR PERMANENT FILE DEVICES)

Flaw requests at this point in deadstart initialization are displayed if permanent files are recovered to determine if there are additional flaws not specified in the previous deadstart on mass storage devices containing permanent files.

Refer to section 6.3.7 and the flowchart in section 6.5 for a detailed explanation.

6.3.10 SYSTEM PFN and CYCLE (MESSAGE)

If permanent files are recovered, the following informative message is displayed to indicate the cycle recovered.

Display Format

```
SYSTEM PFN AND CYCLE
CYCLE cycle
pfn
```

Reply Format

Press CR to continue deadstart initialization.

6.3.11 GO

The GO request is displayed twice. The first request requires a response of AUTO which is the command module to be processed by DS2. The second request is displayed when resident SCM and LCM have been retrieved from the system library. This request indicates that installation parameters can be inserted using the STORE command. Terminate this sequence with the GCM command.

Display Format

```
GO
```

Reply Format

Type at the MCU or submit by card in the deadstart reply deck the following:

```
AUTO
```

AUTO Name of the command module in the released system

Display Format

```
GO
```

Reply Format

Type or submit by card STORE command(s).

```
STORE value,memory,parameter
```

value 1 to 7 character default setting of parameter

memory L=LCM; S=SCM

param 1 to 7 character name of parameter to be set; parameter must be defined in OST

The following parameters in LCM resident (ORL) may be set with a STORE command during deadstart.

T.MAXBUF
T.MAXS
T.MAXL
T.SYSABT
T.BUFCHK
T.SPF

All of these parameters are defined in part III section 6.2.13.

The sample SCPSID deck in part II, section 6.2.4 includes the following STORE cards.

STORE 340,L,T.MAXBUF	Maximum number of buffers that can be allocated to a job
STORE 0,L,T.SPF	SFT disk write is inhibited
STORE 160000,L,T.MAXS	Maximum user FLS; optional for full size machine
STORE 1400000, L, T.MAXL	Maximum user FLL; optional for full size machine

After submitting STORE commands, type or submit by card the following command.

```
GCM
```

6.3.12 ENTER FLAW (FOR REMAINING MASS STORAGE DEVICES)

Requests for flaws at this point occur to determine if there are any flaws on mass storage devices not containing permanent files.

Refer to section 6.3.7 and the flowchart in section 6.5 for a detailed description.

6.3.13 VOLUME LABEL GROUP RESIDES

Display Format

```
VOLUME LABEL GROUP RESIDES
CH aa U bb TRK cccc SECTOR
```

aa	Channel number
bb	Unit number
cccc	Track number
dd	Sector number

This is an informative message providing the physical disk address of the deadstart volume label group. (Part IV, section 3.6.5 contains two diagrams showing the location of the volume label group on mass storage after deadstart.) Retain the information in this message for validation during recovery (part II, section 6.4).

Reply Format

Press CR at the MCU console.

6.3.14 STATION COMMUNICATION CHANNELS

Display Format

```
STATION COMMUNICATION CHANNELS xx xx xx
xx xx xx
```

This is an informative message that provides information to the operator about the CPU channels on which the station initiate message is sent when the deadstart of SCOPE 2.0 is complete.

Reply Format

If no changes are necessary to the channels displayed at the MCU console (which is usually the case), press CR.

If changes are necessary to the information displayed, type the following at the MCU console.

```
Iaa Dbb cc
```

aa Channel number to be inserted

bb cc Channels to be deleted; more than one channel can be specified

6.3.15 DATE

Display Format

```
DATE MM/DD/YY or DATE DD/MM/YY
```

Reply Format

Type at the MCU console:

```
mm/dd/yy or dd/mm/yy
```

mm Month

dd Day

yy Year

If the installation parameter IP.YMD is set to 1, the format of the date is dd/mm/yy; if it is set to 0, the format is mm/dd/yy. The released system is set to 0.

6.3.16 OPERATOR COMMENT

The following message is displayed so that the person deadstarting can enter a comment in the SIF at deadstart. The comment cannot be in SMM command format. The first 60 characters will be entered in a system information file message, record code SISDSD which also contains information about recovery type, EST, CHQ, and values of certain parameters such as FLL and FLS.

Display Format

```
OPERATOR COMMENT
```

Reply Format

Type at the MCU console any text within previously noted limitations. Press CR if there are no comments to be entered.

6.3.17 TIME

Display Format

```
TIME HH,MM,SS
```

Reply Format

Type at the MCU console:

```
hh•mm•ss
```

hh	Hour
mm	Minute
ss	Second

6.3.18 DEADSTART COMPLETE

The following message indicates that the deadstart initialization is completed and that the operator can now log in or initialize the various stations before loading the permanent files and libraries.

Display Format

```
DEADSTART COMPLETE
 LOAD LIBRARY AND PERMANENT FILES
```

6.4 DEADSTART RECOVERY PROCEDURE

Displays and the possible replies during deadstart recovery are described as follows, in order of occurrence with references to detailed descriptions in section 6.3. Enter all replies at the MCU console; none can be entered on cards. Press CR after each entry.

The purpose of system recovery is to reestablish the operating system after system failure. During recovery, the system to recover may be specified as being on tape or a permanent file on mass storage. If recovery is to be from tape, the deadstart routine must be the same as with the initial deadstart. The system may be reconfigured as during initialization; permanent files and I/O queues may be reestablished. All error messages in part IV, section 2.6 pertain to recovery.

A flowchart of deadstart recovery follows with references to the detailed descriptions in section 6.5.

Recovery from tape may fail if changes were made in installation parameters, SCM resident (ORE), or LCM resident (ORL) since the previous deadstart.

If communication with SMM needs to be reestablished, at the MCU console type:

```
CN 10730,10460,5344,10540
```

The parameters for the CN command are described in part IV, section 3.2.3.

1. Type the following information to initiate deadstart recovery.

```
RSpp,addr,c,d
```

pp	1- or 2-digit FLPP number of system disk (FLPP with lowest number)
addr	Physical disk address (track and sector address) of volume label group that is supplied during deadstart initialization (section 6.3.13)
c	Disk control channel; default value is 2
d	Disk data channel; default value is 3

Example: RS4,2000

2. The initial action of deadstart recovery is to attempt the recovery of the SIF buffers that have not been written to system mass storage before recovery. If the relevant pointers and counts in LCM indicate that a recovery attempt might not be possible, the attempt is aborted and the following message appears on the MCU display.

```
SIF LCM BUFFERS RECOVERY ABORTED
```

Dump LCM, SCM, and/or FLPPs when necessary (part IV). Continue the recovery process after the dump by typing the RS command to restart recovery.

3. The EST display (section 6.3.1) appears on the console. It is formatted from the deadstart copy in the volume label group on system mass storage.

   The number of library buffers, the number of system tables, FLL, and FLS are recovered to the value of the previous initialization or recovery. The formats for changing these values are in section 6.3.2.

   a. To add or amend EST entries, type:

```
EST=ord,DT=dt,CH=cc1/cc2,UN=unit
```

ord	EST entry ordinal; 1 through NE.EST-1 (NE.EST=100_8 in released system)
dt	Device type and device characteristics when necessary as described in section 6.3.1
cc_1/cc_2	FLPP primary (cc_1) and secondary (cc_2) channels when necessary; 2 through 17_8
unit	Physical unit number; 0 through 77_8

   b. To delete EST entries shown in the EST display, type:

```
EST=ord
```

ord	EST ordinal to be deleted

   c. To confirm the EST as displayed, press CR.

4. The CHQ display (section 6.3.3) is displayed on the console. It is formatted from the copy retrieved with DS1 from the volume label group.

   a. To confirm the CHQ as displayed, press CR.

   b. To add or amend any CHQ entries, type:

```
CHQ=ord,DT=dt,FC=channel,pp=flpp
```

ord	CHQ entry ordinal; MUX channel number; 0 through 17_8
dt	Device type and characteristics as defined in section 6.3.3

channel — Channel configuration as described in section 6.3.3

flpp — Optional FLPP number; enter when FLPP differs from the number of the CPU channel to which it is attached

c. To specify a channel as unused, type:

```
CHQ=ord
```

5. The following message requests whether recovery is to be from tape or from mass storage.

```
GIVE SYSTEM SOURCE
```

a. To recover the system library from disk, type:

```
D
```

The disk can be dumped at this time.

b. To recovery the system library from tape, type:

```
Tchannel,unit,density,type
```

channel — 2-digit channel number through which the system library is to be sent

unit — Optional 1-digit unit number; default value is 0. If a 7611-1, 7611-2, or on-line tape unit is used to read the tape, it will be unit 0. If it is to be read from the 6000 station, the tape unit must be assigned (part II, section 5.2.2).

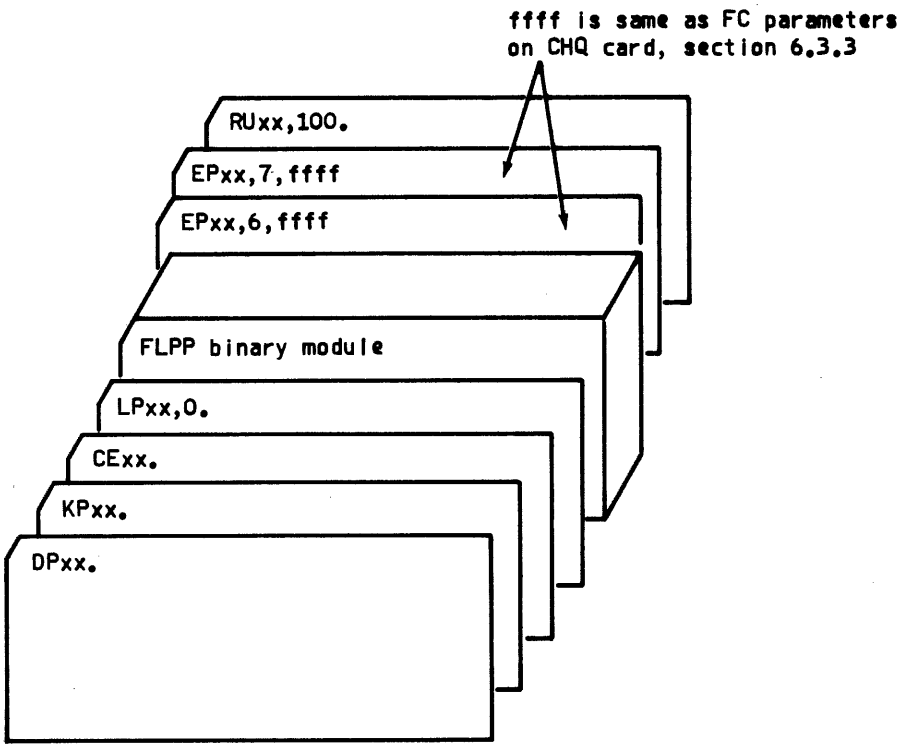
density — Optional 2-digit tape density; LO, HI, HY, or PE; HI is the default value

type — Optional 1-digit tape type; 7 or 9; 7 is the default value

If the driver for the FLPP or station at which the system library tape is to be read is not running, it must be loaded and initialized before typing the T command. Reload and initialize drivers or stations as follows.

a. If the system library is to be read from a 7611-1 station tape unit, type AUTO. in STATION mode and press CR before typing the T command.

b. If the system library is to be read from an on-line, 7611-2, or 6000 station tape unit and the FLPP is not running, reload the FLPP driver (section 6.2.1) at the MCU card reader using the following deck structure.



6. The following message is displayed only when the system library source is tape; it is displayed in order to determine if the system should attempt the recovery of permanent files. (Permanent file recovery is automatically attempted when the system library source is on disk.)

```
PERMANENT FILE RECOVERY Y/N
```

a. To attempt permanent file recovery, type:

```
Y
```

An attempt is made in this case to retrieve the PFD from the system library unit and to set the DAMs for all FATs.

b. To specify that permanent files are not to be recovered, type:

```
N
```

7. The flaw request is displayed for flaws on the system resident mass storage device. Enter any flaws not indicated in the previous deadstart for tracks other than track 20 on this device.

   The display is:

```
ENTER FLAW CchannelUunit TT,SS,NN
```

   channel Channel number; 1 through 17_8

   unit Unit number; 0 or 1

   a. To confirm existing flaws, press CR on the MCU console.

   b. To enter new flaws, type:

```
track,sector,no.
```

   track Track number as described in section 6.3.7

   sector Initial sector as described in section 6.3.7

   no. Consecutive sectors as described in section 6.3.7

8. If permanent files are to be recovered, the following message is displayed. If permanent files are not recovered, the first system deadstarted is entered into cycle 1 of the permanent file named 7000 SCOPE VERSION 2.0.

```
SYSTEM PFN AND CYCLE
```

To specify a cycle of the current pfn, type:

```
CYCcycle
```

cycle — One of the five cycles of the current pfn that is to be recovered or entered into. The cycle number must be a value from 1 to 5 of the permanent file 7000 SCOPE VERSION 2.0. Deadstart from a user catalogued file is not allowed.

If the system library is from tape, the deadstart program catalogues the system library under the specified cycle. If the system library is from disk, the deadstart program recovers the specified cycle of the system.

To recover the current cycle of the system pfn, press CR. If the system library is from disk, the deadstart program uses the same cycle of the system permanent file as in the previous deadstart. If the system library is from tape, the deadstart program catalogues the system library as the same cycle of the system permanent files as in the previous deadstart.

9. During a deadstart recovery with permanent file recovery, additional flaw requests may occur at this time for tracks other than track 20 if any permanent files are resident on a disk other than the system disk unit. Refer to step 7 for reply format.

10. After permanent file recovery is performed, the following informative message is displayed to indicate the cycle recovered.

```
SYSTEM PFN AND CYCLE
CYCLE cycle
pfn
```

x — Cycle

pfn — Permanent filename

Continue recovery by pressing CR.

11. The following display requests the command module name.

```
GO
```

a. Type:

```
AUTO
```

b. The following display message reappears.

```
GO
```

T.MAXS, T.MAXL, and T.MAXBUF are the only parameters set with STORE commands that are recovered from the previous deadstart initialization or recovery. These values (as well as values for T.SYSABT, T.BUFCHK, and T.SPF) may be changed at this time with the store command; values are defined in section 6.3.11.

```
STORE value, memory, parameter
```

c. To continue or if no changes are made with the STORE command, type:

```
GCM
```

12. If there are any mass storage devices not containing permanent files, a request for flaws (on tracks other than track 20) is displayed at this time. Refer to step 7 for reply format.

13. The following message is displayed only during a recovery of a system library from disk. The reply specifies whether to recover SFT I/O files. (There is no attempt to recover the SFT during a recovery of a system library from tape.)

```
SYSTEM FILE TABLE RECOVERY Y/N
```

a. If recovery of SFT I/O files is not wanted, type:

```
N
```

b. To recover SFT I/O files, type:

```
Y
```

A warning message is displayed for each SFT file that cannot be recovered. The message includes the SFT file job name and a reason for it not being recovered. These messages are defined in part IV, section 2.6.1. A job that is a member of a dependency string is not recovered by deadstart recovery.

A copy of the SFT is written to system mass storage periodically during system operation. (T.SPF controls the frequency with which the SFT is written to disk.) Recovery retrieves the SFT and determines its validity. If recovery finds indicators that are erroneous, the recovery of the SFT is aborted. Output files are rewound to beginning of information. Input files for which transmission was complete are recovered. Partially transmitted spooled input files must be retransmitted.

14. Save the following information to use in the next recovery (RS command) or in next deadstart initialization (VOLUME LABEL GROUP request).

```
VOLUME LABEL GROUP RESIDES
 CH aa U bb TRK  cccc SECTOR dd
```

To continue with recovery, press CR.

15. The following message is displayed to verify that the station channels given during the last deadstart are correct (section 6.3.14). When recovery is completed, a station initiate message is sent on the CPU channels listed.

```
STATION COMMUNICATION CHANNELS xx xx xx
xx xx xx
```

a. To continue recovery, press CR.

b. To insert (aa) and/or delete (bb cc) channels, type:

```
Iaa Dbb cc
```

16. The following message is displayed.

```
DATE MM/DD/YY or DATE DD/MM/YY
```

Type:

```
mm/dd/yy or dd/mm/yy
```

17. The following message is displayed so that the person deadstarting can enter a comment in the SFT at deadstart. It cannot be the same format as an SMM command. The first 60 characters are entered in a system information file message, record code SISDSD which also contains information about recovery type, EST, CHQ, and values of certain parameters such as FLL and FLS.

```
OPERATOR COMMENT
```

Type at the MCU console any text within the previously noted limitations.

18. The following message is displayed.

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

Type:

```
hh.mm.ss
```

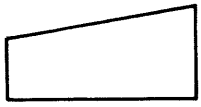
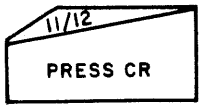
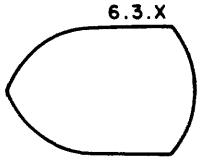
19. The following display indicates end of recovery process and requires no reply.

```
RECOVERY COMPLETE
```

The information supplied by the SYSLIBE operation in the previous deadstart is not recovered; the SYSLIB job must be run again to recover object libraries. Refer to part IV, section 3.6.3.

6.5 FLOWCHARTS

The following flowcharts summarize the possible requests and replies during deadstart. It may aid in constructing a deadstart reply deck or in determining the sequence of events at the MCU console when the SCPSID deck is read at the MCU card reader.

Symbol	Description
	Replies that can be inserted by card in the SCPSID deck or typed at the MCU console
	Replies that can only be typed at the MCU console
	It is assumed that the person installing will press CR after typing any entry (or enter an 11/12 on card). When the press CR, 11/12 symbol occurs in the chart, it means that CR must be pressed again or 11/12 entered on card again in order to continue with the next request.
	A section reference next to a display or reply symbol refers to a description of that step in the previous sections.

6.5.1 DEADSTART INITIALIZATION

The following flowchart pertains to deadstart initialization.

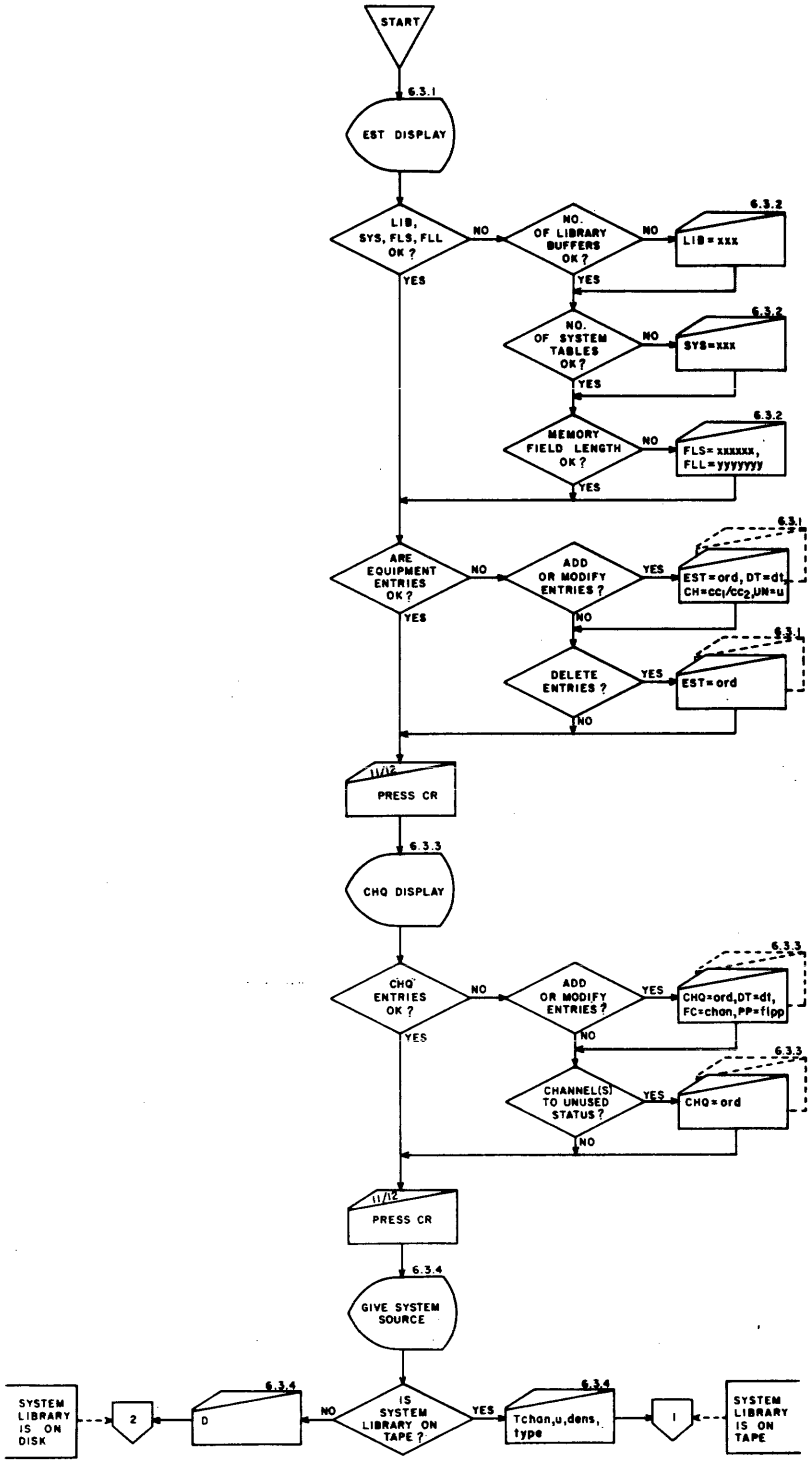


Figure II-6.3. EST Through CHQ

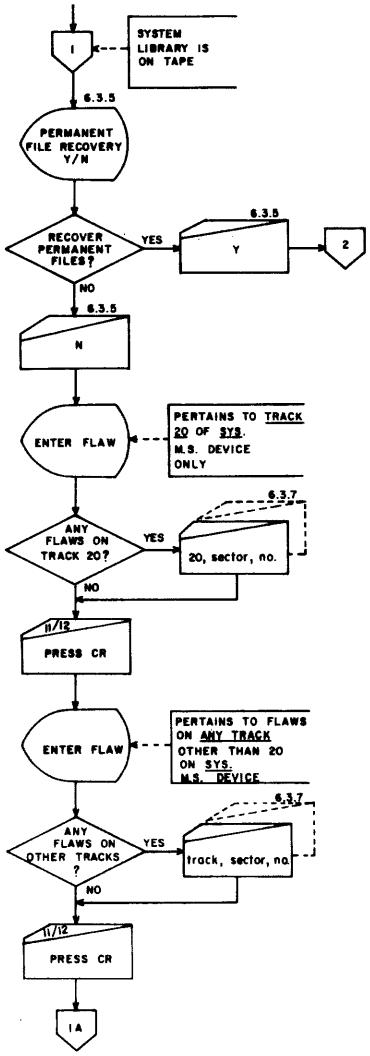


Figure II-6.4. System Library on Tape; No Permanent File Recovery

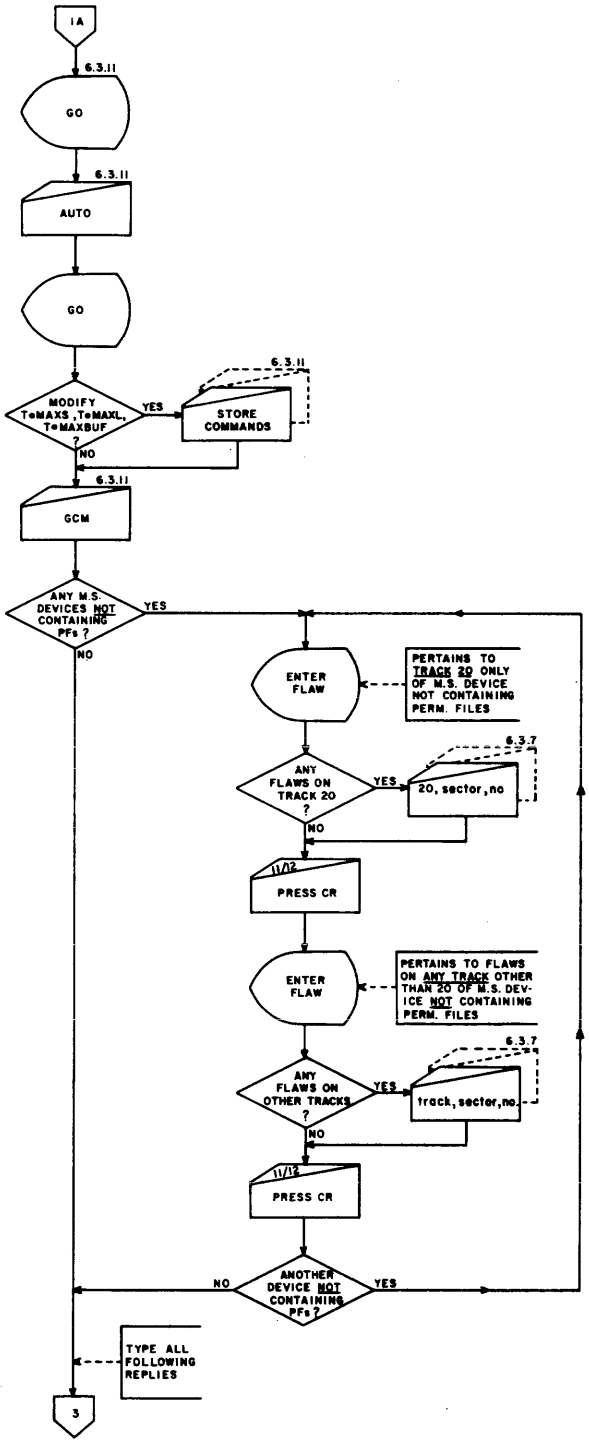


Figure II-6-4. System Library on Tape; No Permanent File Recovery (Continued)

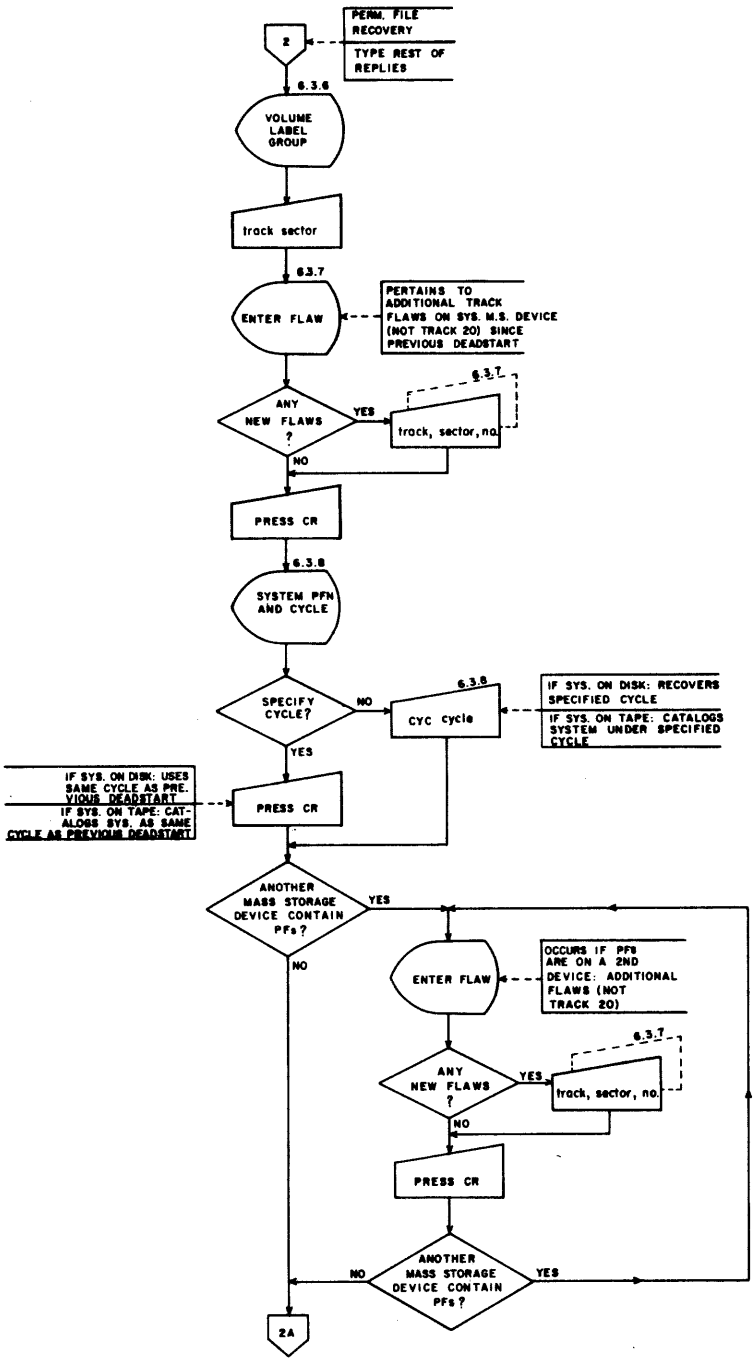


Figure II-6-5. System Library on Tape or Disk; Permanent File Recovery

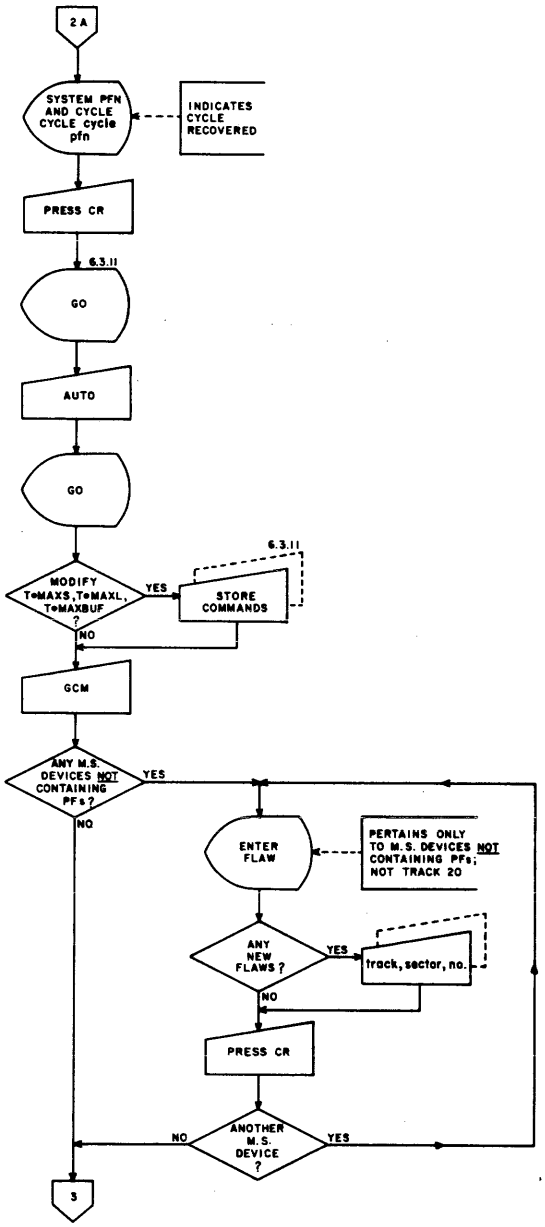


Figure II-6-5. System Library on Tape or Disk; Permanent File Recovery (Continued)

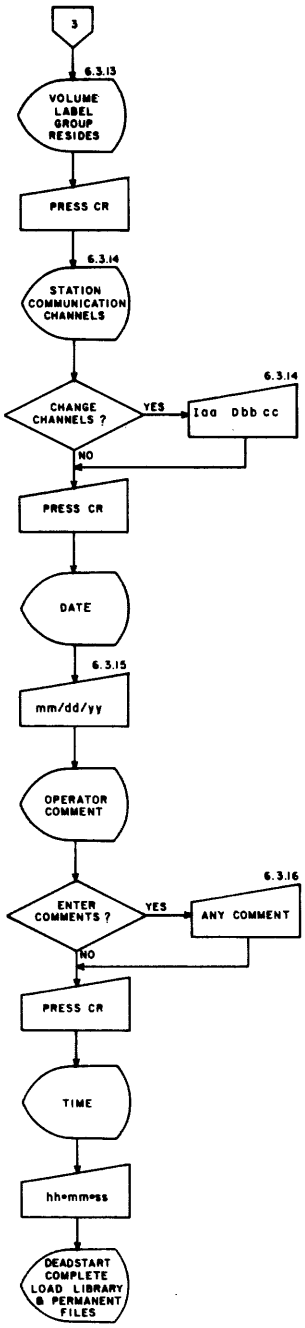


Figure II-6-6. Volume Label Group Through End

6.5.2 DEADSTART RECOVERY

The following flowchart pertains to deadstart recovery. All replies must be typed at the MCU console.

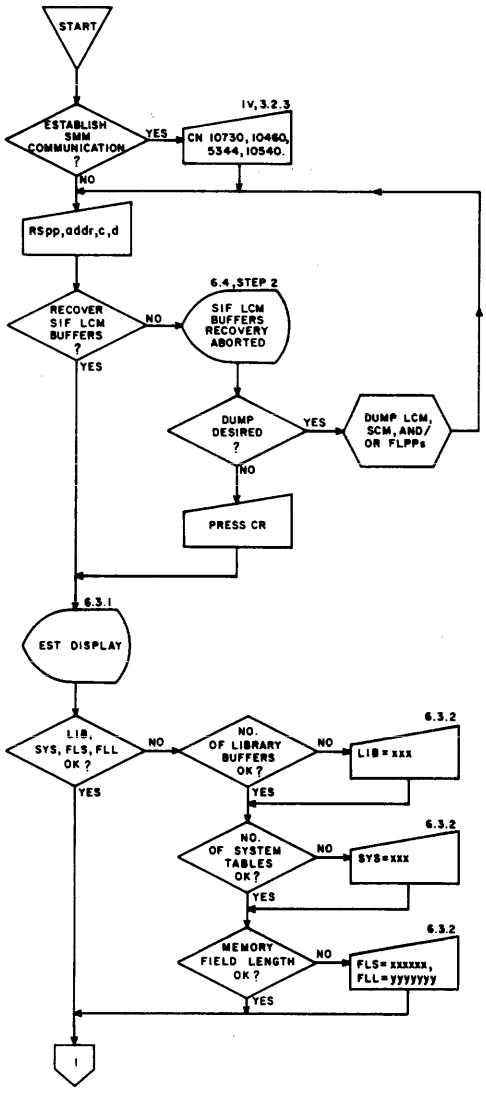


Figure II-6-7. Recovery Flowchart

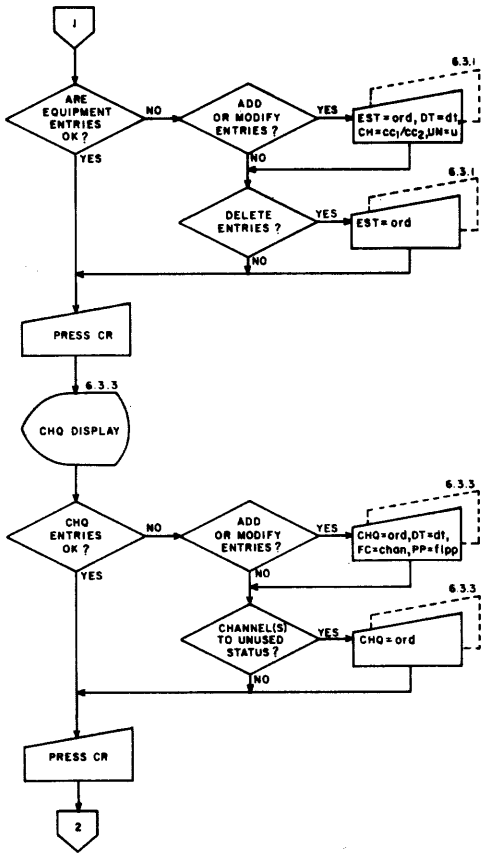


Figure II-6-7. Recovery Flowchart (Continued)

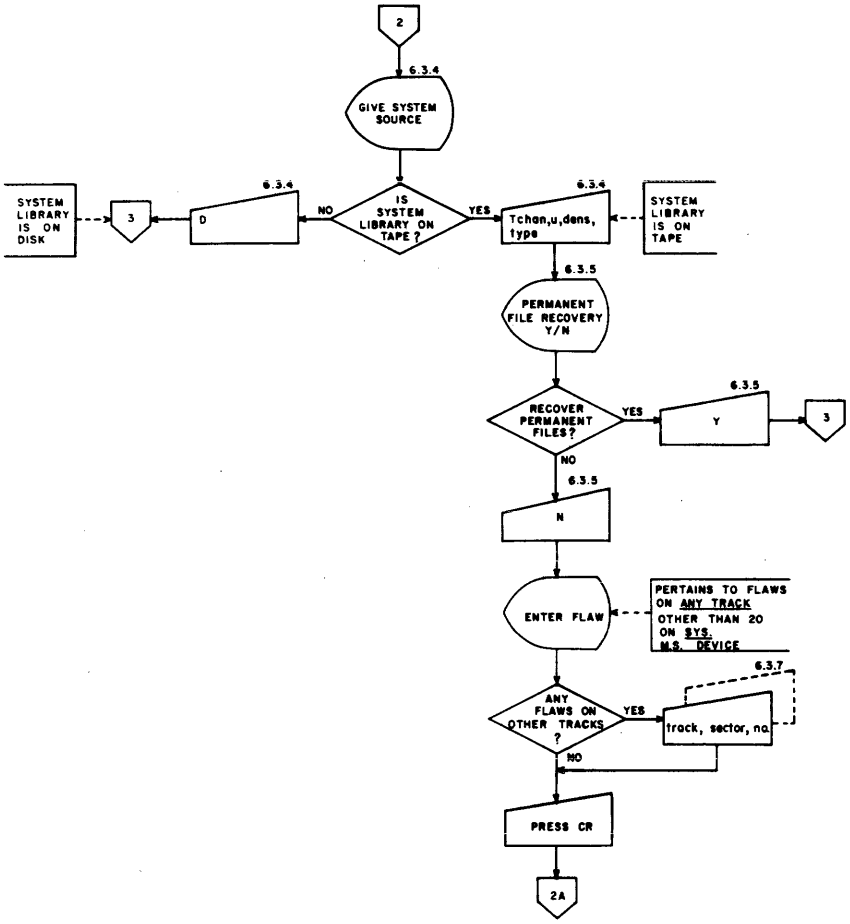


Figure II-6-7. Recovery Flowchart (Continued)

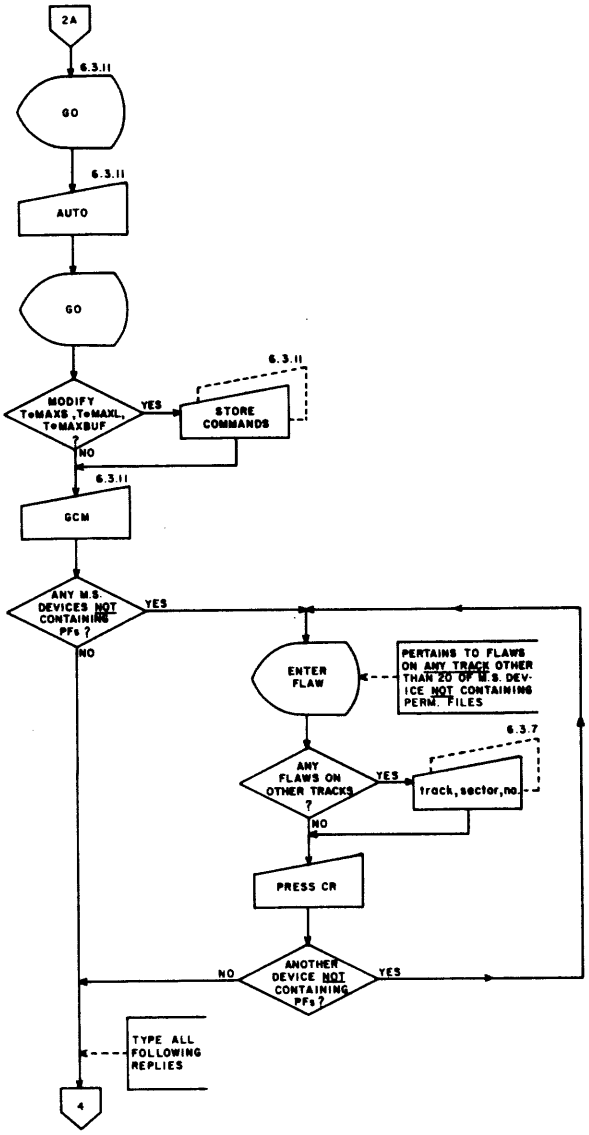


Figure II-6-7. Recovery Flowchart (Continued)

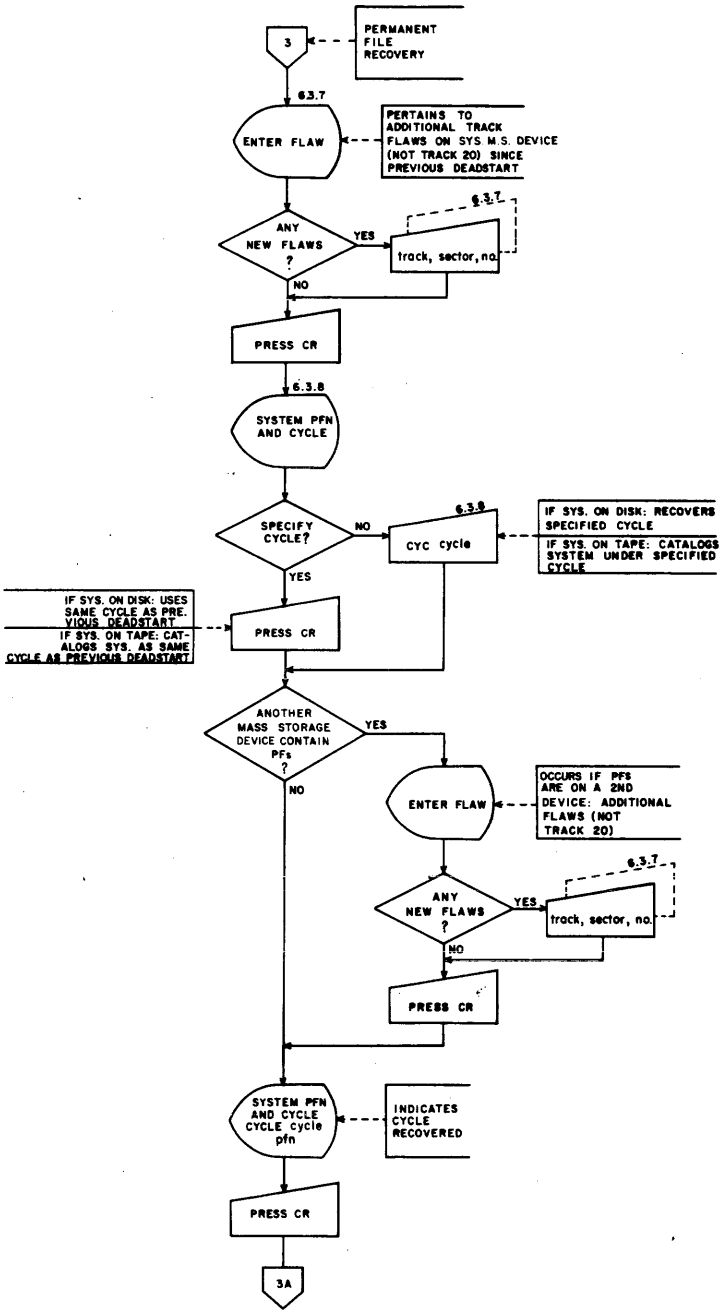


Figure II-6-7. Recovery Flowchart (Continued)

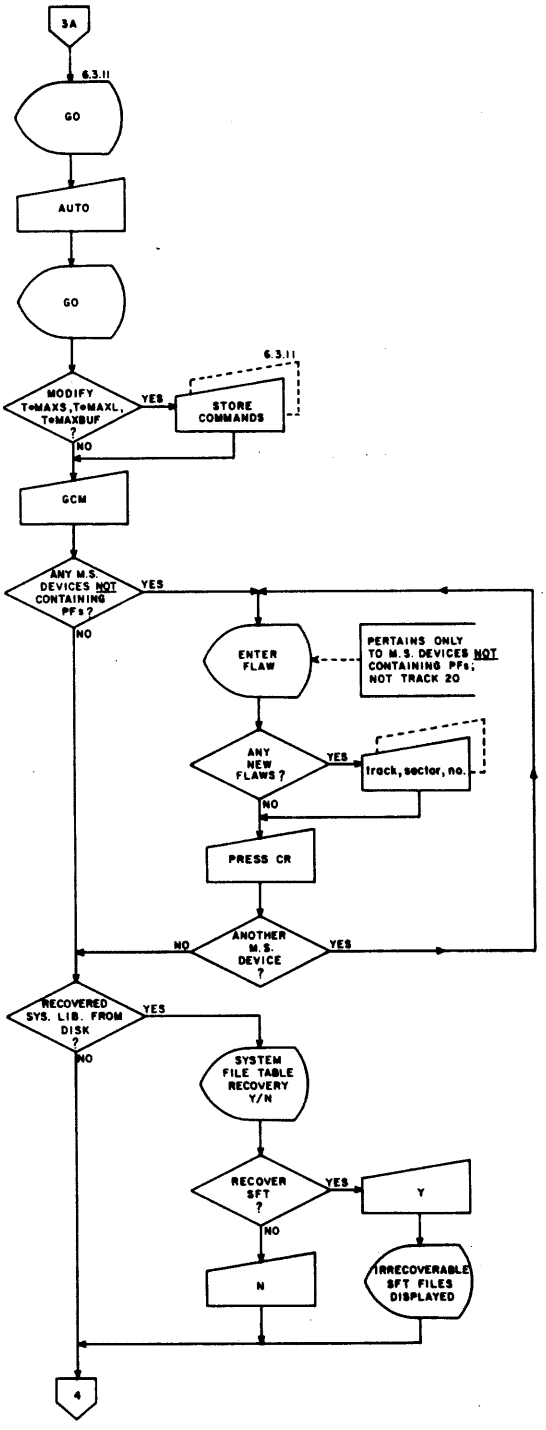


Figure II-6-7. Recovery Flowchart (Continued)

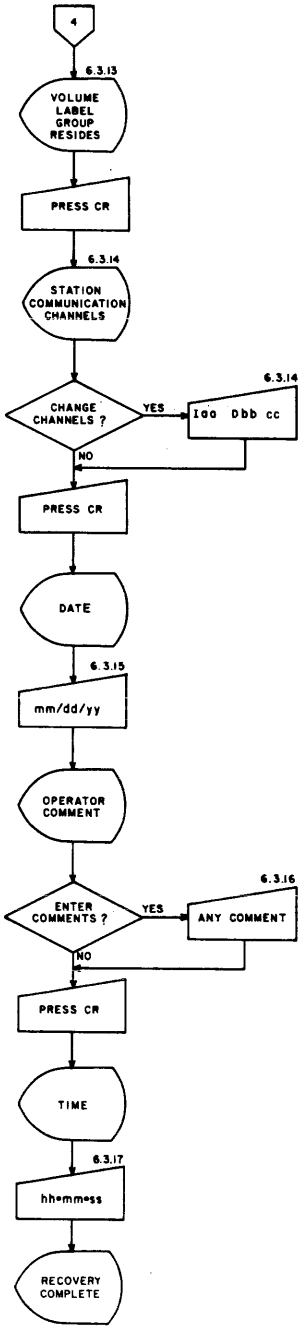


Figure II-6-7. Recovery Flowchart (Continued)

SCOPE 2.0 MAINTENANCE PACKAGE 1

1.1 MODTAPE

The SCOPE 2.0 maintenance package is on the release tape labeled MODTAPE.

1.1.1 CONTENTS

This tape contains the following code.

Corrections to SCOPE 2.0 and product program libraries

Job decks to make these corrections to the released program libraries and to create new program libraries

Job decks to perform other functions needed to install and maintain the system

A worksheet that contains the procedures necessary to initially install or to subsequently update the system

1.1.2 FUTURE RELEASES OF MODTAPE

MODTAPE is part of the initial SCOPE 2.0 release package. Every two weeks after receiving the initial release materials, the customer receives a Programming System Report (PSR summary). This PSR summary lists customer inquiries concerning the system and product set as well as answers to some inquiries. The PSR summaries are numbered consecutively.

When the customer receives the odd-numbered PSR summary (every fourth week), he also receives a new MODTAPE containing corrective code to update the release materials to the current PSR level. All code on the new MODTAPE has been published in the preceding even-numbered PSR summary or in the accompanying odd-numbered summary.

However, some code published in the summaries may cause problems that are identified only after extensive testing. This problem usually occurs only with code presented in even-numbered summaries because this code has not undergone a full system test. Therefore, the retraction of code is usually noted in the odd-numbered summaries, and of course, is not included in that MODTAPE. The MODTAPE represents a fully tested system with no known regressions.

The installation decks on updated MODTAPEs are modified to reflect the current PSR summary level and any necessary changes in build procedures.

1.1.3 STRUCTURE

MODTAPE contains two partitions: MODPL which contains corrections to the released software and SYSDECK which contains installation decks.

MODPL

This partition contains modifications to the released software in UPDATE format with the master control character set to /. Its structure is as follows.

There is a /DECK on MODPL for each product's UPDATE program library. These /DECKs are named xxxPSR with xxx identifying the program library.

xxx	Deck Name	Product
STA	STAPSR	7611-1 I/O Station
MTS	MTSPSR	7611-2 Magnetic Tape Station
SO	SOPSR	SCOPE 2.0 operating system
CM2	CM2PSR	COMPASS 2.0
FN2	FN2PSR	FORTRAN Extended 2.0
RN2	RN2PSR	FORTRAN Run 2.0
FCL		FORTRAN object time routines; located on the SCOPE 3.4 modification tape
COB	COBPSR	COBOL 1.0
SRT	SRTPSR	Sort/Merge 1.0
DIA	DIAPSR	Diagnostic Control Program

The corrective code for each PSR is in its own /COMDECK. Each product's xxxPSR deck contains /CALLs to all the /COMDECKs that contain corrective code for that product's program library.

There are also /DECKs with the name of the individual PSRs that call the same /COMDECKs.

For example, for a MODPL containing three PSRs for product xxx (xxx0001, xxx1200, and xxx1342), there are three /COMDECKs (Dxxx0001, Dxxx1200, and Dxxx1342) and three /DECKs (xxx0001 containing a /CALL to Dxxx0001, xxx1200 containing a /CALL to Dxxx1200, and xxx1342 containing a /CALL to Dxxx1342). The xxxPSR deck calls the /COMDECKs for all three PSRs. A similar structure is repeated for each product's program library. The /IDENT contained in each /COMDECK usually has the same name as the /COMDECK, but they need not always be the same.

SYSDECK

This partition contains job decks in UPDATE format with the master control character set to $. These job decks are used to update and generate materials during the normal system integration activities by SCOPE 2.0 developement and are catagorized in the Pricing Manual as level 3 support. SYSDECK contains the following kinds of jobs.

Installation decks using the binaries from released program libraries and optionally

General decks to generate binaries from released program libraries and optionally incorporate modifications (either from MODPL or from the customer) to produce new release materials

Utility decks to generate the configured system

Decks to verify that the installation is complete and correct

Decks to produce listings useful in maintaining the system

Decks to produce a description of the installation and modification procedures described in this section

1.1.4 PROCEDURE

The procedure in sections 1.2 through 1.6 describe the installation process as:

1. Initial installation using the initial set of release materials that are referred to in sections 1.2.1, 1.2.2, and 1.2.3 and the procedures described in section 1.2.4

2. Installation of PSR corrections that are on the monthly MODTAPEs according to information in section 1.3; this results in a new set of release materials updated to the current PSR summary level

3. Installation of customer modifications to create new materials as described in section 1.4

4. Creation of back-up materials as described in section 1.5

5. Creation of maintenance listings, such as source listings, symbol and macro cross reference listings, and system overlay listings as described in section 1.6

The use of this procedure requires the customer to build a new system with each MODTAPE, even if the new system is not used for production. This is necessary because each subsequent system is built using the one which immediately preceeded it.

The basic building of the system should not occur in a batch (production) environment. However, there are batch notes in this section that specify areas that require modification in order to run in a batch environment. There are also several housekeeping hints to expedite installation.

In the following sections, the phrase product set refers to any part of the system that executes primarily or entirely in the user field length. Therefore, it includes the products that are unbundled (priced separately) as well as COMPASS and DCP. The products that are part of the SCPOPL, such as LIBEDT, UPDATE, and ANALYZE, are not product set members.

1.2 ESTABLISH BASE SYSTEM MATERIALS

1.2.1 TAPES

The initial release tapes described in part I, section 2 are necessary to build a base system. New release tapes (production release tapes) are created using the build procedure in section 1.2.4.

One additional tape is required, a DUMPF/LOADPF tape of all files cataloged with ID=PRDLIB; refer to the permanent files description in section 1.2.2. The worksheet referred to in part III, section 1.7 contains detailed information on all tapes that can be used in the build procedure.

1.2.2 PERMANENT FILES

The use of the SYSDECK procedure outlined in section 1.1.4 requires that various permanent files be created and/or modified. The IDs for these permanent files are:

ID	Function
PRDLIB	Use for files containing executable binary for the system.
S20CPLOPL	Use for files containing program libraries for the product set.
S20OPSOPL	Use for files containing program libraries for the operating system.
S20OPSMOD	Use for the SCOPE 2.0 maintenance package.
TOOLS	Use for files containing utility programs.
SCPSCR	Use for temporary files only during the build procedure.

These IDs are used so that files may be manipulated using the permanent file ID feature. With the exception of PRDLIB files, all permanent files are recreated (if desired) during each system build. They are not used as input to the next build.

PRDLIB

The files cataloged with ID PRDLIB contain the binaries of the SCOPE 2.0 operating system and product set. Subsequent installations of the system create higher cycles of these files; the lower cycles of these replaced files are purged. Make a DUMPF tape containing the highest cycle of each file for back-up purposes.

The procedure to build the LOADPF tapes for ID=PRDLIB requires that each system be built using the system which immediately preceeded it. The LOADPF procedure is:

1. Load the LOADPF tape (ID=PRDLIB) from the previous installation.

2. Build new cycles of files for products to be modified.

3. Execute ATTACH cards for the highest cycles (cycle parameter omitted).

4. Execute a DUMPF (MO=3, IN=2, PW=TYPE3, ID=PRDLIB) to purge inactive files (in this case files which are not the highest cycles).

5. Create a DUMPF tape containing files (each of which is the highest cycle) for use in a batch environment and for use as input to the next modification build.

This LOADPF procedure is not possible in a batch environment. Step four must be delayed until after production when there are no more references to the old files. The old files may be purged with a constructed job deck or by rerunning the SYSDECK job called DPIDPRD. DPIDPRD performs steps three through five with a GO/DROP pause before step four. If a DROP is entered, the job proceeds to step five immediately.

The permanent files with ID=PRDLIB are described as PRDLIB, object libraries, SRTMACS, and SYSLIB.

PRDLIB Permanent File: The permanent file PRDLIB is a library that contains the core image binaries for the product set. For an initial installation, use the OPRDLIB job to create the PRDLIB file. OPRDLIB generates a library from the COMPASS and DCP binaries on the release SCPLIB tape.

For subsequent installations, load the PRDLIB file from the PRDLIB LOADPF tape. As the products are installed, the temporary libraries (xxxLEL) are built. When all of the products are installed, create a new PRDLIB using the NPRDLIB job which updates the old PRDLIB from all of the xxxLELs.

Batch notes: With various cycles of PRDLIB cataloged, it is possible to use editions of the product set other than the one on the deadstart tape. Use the following procedure to pretest new compilers in a batch environment or to use past editions of the compilers. Again be sure that the correct cycle of PRDLIB is attached during the installation process. The following sequence compiles and executes code using the compiler from PSR summary 47 and the object library from PSR summary 45.

```
ATTACH(PRDLIB,PRDLIB,ID=PRDLIB,
CY=47)
LIBRARY(PRDLIB)
FTN.
ATTACH(IOLIB,FTNIOLIB,ID=PRDLIB,
CY=45)
LIBRARY(IOLIB)
LGO.
LIBRARY.
```

Object Library Permanent Files: The object time routine libraries corresponding to the various compilers are individual permanent files, each one with ID=PRDLIB. Catalog these files and use SYSLIB to merge them into the running system after all deadstarts to satisfy external references in compiler generated binaries before execution.

For an initial installation, create these files by passing relocatable binary copies from release tapes through LIBEDT and cataloging the libraries that are produced.

To produce the FORTRAN common library (which SCOPE 3.4 and SCOPE 2.0 share), update, assemble, compile, pass through LIBEDT, and catalog the binary.

For subsequent installations, replace each library by cataloging a higher cycle of the permanent files.

Batch notes: Running system generation jobs in a batch environment conflicts with jobs that attach the object library permanent file and omit the cycle number; therefore, they attach the highest cycle.

To use a different permanent file name of ID for accounting purposes, change the permanent file control statements in the installation, generation, and SYSLIBE decks. Also, since the COBOL build precedure attaches the SORTMRG object library and the SCOPE 2.0 build procedure attaches the FORTRAN Extended object library, modify these jobs.

Make similar modifications for ID changes. Unless all PRDLIB IDs are changed to a single, different ID, the customer must also devise ways to create back-up tapes and to reload files for subsequent installations.

SRTMACS Permanent File: The permanent file SRTMACS contains an XTEXT file that may be used to assemble COMPASS code containing SORT macros. To use the SRTMACS file, attach it to COMPASS specifying it as an XTEXT file.

Batch notes: The problems noted with object library permanent files also pertain to SRTMACS except that only the decks involving Sort need to be changed.

SYSLIB Permanent File: The SYSLIB permanent file contains the binary of SCOPE 2.0 as assembled from SCPOPL. This file may be used to build variations of the released system, such as a system containing accounting overlays.

Batch notes: Customer decks should attach SYSLIB using specific cycle numbers.

S20CPLOPL

There are job decks on SYSDECK that catalog each product set program library with the ID=S20CPLOPL. There is also a deck to create a DUMPF tape for this ID. Execution of these decks is completely optional and is not required for system installation. The decks are provided as a convenience to customers who reference these files frequently enough to warrant their existence as permanent files.

Batch notes: Refer to the cycle number caution described in previous batch notes.

S20OPSOPL

The job deck SCPPL catalogs SCOPE 2.0 with the ID S20OPSOPL. There is also a deck to create a DUMPF back-up of this ID. Creation of this file is optional; but it is necessary to create maintenance listings. Refer to part III, section 1.6.

Batch notes: Refer to the cycle number caution described in previous batch notes.

S20OPSMOD

Catalog the MODPL and SYSDECK program libraries with the ID S20OPSMOD. Both files are required during execution but may be purged afterwards. Both files are discussed in detail throughout this section.

Another file that may be cataloged with this ID is USERMODPL. It is intended to contain customer modifications (other than installation parameters) to the operating system and the product set. USERMODPL is an UPDATE program library (master control character set to /) containing one /DECK for each OPL in the release materials. The deck name format is /DECK xxxMOD. xxx identifies the product.

xxx	Deck Name	Product
STA	STAMOD	7611-1 I/O Station
MTS	MTSMOD	7611-2 Magnetic Tape Station
SCP	SCPMOD	SCOPE 2.0 operating system
CM2	CM2MOD	COMPASS 2.0
FN2	FN2MOD	FORTRAN Extended 2.0
RN2	RN2MOD	FORTRAN Run 2.0
FCL	FCLMOD	FORTRAN object time routines
COB	COBMOD	COBOL 1.0
SRT	SRTMOD	Sort/Merge 1.0
DIA	DIAMOD	Diagnostic Control Program

The contents of each deck should be UPDATE directives (master control character set to *) to apply customer modifications to the products. The generation decks may be tailored to attach USERMODPL and to apply customer modifications to the newly created release materials to produce a modified system.

If using USERMODPL and adding new decks, use a *ADDFILE lfn, deckname directive only if lfn is UPIN or a local scratch file.

The decks on SYSDECK assume that modifications come from a file called UPIN, for example, UPDATE (I=UPIN,...). Do not attempt to execute the ADDFILE directive from the INPUT File.

TOOLS

There are utility programs on SYSDECK written in FORTRAN Extended and COMPASS code. During the build process these programs are compiled and the binaries are cataloged. TOOLS is the ID. For more detail refer to part III, section 1.6.

1.2.3 CARD DECKS

The following card decks must be available to install the system.

Binary Deadstart Decks

These decks are part of the SMMB and station release materials described in part I, section 2. New versions of these decks are generated during a modification of the system.

SCOPE 2.0 SCPSID Deadstart Deck

The SCPSID deck is described in part II, section 6.2. It consists of FLPP and DS1 binary decks and the deadstart reply deck. The binary decks depend upon the PSR summary level of the software and are generated during each system modification. The deadstart reply deck depends upon the hardware system (memory sizes, channels, equipment, flaws) and is not changed for each system modification. It is useful, therefore, to create a single reply deck that may be used with the binary decks for several different systems. Back-up copies are advantageous. It is also useful to have several deadstart reply decks if the customer has several 7000 mainframes.

SYSDECK Job Decks

The MODCAT job catalogs the maintenance package, and the PNCHDKS job is used to specify options.

MODCAT: Run the MODCAT job to catalog MODPL and SYSDECK as permanent files, list the SYSDECK worksheet, and punch the PNCHDKS deck from SYSDECK. The following job reflects the MODCAT version on SYSDECK containing the correct cycle numbers.

JOB DECK MODCAT

```
MODCAT,CP70,T100,MT01.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *    THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED       *
COMMENT. *                                                          *
COMMENT. *    BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA  *
COMMENT. *                                                          *
COMMENT. *    FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)    *
COMMENT. *                                                          *
COMMENT. *    STAGE HAS BEEN DEFINED                                *
COMMENT. *                                                          *
COMMENT. *    PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
ATTACH(A,MODPL,ID=S200PSMOD,PW=SCPCN,
CY=51)
PURGE(A)
EXIT(U)
RETURN(A)
ATTACH(A,SYSDECK,ID=S200PSMOD,PW=SCPCN,
CY=51)
PURGE(A)
EXIT(U)
RETURN(A)
REWIND(OUTPUT)
LABEL(MODTAPE,L=$MODTAPE*SCP2.0$)
STAGE(MODTAPE,MT,HI,E,PRE,
VSN=CY51)
COPYP(MODTAPE,MODPL)
COPYP(MODTAPE,SYSDECK)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG MODPL AND SYSDECK                                *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(MODPL,MODPL,ID=S200PSMOD,PW=SCPCN,CN=SCPCN,EX=SCPEX,MD=SCPMD,
CY=51)
CATALOG(SYSDECK,SYSDECK,ID=S200PSMOD,MD=SCPMD,CN=SCPCN,EX=SCPEX,
PW=SCPCN,
CY=51)
UNLOAD(MODPL,SYSDECK)
COMMENT. ************************************************************
```

JOB DECK MODCAT

```
COMMENT. *                                                                    *
COMMENT. * LIST THE MEMO AND WORKSHEETS FROM SYSDECK. THE MEMO LISTS PRODUCTS  *
COMMENT. * FOR WHICH MODS EXIST ON THE MODPL, AND NOTES THOSE MODS WHICH MAY  *
COMMENT. * HAVE UNUSUAL IMPACT ON THE USER. SIGNIFICANT CHANGES TO SYSDECK    *
COMMENT. * WILL ALSO BE NOTED. THE WORSHEETS SERVE AS A TOOL FOR INSTALLATION. *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
ATTACH(OLDPL,SYSDECK,ID=S200PSMOD,
CY=51)
UPDATE(P,Q,D,K,L=A134,*=$$$$)
COMPASS(I,B=LISTER,L=LIST)
LISTER(I,P=0,TD=4,CD=5,MEMO,L=MEMO,
CY=51)
REWIND(MEMO)
COPY(MEMO,OUTPUT)
COMMENT. ******************************************************************
COMMENT. *                                                                    *
COMMENT. * PUNCH (AND LIST) PNCHDKS - THE JOB USED TO LIST AND PUNCH ALL OTHER *
COMMENT. * DECKS NEEDED FOR INSTALLATION. PNCHDKS MAY NEED TO BE MODIFIED     *
COMMENT. * TO TAILOR THE DECKS AS DESIRED. NOT ALL DECK WILL BE NEEDED        *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
LISTER(I,L,P,DECK,SET=INS,
CY=51)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
$ID MODCATSUP
$/  A $DF STAGE   CARD WILL CAUSE $DF STAGE CARDS TO BE PLACED IN PNCHDKS
$/  WHICH WILL CAUSE DECKS PUNCHED BY PNCHDKS TO HAVE STAGE CARDS INSTEAD
$/  OF REQUEST CARDS FOR ON-LINE TAPES.
$/
$/
$DF STAGE
$C LISTER
$C MEMO   INTRODUCTION TO MODTAPE, DESCRIPTION OF MODPL AND SYSDECK
$C WORKSHEET
$C PNCHDKS
---EOI--- 6/7/8/9
```

PNCHDKS: The worksheet indicates which program libraries the PSR summary is to modify. Only punch the jobs that are needed to install the system by pulling out the unnecessary $C cards in the PNCHDKS job. There are four updates in the PNCHDKS job.

Installation	The decks in this section are related to an initial installation. Included are various utility decks that are also useful in installing PSR updates to the release materials.
PSR modification	The decks in this section apply PSR updates to previous release materials to create new release materials.
Customer modification	The decks in this section apply customer modifications to the release materials with the current PSR summary level modifications.
Utilities	The decks in this section are of a general utility nature. Included are decks to copy tapes, copy decks, catalog files, create DUMPF back-up tapes, and produce maintenance listings.

The decks are discussed in more detail in sections 1.3 through 1.6 of part III.

After determining which decks are required, select the options necessary to tailor the decks. Generation decks (decks that assemble or compile the source to produce the binary for the corresponding element of the system) may be modified using $DEFINE options. Select these options by placing a $DEFINE card into the input section of the appropriate UPDATE in PNCHDKS. The options are:

Option	Description
$DEFINE LIST	If this option is defined, all generation decks (xxxGEN, xxxMOD, and xxxUSR) produce assembly listings, with the exception of SCPGEN. To produce SCOPE 2.0 listings, use the volume listing decks described in part III, section 1.6. If it is not defined, assemblies and compilations do not produce listings.
$DEFINE NONEWPL	If defined, all generation decks omit the creation of a new program library on tape. If not defined, the decks create new program libraries.
$DEFINE EXTEND	If defined, all generation decks attach and extend existing permanent files. If not defined, new cycles of permanent files are cataloged.
$DEFINE STAGE	If defined, all REQUEST statements are replaced in the job decks with STAGE statements.
$DEFINE MODTAPEMD	If defined, generation decks take PSR modifications from the MODPL permanent file.
$DEFINE USERMODS	If defined, generation decks take customer modifications from the USERMODPL permanent file.

Combinations of $DEFINE cards produce decks for specific purposes.

Option	Description
LIST, NONEWPL	Regenerates binary from a program library producing a listing; catalogs binary as a new file.
MODTAPEMD, STAGE	Updates a previous release tape from MODPL to produce a new release tape (staging both tapes); creates binary without an assembly listing; catalogs results as a new permanent file.
USERMODS, EXTEND, NONEWPL	Updates a current release tape (on-line tape) from USERMODPL to produce a COMPILE file but not a new program library; assembles without listings; and extends an existing permanent file of the binary with the new binary.

There are two other $DEFINE options: ARHOPS and XPRD. They tailor the jobs for CDC use. The customer is not required to use these options.

Do not define USERMODS and MODTAPEMD at the same time.

The following conventions apply to job names. Most job names are of the form xxxyyy. xxx is the mnemonic for the product as previously described, and yyy indicates the action. The yyy values for SYSDECK jobs are:

yyy	Action
GEN	Job assembles code with installation options changed. A new program library may or may not be created. Neither USERMOD or MODTAPEMD are defined.
MOD	Job applies PSR modifications before assembling code. A new program library is usually generated. MODTAPEMD is defined.
USR	Job applies customer modifications before assembling code. A new program library may be generated. USERMODS is defined.
INS	Job copies binary from a release tape and processes it.
CPY	Job copies release tape.
PL	Job catalogs release program library.
DCK	Job creates a deadstart deck for product.
VFY	Job verifies installation of product.

Housekeeping hint: Job cards created with $DEFINE STAGE still have the MT parameter defined on the job card. For installation with a small number or no tapes, change the job cards.

Housekeeping hint: Permanent file CATALOG cards in SYSDECK jobs are constructed to catalog a file regardless of the existence of a previous cycle. This is done by both defining the passwords (EX=, MD=, etc.) and specifying them (PW=) and the cycle number. The SCOPE 2.0 permanent file manager uses the password definition if no previous cycle exists and uses the password specification if previous cycles exist. The passwords for all files cataloged by SYSDECK jobs are: EX=SCPEX, MD=SCPMD, CN=SCPCN. The passwords for RD and TK are null.

1.2.4 PROCEDURES

The procedures to establish a base system with the initial release materials are:

1. Deadstart the MCU with the SMMB deck (part II, section 2).

2. Deadstart stations (part II, sections 3, 4, and 5).

3. Deadstart the 7600 with the latest SCOPE 2.0 system (part II, section 6). For an initial installation, construct the deadstart SCPSID deck (part II, section 6.2).

4. Run the MODCAT job to catalog MODPL and SYSDECK, assemble the MEMO LISTER utility, print the formatted MEMO, and list and punch a card copy of PNCHDKS.

5. Modify and run the job PNCHDKS to list and punch the required installation decks from SYSDECK.

6. For an initial installation, run the OPRDLIB job to create an initial copy of the PRDLIB permanent file.

7. For subsequent applications of PSR updates, run the LDIDPRD job to catalog the object libraries, SRTMACS, PRDLIB, and SYSLIB.

Proceed with the next sections to install and configure the base system.

Batch notes: Each SYSDECK job calls two common decks: ACCOUNT and EXIT. ACCOUNT is called immediately after the job card in each job. If the jobs are to be run with a batch system and an ACCOUNT card is required, insert appropriate ACCOUNT cards into the common deck and it will be punched out in each deck. The common deck EXIT is called in most jobs at the end of the control statement section. It contains a PAUSE indicating a successful completion, followed by an EXIT, and then a PAUSE indicating failure. This allows job completion to be noted when using the worksheet. It may not be desirable to do this in a batch environment. To eliminate the pauses, delete the contents of EXIT (but not the common deck itself).

Housekeeping hint: Use the LISTER program to format listings of SYSDECK installation decks and the SYSDECK worksheets. This program accepts several keyword parameters that control the number of lines per page, depth of detail in the table of contents, and page ejects on section headers. For more detail refer to the listing of LISTER.

Housekeeping hints: Since there are so many decks punched by PNCHDKS, the user can reduce the number of cards by deleting the contents of the common decks ACCOUNT, COMMENTB, and COMMENTE. These decks only contain comment cards. This should not be done without a complete familiarity with the SYSDECK procedures. The only manual action required is to replace the 7/8/9 17 cards with 6/7/8/9 cards. Minimally, interpret the job cards.

1.3 BUILD A PRODUCTION RELEASE SYSTEM

Building a release system suitable for production involves one or more of the following procedures for each element of the system.

Generate binary from a program library with modified installation parameters using xxxGEN job

Generate binary from a program library updated from PSR modifications using xxxMOD job

Process existing binary to produce a production format using the xxxINS job

The worksheets described in part III, section 1.7 list the tapes and permanent files that are required and created by these jobs as well as the order of execution that is necessary.

These procedures are described more specifically for the stations in section 1.3.1, for the product set in section 1.3.2, for the FORTRAN object library in section 1.3.3, and for the SCOPE 2.0 operating system in section 1.3.4.

1.3.1 STATIONS

Both the generation jobs (STAGEN and MTSGEN) and modification jobs (STAMOD and MTSMOD) allow the setting of installation parameters. Both create deadstart tapes for initial deadstarts. An initial deadstart copies the deadstart tape to the station disk pack. After the initial deadstart, the station software may be deadstarted from the station disk pack.

1.3.2 PRODUCT SET MEMBERS

The first step for each product set member in building a production release system is to catalog the core image binary as an xxxLEL library file (ID=SCPSCR). There are three ways to do this.

1. Use the xxxINS job to copy the binary from the release tape to mass storage, pass it through LIBEDT, and catalog it.

2. Use the xxxGEN job to update the released program library (defining new installation parameters), assemble or compile it, load it if necessary, pass it through LIBEDT, and then catalog it (run SRTGEN before running COBGEN and COBMOD).

3. Use the xxxMOD job to update the previous release program library from the PSR modifications on MODPL (with possible changes to installation parameters), assemble or compile it, load it if necessary, and then catalog it. The xxxMOD job takes its corrective code from the /DECK xxxPSR (run SRTMOD before running COBMOD or COBGEN).

The next step is to run the NPRDLIB job to combine the separate libraries into one product library PRDLIB (ID=PRDLIB). This file is then used to create a deadstart tape.

For the COBOL and Sort/Merge products, catalog the associated object libraries or XTEXT files. xxxINS jobs process existing binaries; xxxGEN jobs regenerate new copies of the binaries; and xxxMOD jobs apply PSR updates to create new binaries. COByyy creates the COBOL object library COBLIB; SRTyyy creates the Sort/Merge object library SRTLIB and the SRTMACS XTEXT files. All IDs are PRDLIB.

1.3.3 FORTRAN OBJECT LIBRARY

FCLOPL contains the source for the FORTRAN object library. This source is common to both 6000 SCOPE 3.4 and 7000 SCOPE. Customers who use both of these systems should maintain a single version of the program library.

The release tape does not contain binaries for the product because the two systems are not completely compatible at the binary level. Therefore, there is no installation job either. For an initial installation, generate the binary from the FCLOPL tape.

Actually, there are two versions of the SCOPE 2.0 binary. One is assembled to be used with FORTRAN Extended code, the other to be used with FORTRAN Run code.

The procedure to build the FORTRAN object libraries is:

1. If needed, run the FTNLIB7 job to apply PSR modifications to the release program library. The modifications to this program library are on the SCOPE 3.4 modification tape. This job produces a new release tape. A similar procedure may also be used under SCOPE 3.4.

2. Run RLIBGEN to create the permanent file RUNIOLIB (ID=PRDLIB) from the current program library.

3. Run FLIBGEN to create the permanent file FTNIOLIB (ID=PRDLIB) from the current program library. Do not run this job until the file PRDLIB is cataloged.

1.3.4 SCOPE 2.0

Use the following procedure to create a new SCOPE 2.0 deadstart tape that contains the binary of the product set.

1. Run the SCPINS job to update an existing deadstart tape from the PRDLIB file, produce a new deadstart tape, and create a permanent file SYSLIB (ID=PRDLIB) containing the operating system binary.

2. Run the SCPMOD job to update SCPOPL with PSR modifications to produce a new release tape, assemble the system, catalog the binary as the permanent file SYSLIB, and produce a deadstart tape by combining SYSLIB with PRDLIB.

3. Run the SCPGEN job to generate binary from the release tape, catalog the binary as SYSLIB, and create a deadstart tape by combining SYSLIB with PRDLIB.

The SCPINS job adds product set binary to the release deadstart tape SCPLIB, adds new compilers to an old system, or adds old compilers to a new system. The SCPMOD job updates release materials. Either the SCPMOD or the SCPGEN job may be used to redefine installation parameters.

After running the SCPxxx job, run the DBUGINS job to catalog the object library DBUGLIB which is necessary to use the TRAP feature of the SCOPE 2.0 loader.

1.4 ADD CUSTOMER MODIFICATIONS

There may be customer modifications to the operating system and product set. As noted in section 1.2.3, the decks to make these modifications may be punched from SYSDECK by defining the USERMODS parameter in the PNCHDKS input. These decks assume that customer modifications are in the file USERMODPL in UPDATE format with the master control character set to /. The USERMOD job creates this file.

Jobs punched with USERMODS defined should also have EXTEND defined. The assumption is that PSR modifications are applied to produce new release materials (if needed) and then the xxxUSR jobs are run to update existing permanent file libraries using EXTEND. If customer modifications are to be made, do not run the PURGSCR job until after the xxxUSR jobs are complete. Executing the jobs in this way produces updated PSR summary level deadstart tape and a customer modified tape. The release tape may be useful to determine the cause of either a CDC software or customer modification bug.

An alternate way to apply customer modifications is to run each xxxUSR job immediately after running the corresponding xxxMOD, xxxGEN, or xxxINS job. The binaries produced, including the deadstart tape are the modified versions. However, since this procedure does not produce a production release deadstart tape, this procedure is not recommended.

Another SYSDECK option to consider is $DEFINE NONEWPL. Customer modifications may be very small and may not warrant separate program libraries in addition to the release program library.

1.5 CREATE BACK-UP MATERIALS

When the release system is generated, it is recommended that additional copies of tapes, permanent files, and card decks be made for back-up purposes. Several SYSDECK jobs are available to create these materials. They are described in sections 1.5.1 through 1.5.3.

1.5.1 TAPES

xxxCPY jobs copy release materials. MDTCPY copies MODTAPE. All of these jobs create correctly labeled and formatted tapes. The PRDCPY job copies any permanent file DUMP/LOADPF tape. The SDTCPY job copies 7611-1 I/O Station deadstart tapes. The DSLCPY job copies the 7611-2 Magnetic Tape Station deadstart tape and the SCOPE 2.0 deadstart tape.

1.5.2 PERMANENT FILES

There is a set of xxxPL jobs that catalog release program libraries as permanent files. The following jobs create DUMPF back-up tapes for permanent files with the specified ID.

ID	Job Name
PRDLIB	DPIDPRD
S20CPLOPL	DPCOPLS
S20OPSOPL	DPOSOPL
S20OPSMOD	DPOSMOD
TOOLS	DPTOOLS
(ALL)	DUMPF

1.5.3 CARD DECKS

The DSBCPY job punches copies of the SCOPE 2.0 deadstart binary from a deadstart tape. The STADCK and MTSDCK jobs assemble and punch deadstart cards and decks for the 7611-1 and 7611-2 stations respectively. The STADMP job assembles and punches the 7611-1 dump deck. The P80CPY job copies any card deck including the binary of SMMB.

1.6 CREATE MAINTENANCE INFORMATION

After the system is installed and back-up materials are created, there are two more steps: verify installation and generate maintenance listings.

1.6.1 VERIFY INSTALLATION

There is a set of jobs on SYSDECK named xxxVFY. These jobs are to be run against the newly created system to verify that the installation is successfully completed. Use the following steps.

1. Deadstart the new system using the new deadstart tape and new deadstart deck. The most efficient way is to do an initial deadstart specifying that permanent files be recovered (part II, section 6.2).

2. If permanent files were lost, load the PRDLIB tape.

3. Run the SYSLIB job. This job specifies the files from which to satisfy the externals of compiler generated code (part IV, sections 3.6.2 and 3.6.3).

4. Run the SIFACCT job to retrieve the SIF file and establish the master SIF file. This job works under almost any situation and may be used as a model for a similar installation job.

5. Run the verification jobs. These are mostly self-checking.

1.6.2 GENERATE MAINTENANCE LISTINGS

There are several SYSDECK jobs and programs that produce listings containing information useful in the understanding and maintenance of SCOPE 2.0.

The use of these decks is not required to install the system but, if used, produce useful system maintenance information. Each of the decks require the use of products such as FORTRAN Extended and Sort/Merge which are purchased separately from SCOPE 2.0.

CROSS

CROSS produces a cross reference listing of the SCOPE 2.0 system. As punched from SYSDECK it produces two copies of the listing that contains two parts. The first part lists system symbols from OST, overlay names, field names, macros, and OPDEFs in alphabetical order with the references to each one by overlay (for example, T.XYZ is referenced by overlays OE.A, OU.B, OS.C). The second part is a list of overlays and the symbols, overlay names, field names, macros, and OPDEFs referenced by that overlay.

The procedure used to create this listing is:

1. Apply temporary modifications to COMPASS so that COMPASS outputs a file called MACROS that contains the raw data.

2. Compile and execute a FORTRAN Extended program (PRESORT) to convert the raw data into formatted display code records.

3. Sort the records by item type, name, and then overlay name.

4. Compile and execute a FORTRAN Extended program (POSTSRT) to list the records in a formatted report.

5. Resort the records by overlay name.

6. Execute POSTSRT to list the records in a formatted report. The POSTSRT job reads a card from the INPUT file which indicates the style of the sorted input (0 - item by ident, I - ident by item) and a title for page headings.

Source Listings

To obtain source listings of all program libraries, punch the xxxGEN jobs with NONEWPL and LIST defined but without MODTAPEMD or USERMODS defined. Remove CATALOG statements if the product is not to be installed while producing listings.

The exception is SCOPE 2.0 which is described as follows.

SCOPE 2.0 Listings

Part of the information on SYSDECK is a SCOPE 2.0 routine data base with each entry containing the following:

- Overlay or service routine name
- Overlay index
  - Level
  - Overlay number in level
  - Entry point number
- Listing volume number
- UPDATE deck name
- Miscellaneous flags
- Descriptive comments

There is at least one entry in the data base for each UPDATE deck, each overlay name or entry point, and utility program.

There are several programs and decks that list and manipulate the data base. These jobs are DBLIST, OVLVOL, TXTVOL, UTLVOL, and FTNVOL.

DBLIST

This job executes a program that updates the data base (MERGE is the program name), sorts it (SORTMRG), and lists it by overlay number, by UPDATE deck name, by volume number, and by indexes (REPORT is the program name).

It produces two copies of the output. The SCOPE 2.0 program library must be cataloged by the SCPPL job to execute this program.

OVLVOL

This job uses REPORT to selectively generate *COMPILE cards on a file called UPIN according to an input volume number. The job updates V2OPL using the *COMPILE cards to produce a listing of the overlays in the volume. The volume number appears on the first two columns of the input card. An input card for each appropriate volume is included. The job should be run once for each data card in the deck. At the beginning of each volume is a report listing the overlays in the volume.

The format of the data base cards and the correction cards is discussed in the documentation of the UPDATE program MERGE. The suggested procedure is to run the program to get listings of the release system, punch corrective cards to update the data base according to any customer modifications, and rerun the job to produce a corrected data base.

TXTVOL

This job updates V2OPL and lists OST, V2TEXT, the table description common decks, and the deadstart text partition AUTO.

UTLVOL

This job duplicates the function of OVLVOL for utility code assembled with V2TEXT.

FTNVOL

This job duplicates the function of OVLVOL for FORTRAN Extended code.

1.7 SYSDECK WORKSHEET

Run the MODCAT job (part III, section 1.1) to print the worksheets.

SYSTEM MAINTENANCE MONITOR 3.0 (SMMB) 2

Information concerning the modification of SMMB is available through the Customer Engineering Division.

3.1 REQUIREMENTS

The following materials are necessary to build the 7611-1 I/O Station.

3.1.1 TAPES

STAOPL	Described in part I, section 2.3.

3.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (STAMOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (STAUSR job)

3.1.3 DECKS

STAGEN	Regenerates a STALIB tape from STAOPL
STAMOD	Creates a new STAOPL from the old STAOPL and MODPL, then generates a new STALIB tape when assembled.
STAUSR	Creates a new STAOPL from the old STAOPL and USERMODPL, then generates a new STALIB tape when assembled.
STADCK	Assembles and punches the 7611-1 deadstart decks STATDS, STADDS, and STACLR from STAOPL
STADMP	Assmebles and punches the 7611-1 deadstart dump deck from STAOPL

3.1.4 DEPENDENCIES

None.

3.2 CONFIGURATION PARAMETERS

None.

3.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

3.3.1 STADMP

JOB DECK STADMP

```
STADMP,CP70,T100,MT01.
COMMENT. *SVUV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   NONEWPL HAS BEEN DEFINED                               *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * THIS JOB WILL PUNCH THE STATION DUMP DECK. AN EASIER WAY TO DO  *
COMMENT. * THIS IF THE 7611-1 STATION IS UP, IS TO PUNCH THE DECK FROM THE *
COMMENT. * STATION DISK. THIS CAN BE DONE BY TYPING (IN STATION MODE) .... *
COMMENT. *     PUNCH DDUMP1 P80B  (CR)                              *
COMMENT. *          (FOR PRINTER1 (PPU1) LOADER BOOTSTRAP CARD)     *
COMMENT. *     PUNCH DDUMP1 P80B  (CR)                              *
COMMENT. *          (FOR PRINTER2 (PPU2) LOADER BOOTSTRAP CARD)     *
COMMENT. *     PUNCH DUPRINT P80B (CR)                              *
COMMENT. *          (FOR THE ACTUAL DUMP PROGRAM)                   *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(STAOPL,L=$STAOPL*SCP2.0$)
STAGE(STAOPL,MT,HY,E,PRF,
VSN=CY51)
UPDATE(P=STAOPL,Q)
COMPASS(T)
DISPOSE(LGO,P8)                PUNCH 80 COLUMN BINARY
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
*ID DUMP
*/ PRINTER1 (PP1)BOOTSTRAP CARD
*C DDUMP1
*/ PRINTER2 (PP2) BOOTSTRAP CARD
```

```
JOB DECK  STADMP

*C DDUMP2
*/ DUMP PROGRAM
*C DDPRINT
---EOI--- 6/7/8/9
```

3.3.2 STAMOD

```
JOB DECK  STAMOD


STAMOD,CP70,T200,MT01.
COMMENT. *SVDV,07155,3663D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGF HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                   SCOPE 2.0 INTEGRATION ARH248           *
COMMENT. *                   4201 LEXINGTON AVE. N.                 *
COMMENT. *                   ST PAUL, MINN.    55112                *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
LABEL(STAOPL,L=$STAOPL*SCP2.C$)
STAGF(STAOPL,MT,HY,E,PRF,
VSN=CY47)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                       *
COMMENT. *                                                          *
COMMENT. ************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,O,D,B,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE 7611-1 OLDPL, CREATE SEQUENTIAL NEWPL              *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P=STAOPL,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(STAOPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL. *
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN  *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).  *
COMMENT. *                                                          *
```

```
JOB DECK  STAMOD

COMMENT. ********************************************************************
UPDATE(P=RANDPL,F,B,C=0,N,I=NULL)
RETURN(RANDPL)
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. * UPDATE NEWPL INSERTING CURRENT CYCLE INDICATOR                   *
COMMENT. * AND PRODUCE COMPILE FILE                                         *
COMMENT. *                                                                  *
COMMENT. ********************************************************************
UPDATE(P=NEWPL,F,C)
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. * CREATE NEW STALIB TAPE (7611-1 DEADSTART TAPE)                   *
COMMENT. *                                                                  *
COMMENT. ********************************************************************
FILE(STALIB,RT=S)
STAGE(STALIB,MY,HY,POST,
VSN=CY51)
COMPASS(I,B=STALIB,L=LIST)
UNLOAD(STALIB)
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. * CREATE NEW STAOPL TAPE                                           *
COMMENT. *                                                                  *
COMMENT. ********************************************************************
LABEL(NEWSTA,L=$STAOPL*SCP2.0$)
STAGE(NEWSTA,MT,HY,N,POST,
VSN=CY51)
REWIND(NEWPL,NEWSTA)
COPYP(NEWPL,NEWSTA)
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. * TEST NEW TAPE VIA UPDATE                                         *
COMMENT. *                                                                  *
COMMENT. ********************************************************************
UNLOAD(NEWSTA)
STAGE(NEWSTA,MT,HY,E,PRE,
VSN=CY51)
UPDATE(P=NEWSTA,N=SCRAT,W,C=0,I=NULL)
UNLOAD(NEWSTA,SCRAT)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C STAPSP
---EOS--- 7/8/9
*ID CYCLENO
*D CYC51.1        CYCLE INDICATOR
          DATA    1R5,1R1      CYCLE 51


---EOI--- 6/7/8/9
```

3.3.3 STADCK

```
JOB DECK  STADCK


STADCK,T100,CP70,MT01.
COMMENT.
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED                *
COMMENT. *                                                                  *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA          *
COMMENT. *                                                                  *
COMMENT. *   FOR DSD SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)             *
COMMENT. *                                                                  *
COMMENT. *   NONEWPL HAS BEEN DEFINED                                       *
COMMENT. *   STAGE HAS BEEN DEFINED                                         *
COMMENT. *                                                                  *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO         *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248                  *
COMMENT. *                    4201 LEXINGTON AVE. N.                        *
COMMENT. *                    ST PAUL, MINN.    55112                       *
COMMENT. *                                                                  *
COMMENT. *            ( NO PSOS )                                           *
COMMENT. ********************************************************************
COMMENT.
COMMENT. * THIS JOB WILL PUNCH THE 7611-1 DISK OS AND TAPE OS BINARY CARDS.  *
COMMENT. * THIS JOB ALSO PUNCHES STACLC - THE CHANNEL CLEAR CARD.  THESE     *
COMMENT. * CARDS WILL BE PUNCHED IN ORDER. AN EASIER WAY TO DO THIS IF       *
COMMENT. * THE 7611-1 STATION IS UP, IS TO PUNCH THE CARDS FROM THE STATION  *
COMMENT. * DISK. THIS CAN BE DONE BY TYPING (IN STATION MODE) ....           *
COMMENT. *      PUNCH DSC B80B  (CR)                                        *
COMMENT. *           (FOR THE DISK DEADSTART CARD)                          *
COMMENT. *      PUNCH DSCT B80B  (CR)                                       *
COMMENT. *           (FOR THE TAPE DEADSTART CARD                           *
COMMENT. *      PUNCH CLC B80B (CR)                                         *
COMMENT. *           (FOR CHANNEL CLEAR CARD)                               *
COMMENT. *                                                                  *
COMMENT. ********************************************************************
LABEL(STAOPL,L=*STAOPL*SCP2.0*)
STAGE(STAOPL,MT,HY,E,ORF,
VSN=CY51)
REWIND(STAOPL)
UPDATE(P=STAOPL,O)
COMPASS(I)
DISPOSE(LGO,PB)                 PUNCH 80 COLUMN BINARY
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE Y.GO
---EOS--- 7/8/9
*ID STAT
*/ DISK OS BINARY CARD
*C DSC
*/ TAPE OS BINARY CARD
*C SCPC
*/ CHANNEL CLEAR CARD

*C CLC
---EOI--- 6/7/8/9
```

4.1 REQUIREMENTS

The following materials are necessary to build the 7611-2 Magnetic Tape Station.

4.1.1 TAPES

MTSOPL	Described in part I, section 2.4

4.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (MTSMOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (MTSUSR job)

4.1.3 DECKS

MTSGEN	Regenerates a MTSLIB tape from MTSOPL
MTSMOD	Creates a new MTSOPL from the old MTSOPL and MODPL, then generates a new MTSLIB tape when assembled
MTSUSR	Creates a new MTSOPL from the old MTSOPL and USERMODPL, then generates a new MTSLIB tape when assembled
MTSDCK	Assembles and punches the 7611-2 deadstart decks MTSTDS and MTSDDS from MTSOPL

4.1.4 DEPENDENCIES

None.

4.2 CONFIGURATION PARAMETERS

4.2.1 MODIFICATIONS TO MTSLIB AND MTSOPL TAPES

XPP (Multiplexer PPU) Modifications

The XPP program must reside in PPU0.

Parameter	Released Default Value	Significance
CHPAR	0	Parity and deadstart channel number
CHFI	0	FLPP to XPP channel
CHFO	6	XPP to FLPP channel
CHS	4	Channel between XPP and SPP
CHD1	1	Channel between XPP and driver 1
CHD2	2	Channel between XPP and driver 2
CHD3	3	Channel between XPP and driver 3
CHD4	5	Channel between XPP and driver 4

SPP (System PPU) Modifications

Parameter	Released Default Value	Significance
XP	0	XPP channel number
DP1	1	DPP channel number
DP2	2	DPP2 channel number
DP3	3	DPP3 channel number
DP4	5	DPP4 channel number
DA	6	Disk function/status channel
DB	7	Disk data channel
S	4	Display channel
PSTAT	see significance	DPP status table

Released DPP Table	Driver
DH=0,DSU=2,DOF=0	1
DH=0,DSU=1,DOF=0	2
DH=0,DSU=4,DOF=1	3
DH=0,DSU=3,DOF=1	4

For each DPP driver to be modified:

1. Set DH to 1 if driver PPU is disconnected; set to 0 if connected.
2. Set DSU to PPU number (1 to 4) that is connected to other half of 2 x 8 controller (a 2 x 8 controller is connected to two PPUs); set to 0 if 1 x 8 controller.
3. Set DOF to 1 if DPP is OFF; set to 0 if ON.

Parameter	Released Default Value	Significance
USTAT	see significance	Unit status table

Released USTAT Table	Unit
UNT=0, UOF=0, U2C=1, UDC=1	0
UNT=0, UOF=0, U2C=1, UDC=1	1
UNT=1, UOF=0, U2C=1, UDC=1	2
UNT=1, UOF=0, U2C=1, UDC=1	3
UNT=0, UOF=0, U2C=1, UDC=1	4
UNT=1, UOF=0, U2C=1, UDC=1	5
UNT=0, UOF=0, U2C=1, UDC=1	6
UNT=1, UOF=0, U2C=1, UDC=1	7
UNT=0, UOF=0, U2C=1, UDC=3	10
UNT=0, UOF=0, U2C=1, UDC=3	11
UNT=0, UOF=0, U2C=1, UDC=3	12
UNT=1, UOF=0, U2C=1, UDC=3	13
UNT=1, UOF=0, U2C=1, UDC=3	14
UNT=1, UOF=0, U2C=1, UDC=3	15
UNT=0, UOF=0, U2C=1, UDC=3	16
UNT=1, UOF=0, U2C=1, UDC=3	17

For each unit to be modified:

1. Set UNT to 1 for 9-track; set to 0 for 7-track.
2. Set UOF to 1 if unit is OFF; set to 0 if ON.
3. Set U2C to 1 if unit is a 2 x 8 controller; set to 0 if 1 x 8 controller.
4. Set UDC to DPP number (1 to 4) that is connected to unit; if unit is connected to two DPPs, set to lower DPP number.

CHCH — see significance — PPU channel table

Released CHCH Table		PPU
CON	XPMD4+1	0
CON	DP1-XP	1
CON	DP2-DP1	2
CON	DP3-DP2	3
CON	DP4-DP3	5

NOTE

The DPPs that can be connected to a 2 x 8 controller are DPP1 and DPP2 or DPP3 and DPP4.

DPP (Driver PPU) Modifications

Parameter	Released Default Value	Significance
CHF	6	PPU to controller function/status channel; see CHD
CHD	7	PPU to controller data channel.
		The CHF and CHD may be changed for each of the following overlay routines.
		BLB W7U LBP W7L LDC W9U R7U W9L R7L RER R9U WER R9L POS
		If fewer than four driver PPUs are to be used, assemble all overlays and the resident programs for the drivers to be used. For example, if only driver 4 is to be used, assemble only DP4 and all of the overlays; do not assemble DP1, DP2, or DP3.
READ error recovery parameters:		
LIM	6	Number of single backspace attempts for read error recovery
LIMF	6	Number of read recovery attempts without operator intervention
WRITE error recovery parameter:		
LIM	6	Number of write recovery attempts without operator intervention

7611-2 Dump Program Modifications

Parameter	Released Default Value	Significance
CHXPP	3	Data subchannel to XPP
CHDS	2	Deadstart channel
CHFLPO	6	XPP to FLPP data subchannel
CHDSS	0	Station deadstart line
CHDD	7	Dead dump channel

4.2.2 MODIFICATIONS TO MTSDDS AND MTSTDS DECKS

The following modifications apply to both MTSDDS and MTSTDS.

Parameter	Released Default Value	Significance
FLDS	2	FLPP deadstart channel
XPP	3	FLPP to PPU0 channel
DP1	1	Channel to DPP1
DP2	2	Channel to DPP2
DP3	3	Channel to DPP3
DP4	5	Channel to DPP4
SP	4	Channel to SPP
DD	7	PPU0 dead dump channel
DS	0	PPU0 deadstart channel
XP	0	Channel to XPP
FL	0	FLPP to XPP channel
FLO	6	XPP to FLPP channel
CH	77B	Bit configuration of station PPUs; 12-bit word
CHF	6	DDP1 tape function channel
CHD	7	DPP1 tape data channel
DA	6	SPP disk function/status channel
DB	7	SPP disk data channel
S	4	SPP 522 display channel
RBS	DP1	SPP library binary input channel (DP1 is released as 1)
DPN	see significance	Table indicating which driver is reading the STALIB tape

Released DPN Table			PPU	Significance
DPN	CON	xxxx	1	Tape deadstart xxxx=DTDS (in TDS) or INMC
	CON	INMC	2	Not used
	CON	INMC	3	Not used
	CON	SDS	4	SPP deadstart
	CON	INMC	5	Not used

4.2.3 CROSS REFERENCES

If DP1, DP2, DP3, and DP4 are to be modified in SPP, change the following:

1. CHD1, CHD2, CHD3, and CHD4 in XPP
2. CHCH and PSTAT in SPP

If fewer than four drivers are in the configuration, change the following:

1. DP1, DP2, DP3, DP4, CHCK, PSTSTA, and USTAT in SPP

2. CHD1, CHD2, CHD3, and CHD4 in XPP

3. DP1, DP2, DP3, DP4, and DPN in MTSDDS and MTSTDS

If the channel connects between the DPPs and the controllers are to be modified, change the following:

1. CHF and CHD in DPP overlays

2. CHF and CHD in MTSTDS and MTSDDS

3. CHF and CHD in MTSCPU (in SCOPE 2.0 FLPP binary deck)

4.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

4.3.1 MSTMOD

```
JOB DECK  MTSMOD


MTSMOD,CP/u,I1090,MT01.
COMMENT. *SVOV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
LABEL(MTSOPL,L=$MTSOPL*SCP2.0$)
STAGE(MTSOPL,MT,HY,c,PRF,
VSN=CY47)
COMMENT. ***************************************************************
COMMENT. *                                                             *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                         *
COMMENT. *                                                             *
COMMENT. ***************************************************************
ATTACH(MODPL,MODPL,ID=52,OPSMOD,
CY=51)
UPDATE(P=MODPL,O,D,8,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ***************************************************************
COMMENT. *                                                             *
COMMENT. * UPDATE MTSOPL,CREATE SEQUENTIAL NEWPL                       *
COMMENT. *                                                             *
COMMENT. ***************************************************************
UPDATE(P=MTSOPL,F,C=S,N=RANDPL,I=UPIN)
UNLOAD(MTSOPL)
COMMENT. ***************************************************************
COMMENT. *                                                             *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.*
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN    *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).    *
COMMENT. *                                                             *
```

```
JOB DECK  MTSMOD

COMMENT. ************************************************************************
UPDATE(P=RANDPL,F,B,C=0,N,I=NULL)
RETURN(RANDPL)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * UPDATE NEWPL INSERTING CURRENT CYCLE INDICATOR                       *
COMMENT. * AND PRODUCE COMPILE FILE                                            *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
UPDATE(P=NEWPL,C,F)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * PRODUCE MTS BINARIES                                                *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
SKIPF(COMPILE)
COMPASS(I,B=BINARY,L=LIST)
SKIPF(COMPILE)
COMPASS(I,B=BINARY,L=LIST)
UNLOAD(COMPILE)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE NEW MTSLIB (MTS US TAPE)                                     *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
STAGE(MTSLIB,MT,HI,POST,
VSN=CY51)
REWIND(BINARY,MTSLIB)
COPY(BINARY,MTSLIB)
UNLOAD(MTSLIB)
LABEL(NEWMTS,L=$MTSOPL*SCP2.0$)
STAGE(NEWMTS,MT,HY,N,POST,
VSN=CY51)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE NEW MTSOPL TAPE                                              *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
REWIND(NEWPL,NEWMTS)
COPY(NEWPL,NEWMTS)
UNLOAD(NEWMTS)
STAGE(NEWMTS,MT,HY,E,PRE,
VSN=CY51)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * TEST NEW TAPE VIA UPDATE                                            *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
UPDATE(P=NEWMTS,N=SCRAT,W,C=0,I=NULL)
```

```
JOB DECK  MTSMOD


UNLOAD(NEWMTS)
UNLOAD(NEWMTS,SCRAT)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C MTSPSR
---EOS--- 7/8/9
*ID CYCLENO
*D MT0026.1
          DATA   1R5,1R1      CYCLE 51
---EOI--- 6/7/8/9
```

4.3.2 MTSDCK

```
JOB DECK  MTSDCK


MTSDCK,CP70,T100,MT01.
COMMENT.
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. *    THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED               *
COMMENT. *                                                                  *
COMMENT. *    BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA          *
COMMENT. *                                                                  *
COMMENT. *    FOR PSR SUMMARY LEVEL 61 (SEE SYSDECK DESCRIPTION)            *
COMMENT. *                                                                  *
COMMENT. *    NONEWPL HAS BEEN DEFINED                                      *
COMMENT. *    STAGE HAS BEEN DEFINED                                        *
COMMENT. *                                                                  *
COMMENT. *    PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO        *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248                  *
COMMENT. *                    4201 LEXINGTON AVE. N.                        *
COMMENT. *                    ST PAUL, MINN.    55112                       *
COMMENT. *                                                                  *
COMMENT. *            ( NO PSRS )                                           *
COMMENT. ********************************************************************
COMMENT.
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * DECK TO PUNCH MTSTDS AND MTSDDS BINARY DECKS                         *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
LABEL(NEWSCP,L=$SCPOPL2.0*)
STAGE(NEWSCP,MT,HY,E,PRE,
VSN=CY51)
UPDATE(P=NEWSCP,O)
COMPASS(I,P=PUN,L=LIST)
REWIND(PUN)
COPY(PUN,PUNCHB)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
*ID MTS
*/ TAPE DS DECK --MTSTDS--
*D TDS
*/ DISK DS DECK --MTSDDS--
*D EDS
---EOS--- 7/8/9
*/ BELOW ARE CARDS FOR ADDITIONAL COPIES OF BINARY DS DECKS
REWIND(PUN)
COPY(PUN,PUNCHB)
REWIND(PUN)
COPY(PUN,PUNCHB)
---EOI--- 6/7/8/9
```

6000 OR CYBER 70 SERIES SCOPE 3.4 STATION 5

All modifications to SCOPE 3.4 are described in the 6000 SCOPE 3.4 Installation Handbook, Pub. No. 60307400, except for the following station parameter.

Parameter	Released Default Value	Significance
CNTMLO	512_{10}	The number of loops occuring in the CPL program (station FLPP) before the station logs out. Present setting of 512 causes 512 loops in CPL which is about one minute. Equation is: Value=(8.53)(wait time in seconds) 512=(8.53) (60 seconds)

6.1 REQUIREMENTS

The following materials are necessary to build the SCOPE 2.0 operating system.

6.1.1 TAPES

SCPOPL	Described in part I, section 2.6.

6.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (SCPMOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (SCPUSR job)
FTNIOLIB (ID=PRDLIB)	Necessary to satisfy FTN generated externals in LIBEDT and ANALYZE
PRDLIB (ID=PRDLIB)	Necessary to add product set binary to deadstart tape and provide the FTN compiler for building LIBEDT and ANALYZE

6.1.3 DECKS

SCPGEN	Regenerates SCPLIB tape and SCPSID binary deck from SCPOPL; catalogs SYSLIB (ID=PRDLIB)
SCPMOD	Creates a new SCPOPL from the old SCPOPL and MODPL, then generates a new SCPLIB tape and a SCPSID binary deck from the new SCPOPL; catalogs SYSLIB (ID=PRDLIB)
SCPUSR	Creates a new SCPOPL from the old SCPOPL and USERMODPL, then generates a new SCPLIB tape and a SCPSID binary deck from the new SCPOPL; catalogs SYSLIB (ID=PRDLIB)
SCPINS	Replaces product set only; punches SCPSID and catalogs SYSLIB (ID=PRDLIB)
DSBCPY	Punches SCOPE 2.0 SCPSID binary deck from a SCPLIB
DBUGINS	Catalogs binary from SCPOPL as DBUGLIB (ID=PRDLIB)

6.1.4 DEPENDENCIES

FTN and FCL must be installed before building SCOPE 2.0; or at least PRDLIB must contain a FTN compiler updated to the current PSR summary level and FTNIOLIB must be updated to the current PSR summary level.

6.2 CONFIGURATION PARAMETERS

Parameters for configuring the SCOPE 2.0 system are defined in decks IPARAMS in OST and ORL. These decks are on the SCPOPL tape. Unless specified otherwise, the parameters described are in the IPARAMS common deck, which can be listed by assembling the deck OST (system text) using a *COMPILE OST UPDATE directive.

The cross references are the routines affected by the parameter.

6.2.1 DEBUGGING

None that can be changed by the user.

6.2.2 LIBRARIES

None that can be changed by the user.

6.2.3 SYSTEM CONTROL

Parameter	Released Default Value	Significance
IP.DRMGT	1	Assemble static resource management (on-line tape) accounting code which is the tape I/O message TAPE, SECONDS in the accounting summary. 0 No dayfile message 1 Message dayfile

The scheduler uses the following parameters to determine time slices and to select the next job for execution. Factors important to selecting a job include: job status (X1, X2, X3, W1, W2, W3), resource utilization (SCM, LCM, mass storage, on-line tapes), and external priority.

IP.LCMW	1	Weight of LCM for computing scheduling value when job is in X1 or X2 status†. Value=(LCM allocated/1000B*2**IP.LCMW)
IP.IOWT	2000B	Weight of I/O rate in scheduling value when job is in X1 or X2 status†. Value= $\left[(1/2 \frac{\text{CPU time allocated}}{\text{CPU time used}})\right]$ *IP.IOWT

†When a job is in X1 or X2 status, the scheduling value is determined by the LCM value plus the I/O value plus the external priority specified on the job card.

Parameter	Released Default Value	Significance
IP.AUWT	1000B	When a job is rolled out to disk (X3 status†), value is increased by au*IP.AUWT where au is allocatable devices such as tapes.
IP.MINT	3	Minimum job time slice in clock cycles (3.6 milliseconds); refer to IP.TSWT for SCM execution.
IP.MAXT	20D	Maximum job time slice in clock cycles (3.6 milliseconds); refer to IP.TSWT.
IP.TSWT	1	A job's time slice is the amount of time the job is assigned the CPU. At the completion of a time slice, the scheduler interrupts and assigns a new time slice to another job or to the same job. The size of the time slice is expressed in clock cycles (3.6 milliseconds). time slice = (priority+777B/1000B*2**IP.TSWT) The range is further restricted by: IP.MINT<time slice<IP.MAXT If IP.MINT is the same as IP.MAXT, all time slices are equal.
IP.SCVI	4000B	Interval in clock cycles at which SCHL (the scheduler) ages jobs in X2 and X3 status.
IP.LCMI	510B	Interval in clock cycles at which an X2 job is forced from LCM into SCM.
IP.ROLI	12000B	Interval in clock cycles at which SCHL attempts to swap an X3 job into LCM from disk.

6.2.4 JOB MANAGEMENT

Parameter	Released Default Value	Significance
IP.DJMGT	1	Determines whether job management accounting code is to be assembled. 0 Code not assembled 1 Code assembled

†When a job is in X3 status, the scheduling value is determined by the au value plus the external priority specified on the job card.

Parameter	Released Default Value	Significance
IP.MAXBJ	420B	Maximum number of I/O buffers for a job. When the maximum number of buffers is exceeded, the job is unconditionally aborted. This value is reduced by deadstart based on the number of I/O buffers physically available. Refer to T.MAXBUF in part III, section 6.2.13.
IP.MSCT	1000D	Maximum number of user dayfile messages for a job. When this parameter is exceeded, the job is aborted, but may be reprieved. Cross references: OS.CCMM, OS.STYY
IP.SCM	1000B	SCM field length to which the user field length is reduced between job statements. Cross references: OE.INJ, OE.SAM, OS.EXP, OS.TRJ2, OU.NOUN
IP.MPR	7000B	Maximum external priority. If the Pnnn parameter on the job card exceeds this value, priority is reduced to IP.MPR. Cross references: OE.ACJ2, OE.AGE, OE.INJ
IP.PRTY	1000B	Default external priority, if not specified on the job card. Cross references: OE.ACJ2, OE.FINP
IP.TIL	10B	Job default time limit in seconds; if not specified on the job card. Cross reference: OE.ACJ2
IP.MWJ	24000B	Default mass storage limit x ($10,000_8$) in words. The maximum amount of mass storage space that can be allocated to a job specified in units of 4096 words. The job aborts if mass storage limit is exceeded. This limit may be changed with the LIMIT statement. Cross reference: OE.INJ

Parameter	Released Default Value	Significance
IP.TPD	3	If tape density is defaulted on a REQUEST card, it is set to IP.TPD; possible values are: 1 200 bpi 2 556 bpi 3 800 bpi 4 1600 bpi Cross reference: OU.RQM
IP.STG	1	If tape staging direction is defaulted on a STAGE card, it is defaulted to IP.STG; possible values are: 2 Stage out 1 Stage in Cross reference: OS.RQP

6.2.5 DEVICE MANAGEMENT

Parameter	Released Default Value	Significance
IP.RTC	10D	Retry count for tape parity errors on on-line tapes. A read or write operation on tape is tried IP.RTC times before the parity error is declared unrecoverable. Cross references: OE.TQI, OE.TQO

6.2.6 PERMANENT FILES

Parameter	Released Default Value	Significance
IP.DP	4	Defines the number of PFD pages as a power of two. In other words, the number of pages is 2**IP.DP. It is used in hashing the permanent file name. Cross references: AUDIT, DUMPF, LOADPF, OU.ATT1, OU.CG1
IP.OVP	4	Total number of PFD overflow pages; each PFD overflow page represents a 512-word block on the disk.

Parameter	Released Default Value	Significance
		The size of the overflow area is a function of how evenly the permanent file name hash is distributed. It is recommended that the number of overflow pages be at least one quarter of the number of PFD pages. Cross references: OS.CG3, OS.CG4, OU.ATT1
IP.PFRP	1	Default retention period (in days) for permanent files; values may be from 0 to 999. Value 999 is interpreted as indefinite retention. When modifying this parameter, consider the amount of mass storage available. Cross references: LOADPF, OS.CPC1
IP.PPP	0	Determines the type of privacy procedure that is used. 0 Standard privacy procedure is used 1 Installation privacy procedure in overlay OS.IPPP is called via a GOTO. OS.IPPP checks file privacy and must be added to the system. Input to OS.IPPP can be obtained through calling the common deck named PFPARAM. Location PFN is the beginning of the file definition block (FDB). On returning, OS.IPPP must set the permission bits right justified in register B3. OS.IPP is called when an ATTACH or new cycle CATALOG occurs and must do its own password checking. The PP parameter is available to OS.IPPP in the FDB parameter area. Cross reference: OSPWCK
IP.CY	3	Average number of cycles per unique permanent file name; used to determine the amount of disk space reserved for PFC during deadstart initialization. Value may be from 1 to 5 but in most cases should be 3 or greater. Cross reference: OST
IP.UVPM	17B	Determines which permissions are granted when the universal password is submitted. Submitting the universal password may cause either one or a combination of permissions to be granted.

Parameter	Released Default Value	Significance
		UVPM Bit Position / Permission 1 Read 2 Extend 4 Modify 10B Control 17B All The released system defines the universal password as QZUYVXW. It may be modified by reassembling the deck OSPWCK using the following UPDATE directive. *DELETE OSPWCK.167 PASSWD DIS,*UNIVERSAL* *COMPILE OSPWCK Cross reference: OS.PWCK
L.APF	512D	Length of the attach permanent file table (APF); determines the number of permanent files that may be attached at a given time. The permanent file manager uses the first two APF words for communication information. The remainder of the APF contains one word entries. Every permanent file attached has an APF entry. There are no limitations on the size of the table. Cross references: ORL, OS.CAPF, OS.SAPF, OS.UCL, AUDIT, LOADPF, ODS
L.UD	0	Number of words reserved by the installation in the PFD entry. This space begins at word LE.PFD of each entry. Cross references: AUDIT, LOADPF, DUMPF, OSCG4, OS.CPFC, OU.EXD1, OS.PFSR, OS.DPF, ODS, OV.FPDP
L.UC	0	Number of words reserved by the installation in the PFC entry. This space begins at word LE.PFC of each entry. Cross references: AUDIT, LOADPF, DUMPF, OS.EXD2, ODS, OU.FPDP

6.2.7 LOADER

Parameter	Released Default Value	Significance
IP.MAP	2	Map options Bit Positions — Significance 0 — No map 1 — S type map 2 — B type map or partial map 4 — E type map 10B — X type map or map is on Cross reference: OS.COG
IP.LDPRS	0	Loader option to preset core image before loading 0 — No presetting 1 — Zeros 2 — Ones 3 — Indefinite 4 — Infinite 5 — Negative indefinite 6 — Negative infinite 7 — Alternating zeros (2525252525) 10B — Alternating ones (5252525252) Cross references: OS.COG, OS.CAL4, OS.EXP
IP.LDRWD	1	Option for rewinding of load files 0 — No rewind 1 — Rewind Cross reference: OS.COG
IP.LDER	2	Set level of errors at which loader aborts 1 — All 2 — Fatal 3 — None Cross references: OS.COG, OS.EXP

6.2.8 INFORMATION ROUTING

Parameter	Released Default Value	Significance
IP.FCAUS	2	File copies default input allocation unit size for unit record input, 6000, attached permanent file, and staged tape input.

Parameter	Released Default Value	Significance

Value	Sectors Per Allocation Unit	Longest File Length
1	5 or 1/8 track	11,703,000B words
2	12_8 or 1/4 track	23,607,000B words
3	24_8 or 1/2 track	47,417,000B words
4	50_8 or 1/1 track	116,037,000B words
5	120_8 or 2-tracks	235,077,000B words

Cross references: OE.SFBH, OS.FRBF

IP.SFTIJ — 100B

Indicates the maximum number of SFT entries that may be allocated to input jobs at one time. All input SFT entry requests attempting to exceed this maximum are rejected.

Cross references: OE.SSOM, ORL

IP.SPFTH — 1

The number of SFT events that occur before the SFT is written to disk for recovery. An SFT entry is defined as one of the following:

- Unit record input file suspended (I/O delink received from station)
- Unit record output file SFT entry created
- End of volume on unit record input or output
- Unit record SFT entry is returned to the system

A unit record input file is not submitted to job acceptance, nor is an output file announced to a station until its SFT entry has been written to the disk to be made permanent. Refer to the following example under IP.SPFRQ.

Cross reference: ORL

IP.SPFRQ — 5000D

Indicates the frequency in clock interrupts that SFT activity is monitored. The SFT event count is compared to IP.SPFTH at the interval specified. If the two values differ and a write is not in progress, the SFT is written to disk. If the current event count is identical to IP.SPFTH, nothing occurs until the next interval is reached.

Parameter	Released Default Value	Significance
		For example, if IP.SPFTH were three and IP.SPFRQ were 5000D, after 5000D clock interrupts the SFT would be checked for activity. If during that time only two events occurred, the SFT would be written to disk. Cross references: OE.SPFE, OE.RSFT

6.2.9 SYSTEM STATISTICS

Parameter	Released Default Value	Significance
IP.SIFNB	1	Number of system I/O buffers permanently assigned to the SIF after deadstart is equal to IP.FCLRF+3+IP.SIFNB. Under normal SIF usage, IP.SIFNB should be set to zero. However, when performance measurement is being done or when large quantities of output to the SIF is expected, set IP.SIFNB to higher values. One check on the validity of this value is the SIF reject count supplied by the RTRVSIF operation. If this value is large (consistently over 1), IP.SIFNB should be increased. If it is zero, IP.SIFNB should be decreased. Cross references: OE.SFQM, ORL
IP.SIFSC	100D	The current SIF buffers being filled as well as an updated FAT are written to disk when their number equals IP.SIFSC or IP.FURSI, whichever is less. In the released system IP.FURSI is 5. If IP.SIFSC is set greater than IP.FURSI-1, it has no effect. Setting it to less than IP.FURSI-1 causes the SIF buffers to be written to disk more often. Cross reference: OE.SFBF
IP.QUEA	1	Determines whether system queue is maintained and logged to the system information file. If IP.QUEA=1, information is gathered. If IP.QUEA≠1, no information is gathered.
IP.JACCT	1	Extended accounting 0 Summary entries for tape and mass storage time are not entered in the job dayfile at job termination; they are sent to the SIF. 1 Tape and mass storage time are printed.

Parameter	Released Default Value	Significance
IP.JACTL	0	ACCOUNT statement 0 ACCOUNT statement is not required and may appear anywhere in the job control statements. 1 ACCOUNT statement is required and must appear immediately after the job control statement. The ACCOUNT statement may also appear elsewhere in the job control statements unless the installation supplied code in OS.ACRD flags it as an error.

6.2.10 RECORD MANAGER

The first four record manager parameters control the conditional assembly of accounting code and the output of a particular message in the dayfile accounting summary. If the parameter is set to 0, the accounting code is not assembled and no dayfile message is printed. If the parameter is set to 1, the code is assembled (to update a field in JSLCM, the job supervisor LCM area) and a dayfile message is printed.

Parameter	Released Default Value	Significance
IP.FLA	1	Accounting of number of open/close calls accumulated. The dayfile message is RM771, file open and close requests. The JSLCM field updated is JSOCC.
IP.RMA	1	Accounting of the data transfer calls and number of data words moved. The dayfile message is RMT772, data transfer requests. The JSLCM field updated is JSDTC.
IP.BMA	1	Accounting of the number of buffer manager data transfer calls, manager control or positioning calls, queue manager calls, and recall calls. The dayfile message is RM774, buffer manager data transfer requests. The JSLCM field updated is JSBMC.

Parameter	Released Default Value	Significance
IP.FMA	1	Accounting of the time waiting for mass storage; it is a means to calculate utilization.
IP.LDEN	0	Specifies tape label density for 7-track tapes 0 Same as user file density 1 200 2 556 3 800 } For compatibility with 6000 SCOPE 3.4 and earlier versions

6.2.11 DEADSTART PARAMETERS

Parameter	Released Default Value	Significance
IP.YMD	0	Date format for the date request at deadstart time 0 DATE IS MM/DD/YY 1 DATE IS DD/MM/YY Cross reference: DS2
IP.LIB	220B	Number of library buffers Cross reference: DS1
IP.SYS	40B	Number of system table buffers Cross reference: DS1

6.2.12 TEMPORARY PARAMETERS

Parameter	Released Default Value	Significance
IP.SCMSI	100,000B	SCM size for half size SCM; full size SCM is 200,000B.
IP.LCMSI	764,000B	LCM size for half size LCM; full size LCM is 1,764,000B.

6.2.13 ORL PARAMETERS

The following parameters are in ORL. They can be modified (except for L.ZERO) with STORE statements during deadstart initialization and deadstart recovery. But the values specified during deadstart are only recovered with a deadstart recovery, not with a deadstart initialization.

Parameter	Released Default Value	Significance
T.SPF	IP.SPFTH=1	System permanent file (SFT) recovery; value is in IP.SPFTH (section 6.2.8)
T.MAXBUF	IP.MAXBJ=4208	Maximum number of I/O buffers for a job; value is in IP.MAXBJ (section 6.2.4). If the number of maximum buffers is set too high for the system, deadstart sets T.MAXBUF to the smaller of the two following values. 1. T.MAXBUF as set by a STORE command during deadstart or the default value (a STORE setting overrides the default value). 2. n-(r+s+4) n Total number of system I/O buffers physically available. s Number of buffers reserved for SIF. This value is IP.SFNB+IP.FCLRF+3 (8 in released system). r Number of buffers reserved for SCM image of job to be rolled to disk, the largest integer not greater than $\left\lceil \frac{T.MAXS+ljs+777B}{1000B} \right\rceil$ ljs is length of job supervisor (1200_8 in released system).
T.MAXL	400,000B	Maximum user LCM field length for a half size machine. The value for a full size machine would be 1,400,000B.
T.MAXS	60,000B	Maximum user SCM field length for a half size machine. The value for a full size machine would be 160,000B.
T.LCMCHK	7	The number of LCM words to ensure that the resident loader checks so that no system overlay has written into low LCM. Refer to part IV, section 3.6.6, System Debug.

Parameter	Released Default Value	Significance
T.BUFCHK	0	A flag word indicating whether or not the buffer check code is to be executed. Refer to part IV, section 3.6.6, System Debug for further information. 0 Code not executed 1 Code is executed; system aborts if buffer release problem is detected. CAUTION System performance is degraded by as much as fifteen per cent when BUFCHK is activated.
T.SYSABT	0	Defines the conditions under which a system error results in a system halt (crash). (The SAVE/RESTORE and CALL stacks are preserved on all crashes.) 0 No crash 1 Crash on system EEA exits only. That is: an EEA exit (such as program range error or direct range errors) in the systems interchange, interrupt handlers, executive, or job supervisor results in a crash. 2 Crash on system EEA and error exits. That is: an EEA or error exit (such as via ABORTJ macro) in the systems interchange, interrupt handlers, executive level, or job supervisor results in a crash. There is not a system halt on user program EEA or error exits.

The following parameter is in ORL but can only be modified by reassembling ORL.

L.ZERO	1000B	The first L.ZERO words in LCM are set to zero. (The system has been run with L.ZERO set to 200B and 1000B.)

6.2.14 UPDATE

The following parameters are in *DECK UPDATE.

Parameter	Released Default Value	Significance
DECLKEY	1	Enables DECLARE directive
CHAR 64	1	Supports full 64-character set

Parameter	Released Default Value	Significance
PMODKEY	1	Enables G option and PULLMOD card
AUDITKEY	1	Allows audit functions
EDITKEY	1	Allows merge and edit
OLDPLKEY	1	Enables UPDATE to read both old style and new style old program libraries
EXTOVLP	1	Enables detection of four types of overlap involving two or more cards in a correction set 1. Cards refer to same card 2. Card activates already active card 3. Card deactivates already inactive card 4. Insert refers to inactive card
DYNAMFL	1	Dynamic field length; field length is expanded as necessary.

6.2.15 DSC DECK IN FLPP DECK

The DSC deck is on the SCOPE 2.0 library. It is part of the FLPP deck during installation when the 7611-2 Magnetic Tape Station is part of deadstart. If no parameters are changed, the driver is assumed to be in PPU1.

Parameter	Released Default Value	Significance
FLDS	2	FLPP deadstart channel
CHXP	3	Channel from FLPP to PPU0
CHSCM	1	Channel from FLPP to SCM
CHDS	0	Channel to deadstart all other PPUs in the station
CHDR1	1	Driver←→XPP channel, XPP program
CHFP	6	XPP←→FLPP channel
CHXPP	0	Driver←→XPP channel, driver program
CHF	6	PPU tape controller function/status channel
CHD	7	PPU tape controller data channel

6.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are decks released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

6.3.1 SCPINS

JOB DECK SCPINS

```
SCPINS,CP70,MT01,T1000.
COMMENT. *SVDV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * THIS JOB TAKES AN EXISTING DEADSTART TAPE AND REPLACES THE PRODUCT  *
COMMENT. * SET FROM PRDLIB (CREATED BY NPRDLIB) AND PRODUCES A NEW DEAD START  *
COMMENT. * TAPE                                                    *
COMMENT. *                                                          *
COMMENT. ************************************************************
FILE(DSTAPE,RT=W,BT=C)
STAGE(DSTAPE,MT,HI,PRE,
VSN=CY51)
REWIND(DSTAPE)
COPY(DSTAPE,SCPLIB)
UNLOAD(DSTAPE)
ATTACH(PRDLIB,PRDLIB,ID=PRDLIB)
LIBEDT(M)
FILE(NEWLIB,FO=SQ)
FILE(NDSTAPE,RT=W,BT=C)
STAGE(NDSTAPE,MT,HI,POST,
VSN=CY51)
COPY(NEWLIB,NDSTAPE)
UNLOAD(NDSTAPE)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG SYSLIB (SYSTEM LIBRARY WITHOUT PRODUCT SET)      *
COMMENT. *                                                          *
COMMENT. ************************************************************
EXIT(C)
CATALOG(SYSLIB,SYSLIB,ID=PRDLIB,PW=SCPCN,CN=SCPCN,MD=SCPMD,EX=SCPEX,
```

```
JOB DECK  SCPINS


CY=51)
PAUSE.  MP002 - NEW FILE CATALOGED SUCCESSFULLY..TYPE X.GO
EXIT.
PAUSE.  MP004 - NEW PRDLIB CYCLE EXISTS. TYPE X.GO TO REPLACE
ATTACH(A,SYSLIB,ID=PRDLIB,PW=SCPCN,
CY=51)
PURGE(A)
RETURN(A)
CATALOG(SYSLIB,SYSLIB,ID=PRDLIB,PW=SCPCN,CN=SCPCN,EX=SCPEX,MD=SCPMD,
CY=51)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
LIBRARY(NEWLIB,NEW=2500)
OLDLIB(SCPLIB)
REPLACE(*,PRDLIB,LIB)
FINISH.
LISTLIB(*,NEWLIB,N=1)
*/ CREATE SYSLIB FROM SCPLIB
LIBRARY(SYSLIB,NEW=2500)
OLDLIB(SCPLIB)
*/ DELETE PRODUCT SET - ERROR(SKIP) USED TO PREVENT ABORT BECAUSE OF A DELETE
*/ ON A PRODUCT NOT ON THE ORIGINAL DEADSTART TAPE
ERROR(SKIP)
DELETE(RUN*JOBDIAGF)
DELETE(COMPASS*$COMP1$$$)
DELETE(FTN*$PASS14$$$)
DELETE(COBOL*COBOL51)
DELETE(SORTMRG*SORT20)
DELETE(DCP/MEMC/CT73)
ERROR(ABORT)
FINISH.
LISTLIB(*,SYSLIB,N=1)
---EOI--- 6/7/8/9
```

6.3.2 DBUGINS

JOB DECK DBUGINS

```
DBUGINS,CP70,T100,MT01.
COMMENT. *SVDV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN HILLS, MINNESOTA    *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * OBTAIN DEBUG OBJECT TIME BINARIES FROM SCPOPL             *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(NEWSCP,L=$SCPOPL2.0$)
STAGE(NEWSCP,MT,HY,E,PRF,
VSN=CY51)
REWIND(NEWSCP)
COPYP(NEWSCP,OLDPL)
UNLOAD(OLDPL)
COPYP(NEWSCP,DBUGBIN)
REWIND(DBUGBIN)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF DEBUG OBJECT TIME BINARIES *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG DBUGLIB7  (DEBUG OBJECT TIME LIBRARY)           *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(DBUGLIB,DBUGLIB7,ID=PROLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(DBUGLIB)
PAUSE.  MP006 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
```

```
JOB DECK  DBUGINS


EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
LIBRARY(DBUGLIB,NEW=2000)
REPLACE(*,DBUGBIN)
FINISH.
LISTLIB(*,DBUGLIB,N=1)
---EOI--- 6//7/8/9
```

6.3.3 SCPMOD

JOB DECK SCPMOD

```
SCPMOD,T7000,CP70,MT62.
COMMENT. *SVDV,07155,36A3D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                   SCOPE 2.0 INTEGRATION ARH248           *
COMMENT. *                   4201 LEXINGTON AVE. N.                 *
COMMENT. *                   ST PAUL, MINN.    55112                *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                       *
COMMENT. *                                                          *
COMMENT. ************************************************************
ATTACH(MODPL,MODPL,ID=S20UPSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,8,*=/,C=UPIN)
RETURN(MODPL)
COPYS(INPUT,SUP)
REWIND(SUP)
COPYS(INPUT,SUP)
COPYS(INPUT,FLIB)
COPYS(INPUT,SLIB)
REWIND(FLIB,SLIB)
LABEL(SCPOPL,L=$SCPOPL2.0$,R)
STAGE(SCPOPL,MT,HY,E,PRF,
VSN=CY47)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE SCPOPL, CREATE SEQUENTIAL NEWPL                    *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P=SCPOPL,F,C=0,I=UPIN,N=RANDPL)
UNLOAD(SCPOPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
```

```
JOB DECK  SCPMOD

COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED    *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.*
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN       *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).       *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
UPDATE(P=RANDPL,F,B,C=0,N,I=NULL)
RETURN(RANDPL)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * UPDATE NEWPL INSERTING SYSTEM HEADER INFORMATION.                  *
COMMENT. * ALLOCATION A4 IS USED FOR COMPILE TO AVOID FILE OVERFLOW           *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
REQUEST(COMPILE,A4,T3)
UPDATE(P=NEWPL,F,I=SUP)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * ASSEMBLE OST                                                       *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
COMPASS(I,S=0,R=OST,L=0,A)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * ASSEMBLE SYSTEM OVERLAYS                                           *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
COMPASS(I,G=OST,B=OVLLGO,L=0,A)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * ASSEMBLE  V2TEXT                                                   *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
COMPASS(I,S=0,R=V2TEXT,L=0,A)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * ASSEMBLE SYSTEM UTILITIES                                          *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
COMPASS(I,G=V2TEXT,B=UTLLGO,L=0,A)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * ASSEMBLE  DEBUG LIBRARY                                            *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
COMPASS(I,S=0,R=DBUGRTN,L=0,A)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
```

```
JOB DECK  SCPMOD

COMMENT. * COMPILE  ANALYZE AND LIBEDT                                          *
COMMENT. * USING LATEST FTN COMPILER (ON HIGHEST CYCLE OF PRDLIB)                *
COMMENT. *                                                                      *
COMMENT. ********************************************************************
ATTACH(PRDLIB,PRDLIB,ID=PRDLIB)
LIBRARY(PRDLIB)
FTN(I=COMPILE,L=0,B=LBANLGO)
LIBRARY.
REWIND(LBANLGO)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * SAVE AUTO MODULE FROM COMPILE                                      *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
COPYS(COMPILE,AUTO7)
RETURN(COMPILE)
REWIND(AUTO7)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * CREATE NEW SCPOPL TAPE...FIRST PARTITION= NEWPL,                   *
COMMENT. * SECOND PARTITION= DEBUG LIBRARY BINARIES                           *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
LABEL(NEWSCP,L=$SCPOPL2.0$)
STAGE(NEWSCP,MT,HY,N,POST,
VSN=CY51)
REWIND(NEWPL,DBUGBIN)
COPYP(NEWPL,NEWSCP)
UNLOAD(NEWPL)
COPYP(DBUGBIN,NEWSCP)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * TEST NEW TAPE VIA UPDATE                                          *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
UNLOAD(NEWSCP)
STAGE(NEWSCP,MT,HY,E,PRE,
VSN=CY51)
REWIND(NEWSCP)
COPYP(NEWSCP,OLDPL)
COPYP(NEWSCP,NEWBIN)
UNLOAD(NEWSCP)
UPDATE(P,N,A,F,C=0,I=NULL)
RETURN(OLDPL)
RETURN(DBUGBIN,NEWBIN)
UNLOAD(NEWSCP)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * BUILD LIBEDT AND ANALYZE FROM RELOCATABLE BINARY.                  *
```

```
JOB DECK  SCPMOD

COMMENT. *        1. BUILD A LIBRARY CONTAINING RELOCATABLE CODE.                *
COMMENT. *        2. DO A LIBLOAD AND NOGO FOR EACH PROGRAM SATISFYING          *
COMMENT. *           EXTERNALS FROM THE LIBRARY IN STEP 1.                       *
COMMENT. *                                                                       *
COMMENT. ***************************************************************************
LIBEDT(I=FLIB,M)
ATTACH(FTNLIBX,FTNICLIB,ID=PRDLIB)
COMMENT. ***************************************************************************
COMMENT. *                                                                       *
COMMENT. * SATISFY EXTERNALS FOR DOING I/O FROM LATEST FORTRAN OBJECT LIBRARY    *
COMMENT. *                                                                       *
COMMENT. ***************************************************************************
LIBRARY(UTLREL,FTNLIBX)
MAP(ON)
LIBLOAD(UTLREL,OSULIB)
NOGO(LIBEDT7,LIBEDT,CONTENT,COPYLB,EDITLIB)
LIBLOAD(UTLREL,ANLYSIS)
NOGO(ANAL7,ANALYZE)
REWIND(UTLLGO)
LIBRARY.
COMMENT. ***************************************************************************
COMMENT. *                                                                       *
COMMENT. * BUILD SYSTEM USING LIBEDT. PICK UP PRODUCTS FROM PRDLIB.              *
COMMENT. * PUNCH BINARIES FOR DEADSTART DECK                                     *
COMMENT. * CATALOG SYSLIB (SYSTEM LIBRARY WITHOUT PRODUCT SET)                   *
COMMENT. *                                                                       *
COMMENT. ***************************************************************************
LIBEDT(I=SLIB,M)
COMMENT. ***************************************************************************
COMMENT. *                                                                       *
COMMENT. * CREATE NEW DEADSTART TAPE                                            *
COMMENT. *                                                                       *
COMMENT. ***************************************************************************
FILE(NDSTAPE,RT=W,RT=C,FO=SQ)
STAGE(NDSTAPE,MT,HI,POST,
VSN=CY51)
REWIND(NDSTAPE)
FILE(NEWLIB,RT=W,FO=SQ)
COPY(NEWLIB,NDSTAPE)
UNLOAD(NDSTAPE)
CATALOG(SYSLIB,SYSLIB,ID=PRDLIB,PW=SCPCN,CN=SCPCN,EX=SCPFX,MD=SCPMD,
CY=51)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C SUPSR
---EOS--- 7/8/9
*/    ************************  SET     HEADER  ************************
```

```
JOB DECK  SCPMOD

*/
*/  SET SYSTEM IDENTIFICATION
*/  ( 41-50 CHARACTER DATA STRING)
*/
*/
*/
*ID PDX
*DC ORL
*D  ORL.418
          DATA   H$1**** 7000 SCOPE 2.0 **** SYSTEM 51T      $
*/   (WHERE NN IS THE LATEST PSR SUMMARY LEVEL AND A IS AN ALPHA SUFFIX)
*/
*/  SET INSTALLATION IDENTIFICATION - DATA CARD MUST BE REPLACED
*/  ( 41-50 CHARACTER DATA STRING)
*/
*/
*D ORL.421
          DATA   H$0(THIS CARD SHOULD BE REPLACED WITH INSTALL.ID)$
*/
*/
*/    ************************  SET    BUFCHK   ************************
*/
*/
*/
*/  THE FOLLOWING 2 CARDS WILL CAUSE BUFCHK TO CHECK FOR UNRELEASED
*/  BUFFERS AND CRASH THE SYSTEM IF ONE SHOULD BE DETECTED
*D S00163AB.2
          VFD    60/1               ABORT IF BUFFER RELEASE BUG
*/
*/
*/
*/    ******************** SET INSTALLATION PARAMETERS ********************
*/
*/
*/    ******************  MAKE INSTALLATION CHANGES  ********************
*/
*/
---EOS--- 7/8/9
LIBRARY(UTLREL,NEW=1200)
REWIND(UTLLGO,LBANLGO)
TYPE(REL)
REPLACE(*,UTLLGO)
REPLACE(*,LBANLGO)
FINISH.
LISTLIB(*,UTLREL,N=1)
---EOS--- 7/8/9
LIBRARY(SYSLIB,NEW=2500)
REWIND(OST,OVLLGO,V2TEXT,UTLLGO,ANAL7,LIBED17)
REPLACE(OST,OST)
```

```
JOB DECK  SCPMOD


REPLACE(*,OVLLGO)
REPLACE(V2TEXT,V2TFXT)
REPLACE(*,AUTO7)
REPLACE(USUDMPL+COPYBR,UTLLGO)
REPLACE(UPDATE,UTLLGO)
REPLACE(*,ANAL7)
REWIND(LIBEDT7)
REPLACE(*,LIBEDT7)
*/ PUNCH BINARIES FOR DS DECK - 2 COPIES
PCOPY(USR/CPL/MTD/TDS/DSC/DS1,PUNCHB)
PCOPY(USR/CPL/MTD/TDS/DSC/DS1,PUNCHB)
FINISH.
LISTLIB(*,SYSLIB,N=1)
LIBRARY(NE*LIB,NEW=2500)
CLOLIB(PPDLIB)
REPLACE(*,SYSLIB,LIB)
*/
*/ USE THE DIRECTIVE ... DELETE(FIN+$PASS14$$$) ... IF FORTRAN EXTENDED
*/ IS NOT AN INSTALLED PRODUCT (USED ONLY FOR SYSTEM AND PRODUCT MAINTAINENCE)
*/
*/
FINISH.
LISTLIB(*,NEWLIB,N=1)
---EOS--- //8/9
---EOI--- 6/7/8/9
```

6.4 SYSTEM INFORMATION FILE (SIF)

The SIF is generated by SCOPE 2.0 during the execution of the system and serves as a general log of hardware, system, and job activities. The installation may also add code to the system to collect a variety of information associated with other activities such as debugging and performance measurement. Section 6.4.2 gives details and an example of adding information classification to the SIF. The user obtains the contents of the current SIF by executing the RTRVSIF control statement. The ANALYZE control statement enables the user to extract and reduce the information desired. Section 6.4.1 defines the RTRVSIF parameters and section 6.5 gives instructions on how to add any directives for ANALYZE. Section 6.4.3 details the content of SIF as released. Two macros, SIFX and SIFJ, add data to the SIF. The SIFX adds data from the executive level. The SIFJ adds data from the job supervisor level.

SCOPE 2.0 formats data in the SIF as type W records. Other format conventions are given in the following paragraphs.

The first word (word 0) of each record contains date and time consisting of the following three fields.

Bits 59 through 51	Year in binary
Bits 50 through 42	Day in year in binary
Bits 41 through 00	Time of day expressed in clock periods

The first two fields constitute a Julian date. The contents of word 0 are generated as the result of the SIFX/J macro call.

Various parts of SCOPE 2.0 format the rest of the record, depending on the part of the system that creates or captures the data. For example, an I/O queue manager discovers an error and formats a record including room for word 0 and issues a SIFX/J macro to add the record to the SIF.

The SIF follows mainly a binary code format. Each SIF record of an installation format is identified by a unique 6-bit number between 00 and 77 octal. Each record written to the SIF contains the value of this 6-bit record code which identifies the format of the record. An installation must establish a unique record code for any added installation defined record formats. The record code appears in word 3 (bits 59 through 54) of each record. The record code also classifies the general contents of the record and indicates the prefix to be used for all field location symbols associated with the record. The four classifications are as follows.

Record Code (Octal)	Symbol	Contents
00 through 17	SIHxxx	Hardware information
20 through 37	SISxxx	System information
40 through 57	SIJxxx	Job information
60 through 77	SIXxxx	Miscellaneous information

There are four common decks (SIH, SIS, SIJ, SIX) which contain the various descriptions of records in the corresponding classification. For example, an installation could add a record that documents an error in drum I/O by calling the record SIHDRM, giving it a record format value of 10B and placing the record in the SIH common deck. Similarly, an installation might add a debugging record called SIXDBG, with a record format value of 70B, and appearing in the SIX common deck.

Macros are available to describe the format of added records. These macros also supply individual fields (and the record format as a whole) with unique field location symbols which may be used to construct the record and later analyze it via the program ANALYZE (refer to the SCOPE 2.0 Reference Manual). These macros are discussed in detail in section 6.4.2.

6.4.1 RTRVSIF - RETRIEVE SYSTEM INFORMATION FILE

The RTRVSIF control statement retrieves the current contents of the SIF and begins a new SIF. The current contents of the SIF are cataloged as a permanent file before being made available to the job. The file lfn must not exist before this function. For a detailed description of the SIF, refer to section 6.4.2.

RTRVSIF(lfn, pfn, p_1, p_2, . . . , p_n)

Parameters:

lfn	Logical file name, required parameter
pfn	Permanent file name, required parameter

Optional parameters:

PP	Privacy procedure. Written PP=procedure†
RP	Retention period. Written RP=number (0 through 999)
CY	Cycle number. Written CY=number (0 through 63)
TK	Turnkey password. Written TK=password†
CN	Control password. Written CN=password†
MD	Modify password. Written MD=password†
EX	Extend password. Written EX=password†
RD	Read password. Written RD=password†
ID	Creator identification. Written ID=name†
PW	Password list (has meaning only when new cycle is cataloged. Written PW=list.†

6.4.2 SIF DEFINITION FORMAT AND MACROS

As an introduction to the detailed content of the released SIF and as a guide to an installation adding record classifications, the following macros are described. The macros define the format of the SIF record.

1. RECHDR defines the name of the record and the code that identifies the record format. This macro must precede all of the field definitions for the record. The macro is written:

LOCATION	OPERATION	VARIABLE SUBFIELDS
symbol	RECHDR	value

†From 1 to 9 alphanumeric characters

symbol	Section 6.4.3
value	Octal code from section 6.4.3

2. FIELD defines an individual field in a record. The macro is written:

LOCATION	OPERATION	VARIABLE SUBFIELDS
symbol	FIELD	wp,lb,rb

symbol	Name of field (it must begin with the same prefix as the record name)
wp	Word offset
lb	Upper bit
rb	Lower bit of the field

3. OCTNUM, DECNUM, DISPLAY define how the ANALYZE directive LIST is to display the field. Each of the macros is written:

LOCATION	OPERATION	VARIABLE SUBFIELDS
symbol	macro	

symbol	Corresponding FIELD macro. The FIELD must occur after the OCTNUM, DECNUM, or DISPLAY macro.
macro	OCTNUM, DECNUM, or DISPLAY

4. RECEND terminates the description of the record. This macro must follow all of the field definitions. The name field and parameter fields are ignored.

LOCATION	OPERATION	VARIABLE SUBFIELDS
	RECEND	

The general format of a SIF record description is:

```
                TITLE SIqxxx - title for record
*
*               Description of record including
*               General contents, how, and why
*               Placed in the SIF
*
SIqxxx          RECHDR  num
    SIqaaa      FIELD   ...
    SIqbbb      FIELD   ...
                ...
                RECEND
```

To illustrate, an installation might add a new record to collect information about the progress of jobs through the system by outputting a record at each major milestone in the processing of a job. The record would be described as follows.

```
                TITLE SISJOB - JOB MILESTONE RECORD
*
*               THIS RECORD IS OUTPUT WHENEVER A JOB PASSES
*               A MILESTONE IN ITS PROCESSING
*
 SISJOB         RECHDR 30B
 SISJNM         DISPLAY
 SISJNM         FIELD 1,59,18       JOB NAME
 SISJCD         FIELD 1,17,00       MILESTONE CODE
*                                   =00 - BEGIN INPUT FROM STATION
*                                   =01 - END INPUT FROM STATION
*                                   =02 - BEGIN JOB
*                                   =03 - END JOB
*                                   =04 - BEGIN OUTPUT TO STATION
*                                   =05 - END OUTPUT TO STATION
 SISJAM         FIELD 2,59,00       ADDITION INFORMATION
 SISJRF         FIELD 3,59,54       RECORD CODE (=30B)
                RECEND
```

For this example, the installation places this code in the common deck SIS. The installation then adds code to the appropriate parts of the operating system to place the operating data in the fields of this record. Refer to operating system listing for additional detail and examples. The routines building the record must include UPDATE *CALLs for the common decks SIFMACR (which defines all of the macros in this section except FIELD) and SIS, in that order.

In order to use the symbol field locations defined in the record, the deck OSUANLZ must be reassembled with the modified common deck SIH and ANALYZE rebuilt (s ction 6.5). It is not necessary to do this, however, if the user needs nothing more than octal dumps. It is also possible to copy records based on an absolute nonsymbolic record format number.

Example:

```
RTRVSIF(SIFDATA,SIF,CY=10,PW=SPECIAL)
```

Retrieval and catalog current contents of SIF as cycle 10 of permanent file SIF.

For another example, the following program can be used to dump the dayfile from the system information file. In addition it creates a permanent file (PFILE) from the current contents of the SIF.

```
JOHN,CP70.
PASSWRD(BLANKET)
RTRVSIF(MYFILE,PFILE)
ANALYZE(M)
7/8/9
SIF(MYFILE,R)
LIST(RTYPE=SIJJDF)
6/7/8/9
```

If further information were to be retrieved from this same portion of the SIF it would have to be retrieved from the permanent file (PFILE). For example, to get a second dump of the same dayfile dumped previously, the following program is required.

```
JOHN,CP70.
ATTACH(MYFILE,PFILE)
ANALYZE(M)
7/8/9
SIF(MYFILE,R)
LIST(RTYPE=SIJJDF)
6/7/8/9
```

6.4.3 STANDARD RECORD FORMATS IN THE RELEASED SCOPE 2.0

Octal Code	Symbol	Contents
01	SIHMCU	MCU keyword entry record
02	SIHGCP	Diagnostic control program message
03	SIHCEP	Central parity error record
04	SIHDSK	Disk error record
05	SIHTAP	Tape error record
06	SIHPEP	FLPP error record
07	SIHSTN	Unit record station record
14 through 17		Reserved for installation
20	SISFLR	File router error record
21	SISETO	Event table overflow record
22	SISTIM	Time snapshot record
23	SISQIN	Queue snapshot record
24	SISSNP	System information snapshot
25	SISCHQ	Channel queue information
26	SISDSD	Deadstart and recovery information
33 through 37		Reserved for installation
40	SIJJDF	Job dayfile record
41	SIJACT	Job accounting record
42	SIJPFM	Permanent file report record
43	SIJDST	Data Manager statistics record
54 through 57		Reserved for installation
67 through 75		Reserved for installation
76	SIXTIM	General debugging/timing record

6.4.4 SIH, SIS, SIJ, SIX RECORD FORMATS

The SIF record formats and field descriptions are contained in common decks on the SCOPE 2.0 OLDPL. A current listing of all the SIF records can be obtained by listing the common decks SIH, SIS, SIJ, SIX along with the SIF macro common deck SIFMACR.

Run the TXTVOL job on the MODTAPE (described in part III, section 1.6.2) to accomplish this.

6.5 MODIFICATION OF ANALYZE

This section describes the steps to modify the ANALYZE utility by the installation. The directives of the released version of ANALYZE are general and apply to a variety of record formats. However, the installation may add directives. The following section gives the steps to add a new directive.

1. Identify the new directive for the system

2. Code function for the new directive

3. Add new code to ANALYZE deck

The installation may reference further detail and examples if needed by obtaining a listing of the released ANALYZE module.

6.5.1 IDENTIFYING THE FUNCTION

The verb table VRBTAB contains one entry per function. The entry supplies the basic information needed by the general directive scan routine. Most new directives only require modification to the deck OSUANLZ (in the verb table) and to the ANALYZE generation deck (to include new decks in the load). Where nonstandard processing of a directive is required, the deck OSUAN1 may have to be modified.

The verb table entry is generated by a VERB macro entry which is written:

LOCATION	OPERATION	VARIABLE SUBFIELDS
name	VERB	program, space, ftndx, unmfld, numval, prcndx

name — Verb symbol, 1 to 7 characters, none of which is an ANALYZE delimiter.

program — Name of the entry point with a FORTRAN Extended calling sequence which performs the processing for the verb.

space — Storage space required by the function in the array FUNARY in the labeled common FUNCOM. All variables which must be maintained between the processing of individual records must be stored in FUNARY within the area indicated by space. This area should include words indicated by numfld and numval; if n variables are to be accumulated then space must be n + numfld + numval.

ftndx — Value of a constant to be placed in the function common variable FUNQAL to identify the verb where two or more verbs map to the same entry point. The value may not exceed 12 bits.

numfld — Number of variable field inputs to the verb. If numfld is 2, then FIELD/1 and FIELD/2 must both appear.

numval — Maximum number of constant inputs to the function. If numval is 1, then at most VALUE/1 may appear.

prendx — If other than zero, this indicates that the function is to be called during the cracking of the directive. In such a case FUNSEQ is set to minus one (-1).

Example:

```
AVERAGE VERB AVERAGE,6,,1,2
```

In the case of AVERAGE the example specifies: that the entry point AVERAGE processes the function indicated by the directive verb AVERAGE, that the function accumulates six local variables above and beyond the standard contents of the common FUNCOM, that one FIELD/1=parameter is required in the parameter field of the directive, and that up to two VALUE/i parameters may be present.

6.5.2 CODING THE FUNCTION

Because ANALYZE does most of the processing, the coding for directives may be very simple. The records are read, unpacked, and tested to see if the records should be processed. If a record is determined to be part of the input set for the directive, the copy of the function common FUNCOM is moved from LCM to SCM and the function is executed. Upon entry, the following variables are set.

FUNSEQ — Set by ANALYZE to facilitate initialization and final report generation. If PRCNDX is set in the VERB table, the function is called with FUNSEQ equal to minus one. The directive at this point is cracked into arrays FIELD and VALUE in the labeled common ANLCOM where it may be processed by the function.

Upon initial entry with the first record, FUNSEQ is set to 0. Initialization and processing of the first record can then occur. For all other records, FUNSEQ is set to 1. To indicate the end of the ANALYZE run, FUNSEQ is set to 2, and the function is executed one more time with no input record. This last execution is to complete the function output and print reports, graphs, etc.

FUNQAL — Set by ANALYZE from the verb table, FUNQAL is the function qualification value. This field would only be used if several verbs were processed by the same entrypoint, to determine the verb.

FUNCNT — Maintained by ANALYZE to indicate the number of records processed by the function.

The fields in RECOM that are set by ANALYZE for the function include:

RECNAM — The name of the current record, determined by looking the record code up in the symbol table. If the record code is not found, RECNAM is set to UNKNOWN.

RECODE — The current record code

RECBJD — The binary Julian date from the current record

RECWCT — The binary wall clock time of the current record in clock periods

RECLTH — The binary length of the current record in words

RECORD — The actual record, as read from the SIF

Upon initial entry to the function (FUNSEQ=0), the following variables are set in the array FUNARY in the labeled common FUNCOM.

FUNARY(1) through FUNARY (numfld) inclusive hold the indexes to RECOM for each unpacked (right justified, zero filled unless DISPLAY which is left justified, zero filled) input variable. If FIELD/i=a, then RECOM(FUNARY(i)) is the value of a, from the current record.

FUNARY(numfld+1) through FUNARY(numfld+numval) hold the constants specified by the VALUE/i parameter. If no VALUE/i parameter was specified, then the word would contain -1.

To prevent confusion it is suggested that the function code equivalence meaningful names to these locations.

There are many useful utility routines available to a new function. Specifically the installation should look at OSUXCV, OSUXSV, OSUXHT, and OSUXPL for the routines and the calling sequences. The following are of general interest.

1. CALL OSUDPT $\binom{0}{1}$ prints directive of the current function, plus */ comments preceding the directive.

   0 Double space before print
   1 Eject before print

2. WALTYM(n) where n is the time of day in clock periods, as logged in each SIF record header (RECWCT). Output is Δhh.mm.ssΔ in display code.

3. CALDAT(n) where n is the calendar Julian date in binary as logged in each SIF record header (RECBJD). Output is Δmm/dd/yyΔ in display.

   SYMNDX (6Lname) where name is display code (left justified, zero filled) for the desired record field. SYMNDX returns the index to RECFLD where that field was unpacked.

   Example:

   ```
   JOB = RECFLD(SYMNDX (6LSIJJNM))
   ```

   The user should not use the index unless the user knows RECFLD contains the record in which the filed occurs.

If the name is not known, a negative value is returned.

SYMVAL (6Lname) where name is the display code of the symbol of the desired record field. SYMVAL returns the value of the unpacked filed. If the field name is unknown, a negative one is returned. If the field is not available because the associated record is not in memory, a negative two is returned.

To enable access to FUNCOM and RECCOM, the two common decks should be called in the function. The decks are:

```
CALL    OSUAFC      ANALYZE FUNCTION COMMON
CALL    OSUARC      ANALYZE RECORD COMMON
```

Example:

1. Place SUM verb in verb table located in deck OSUANLZ.

```
SUM     VERB     =XSUM,3,,1,1
```

2. Code SUM function

```
                SUBROUTINE SUM
*               ANALYZE FUNCTION TO COMPUTE THE SUM
*               AND AVERAGE OF A SPECIFIED FIELD
*               (FIELD/1) WITH ONE CONVERSION
*               FACTOR (VALUE/1).
*CALL           OSUAFC
*CALL           OSUARC
                INTEGER FIELD1, VALUE1, TOTAL
*               ANALYZE/FUNCTION COMMUNICATION S
                EQUIVALENCE (FIELD1,FUNARY(1)),
                * (VALUE1, FUNARY(2)),
*               SAVE CUMMULATIVE TOTAL FOR EACH RTYPE SUM
                * (TOTAL,FUNARY(3))
*               INITIAL ENTRY POINT -
*               .... CHECK ENTRY CONDITION
                IF (FUNSEQ-1) 1000,2000,3000
*
*               INITIAL ENTRY - INITIALIZE FUNCTION
*
1000            TOTAL = RECCOM(FIELD1)
*               CHECK FOR DEFAULTED VALUE/1
                IF(VALUE1. LE.0) VALUE1=1
                RETURN
*
```

```
*               INTERMEDIATE RECORD INPUT-PROCESS IT
*
2000            TOTAL = TOTAL + RECCOM (FIELD1)
                RETURN
*
*               TERMINATION PROCESSING
*
3000            TOT = FLOAT (TOTAL)/FLOAT(VALUE1)
                AVRG = TOT/FLOAT (FUNCNT)
*               PRINT DIRECTIVE AND COMMENTS
                CALL OSUDPT(0)
*               PRINT SUM AND AVERAGE
                PRINT 9000, FUNCNT, TOT, AVRG
9000            FORMAT(*NUMBER OF RECORDS PROCESSED=*
                *,I10/*SUM OF RECORD=*,F15.6,
                */*AVERAGE VALUE=*,F15.6)
                RETURN
                END
```

With this code assembled in ANALYZE, a sample directive would be:

```
SUM(FIELD/1+SIJAJT,VALUE/1+CPSEC,RTYPE=SIJACT,SIJAJT=CPSEC+CPHR)
```

This gives the sum of job times greater than one second and less than one hour, as well as the average time of those jobs in seconds.

6.5.3 WRITING A LIST SUBFUNCTION

In the released version of ANALYZE there is a LIST function. For those record formats which are known, the printout of each record consists of a header (record name, length, time of day, and date entered into the SIF) and each field symbol along with the value of the symbol printed in octal, decimal, or display as described with the macros of section 6.4.2. If the record format is unknown, LIST merely dumps the record contents.

An exception to this is a LIST of the dayfile records. A special sublist routine lists these records just as they appear at the end of each job's output.

To add a new special list program for a specific record type to be called by the LIST directive, the installation should include a LISTER macro at the end of the deck OSUANLZ. The macro is written:

LOCATION	OPERATION	VARIABLE SUBFIELDS
program	LISTER	recode

program	Gives entry point name of subfunction
recde	Record format value or a symbol equated to it selected from section 6.4.3

Example:

```
DSKLST   LISTER   SIHDSK
```

Causes the routine DSKLST to be called whenever an SIHDSK record is the input of the LIST function.

7.1 REQUIREMENTS

The following materials are necessary to install COMPASS 2.0.

7.1.1 TAPES

CM2REL	Described in part I, section 2.7

7.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (CM2MOD jobs)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (CM2USR job)

7.1.3 DECKS

CM2GEN	Regenerates COMPASS binary from CM2REL and catalogs CM2REL (ID=PRDLIB)
CM2MOD	Creates a new CM2REL from the old CM2REL and MODPL, then generates the binary from the new CM2REL and catalogs CM2LEL (ID=PRDLIB)
CM2USR	Creates a new CM2REL from the old CM2REL and USERMODPL, then generates the binary from the new CM2REL and catalogs CM2LEI (ID=PRDLIB)
CM2INS	Catalogs binary from a CM2REL as CM2LEL (ID=PRDLIB)

7.1.4 DEPENDENCIES

None.

7.2 CONFIGURATION PARAMETERS

None.

7.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0

7.3.1 CM2INS

JOB DECK CM2INS

```
CM2INS,CP70,MT01,T1000.
COMMENT. *SVDV.07155,36P3D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO *
COMMENT. *                   SCOPE 2.0 INTEGRATION ARH248           *
COMMENT. *                   4201 LEXINGTON AVE. N.                 *
COMMENT. *                   ST PAUL, MINN.    55112                *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
COMMENT. ****************************************************************
COMMENT. *                                                              *
COMMENT. * INSTALL COMPASS FROM RELEASE TAPE                            *
COMMENT. *                                                              *
COMMENT. ****************************************************************
LABEL(CM2REL,L=$CM2REL*SCP2.0$)
STAGE(CM2REL,MT,HY,E,PRE,
VSN=CY51)
REWIND(CM2REL)
COPYP(CM2REL,OLDPL)
COPYP(CM2REL,CMPBIN)
COMMENT. ****************************************************************
COMMENT. *                                                              *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF COMPASS BINARIES              *
COMMENT. *                                                              *
COMMENT. ****************************************************************
LIBEDT(M)
COMMENT. ****************************************************************
COMMENT. *                                                              *
COMMENT. * CATALOG CM2LEL (USED BY NPRDLIB)                            *
COMMENT. *                                                              *
COMMENT. ****************************************************************
CATALOG(CM2LEL,CM2LEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
RETURN(CM2LEL)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
```

```
JOB DECK  CM2INS


---EOS--- 7/8/9
LIBRARY(CM2LEL,NEW=2000)
REWIND(CMPBIN)
REPLACE(*,CMPBIN)
FINISH.
LISTLIB(*,CM2LEL,N=1)
---EOI--- 6/7/8/9
```

7.3.2 CM2MOD

```
JOB DECK  CM2MOD

CM2MOD,CP70,T1000,MT01.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                   SCOPE 2.0 INTEGRATION ARH248           *
COMMENT. *                   4201 LEXINGTON AVE. N.                 *
COMMENT. *                   ST PAUL, MINN.    55112                *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ***************************************************************
COMMENT. *                                                             *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                         *
COMMENT. *                                                             *
COMMENT. ***************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,B,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ***************************************************************
COMMENT. *                                                             *
COMMENT. * UPDATE COMPASS OLDPL, CREATE SEQUENTIAL NEWPL               *
COMMENT. *                                                             *
COMMENT. ***************************************************************
LABEL(CM2REL,L=$CM2REL*SCP2.0$)
STAGE(CM2REL,MT,HY,E,PRF,
VSN=CY45)
UPDATE(P=CM2REL,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(CM2REL)
COMMENT. ***************************************************************
COMMENT. *                                                             *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.*
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN    *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).    *
COMMENT. *                                                             *
```

```
JOB DECK  CM2MOD

COMMENT. ************************************************************
UPDATE(P=RANDPL,F,C,N,W,I=NULL)
RETURN(RANDPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * ASSEMBLE COMPASS BINARIES                                *
COMMENT. *                                                          *
COMMENT. ************************************************************
COMPASS(I,B=CMPBIN,A,L=LIST)
RETURN(COMPILE)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF COMPASS BINARIES          *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG CM2LEL (USED BY NPRDLIB)                         *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(CM2LEL,CM2LEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
RETURN(CM2LEL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE NEW CM2REL TAPE...FIRST PARTITION= NEWPL,         *
COMMENT. * SECOND PARTITION= COMPASS OVERLAY                        *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(NEWCMP,L=$CM2REL*SCP2.0$)
STAGE(NEWCMP,MT,HY,N,POST,
VSN=CY51)
REWIND(NEWCMP,CMPBIN)
COPYP(NEWPL,NEWCMP)
COPYP(CMPBIN,NEWCMP)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * TEST NEW TAPE VIA UPDATE
COMMENT. *                                                          *
COMMENT. ************************************************************
UNLOAD(NEWCMP)
STAGE(NEWCMP,MT,HY,E,PRE,
VSN=CY51)
REWIND(NEWCMP)
COPYP(NEWCMP,OLDPL)
COPYP(NEWCMP,NEWBIN)
UNLOAD(NEWCMP)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
```

```
JOB DECK  CM2MOD

RETURN(OLDPL,SCRAT,NEWBIN)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C CM2PSR
---EOS--- 7/8/9
LIBRARY(CM2LEL,NEW=2000)
REWIND(CMPBIN)
REPLACE(*,CMPBIN)
FINISH.
LISTLIB(*,CM2LEL,N=1)
---EOI--- 6/7/8/9
```

8.1 REQUIREMENTS

The following materials are necessary to install FORTRAN Extended 2.0

8.1.1 TAPES

FN2REL	Described in part I, section 2.8

8.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (FN2MOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (FN2USR job)

8.1.3 DECKS

FN2GEN	Regenerates FTN binary from FN2REL and catalogs FN2LEL (ID=PRDLIB)
FN2MOD	Creates a new FN2REL from the old FN2REL and MODPL, then generates the binary from the new FN2REL and catalogs FN2LEL (ID=PRDLIB)
FN2USR	Creates a new FN2REL from the old FN2REL and USERMODPL, then generates the binary from the new FN2REL and catalogs FN2LEL (ID=PRDLIB)
FN2INS	Catalogs binary from FN2REL as FN2LEL (ID=PRDLIB)

8.1.4 DEPENDENCIES

None.

8.2 CONFIGURATION PARAMETERS

The following assembly options and parameters are defined in deck FTN.

Parameter	Released Default Value	Significance
CCABT	1	Option to abort the job if there is an error on the FTN control card. 0 No abort 1 Abort on control card error
CTIMO	1	Option to issue compilation time message to dayfile at completion of an FTN compilation. 0 No message 1 Issue dayfile message for CPU compilation time
DEF.DFL	60000_8	FTN raises the SCM field length to this value in system controlled field length mode and aborts the job in user controlled field length mode if less than MIN. DFL is supplied for an FTN D option compilation. DEF. DFL may not be set to less than MIN. DFL.
DEF. FL	50000_8	FTN raises the SCM field length to this value in system controlled field length mode and aborts the job in user controlled field length mode if less than MIN. FL is supplied for an FTN compilation. DEF. FL may not be set to less than MIN. FL.
LMAX	57D	The source listing from an FTN compilation contains LMAX lines per page; does not apply to intermixed COMPASS programs.
MIN. DFL	57000_8	Minimum SCM field length necessary for an FTN compilation if the D option is selected on the FTN control card. MIN. DFL may not be set to less than MIN. FL + 17000B.
MIN. FL	40000_8	Minimum SCM field length necessary for an FTN compilation. IP. MNFL may not be set to less than 40000_8.

8.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

8.3.1 FN2INS

JOB DECK FN2INS

```
FN2INS,CP70,T1000,MT01.
COMMENT. *SVDV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * INSTALL FTN FROM RELEASE TAPE                            *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(FN2REL,L=$FN2REL*SCP2.0$)
STAGE(FN2REL,MT,HY,E,PRE,
VSN=CY47)
REWIND(FN2REL)
COPYP(FN2REL,OLDPL)
COPYP(FN2REL,FTN)
REWIND(FTN)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF FTN OVERLAY              *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG FN2LEL (USED BY NPRDLIB)                         *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(FN2LEL,FN2LEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(FN2LEL)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
```

```
JOB DECK  FN2INS

PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
LIBRARY(FN2LEL,NEW=2000)
REWIND(FTN)
REPLACE(*,FTN)
FINISH.
LISTLIB(*,FN2LEL,N=1)
---EOI--- 6///8/9
```

8.3.2 FN2MOD

JOB DECK FN2MOD

```
FN2MOD,CP70,T1000,MT01.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                   SCOPE 2.0 INTEGRATION ARH248           *
COMMENT. *                   4201 LEXINGTON AVE. N.                 *
COMMENT. *                   ST PAUL, MINN.    55112                *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
LABEL(FN2REL,L=$FN2REL*SCP2.0$)
STAGE(FN2REL,MT,HY,E,PRE,
VSN=CY45)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                       *
COMMENT. *                                                          *
COMMENT. ************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,8,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  UPDATE  FTN OLDPL, CREATE SEQUENTIAL NEWPL              *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P=FN2REL,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(FN2REL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.*
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN     *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).     *
COMMENT. *                                                          *
```

JOB DECK FN2MOD

```
COMMENT. ************************************************************************
UPDATE(P=RANUPL,F,B,C=6,N,I=NULL)
RETURN(RANUPL)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *  UPDATE NEWPL INSERTING CURRENT CYCLE INDICATOR                      *
COMMENT. *  AND PRODUCE COMPILE FILE                                            *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
UPDATE(P=NEWPL,C,F)
SKIPF(COMPILE,2)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *  ASSEMBLE A BASIC FTN COMPILER                                       *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
COMPASS(I=COMPILE,B=SYSMAIN,L=LIST,A)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * PRODUCE  BASIC FTN OVERLAY                                          *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
REWIND(SYSMAIN)
LOAD(SYSMAIN)
NOGO.
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * MAKE LIBEDT FORMAT LIBRARY OF BASIC ABSOLUTE FTN -NEWFTN             *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
LIBEDT(M)
RETURN(FTN)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * USING BASIC FTN, COMPILE THE REST OF THE COMPILER                    *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
LIBRARY(NEWFTN)
FTN(I=COMPILE,OPT=1,R=3,A,B=SYS15,L=LIST)
FTN(I=COMPILE,OPT=1,R=3,A,B=DBGSKEL,L=LIST)
FTN(I=COMPILE,OPT=1,R=3,A,B=DBGSKEL,L=LIST)
LIBRARY.
RETURN(COMPILE,NEWFTN)
REWIND(SYSMAIN,SYS15,DBGSKEL)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE FTN RELOCATABLE BINARIES                                     *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
```

```
JOB DECK  FN2MOD

COPYL(DBGSKEL,SYSMAIN,SYSDBG)
RETURN(DBGSKEL)
REWIND(SYSMAIN,SYSDBG)
COPYP(SYSMAIN,SYSFTN)
SKIPB(SYSFTN)
COPYP(SYS15,SYSFTN)
SKIPB(SYSFTN)
COPYP(SYSDBG,SYSFTN)
RETURN(SYSMAIN,SYS15,SYSDBG)
REWIND(SYSFTN)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  PRODUCE  FTN OVERLAY                                    *
COMMENT. *                                                          *
COMMENT. ************************************************************
LOAD(SYSFTN)
NOGO.
REWIND(FTN)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF FTN OVERLAY              *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG FN2LEL (USED BY NPRDLIB)                        *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(FN2LEL,FN2LEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(FN2LEL)
LABEL(NEWFTN,L=$FN2REL*SCP2.0$)
STAGE(NEWFTN,MT,HY,N,POST,
VSN=CY47)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  CREATE NEW FN2REL TAPE...FIRST PARTITION - NEWPL,       *
COMMENT. *  SECOND PARTITION - FTN OVERLAY                          *
COMMENT. *                                                          *
COMMENT. ************************************************************
REWIND(NEWFTN,FTN)
COPYP(NEWPL,NEWFTN)
COPYP(FTN,NEWFTN)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * TEST THE NEW TAPE VIA UPDATE                            *
COMMENT. *                                                          *
COMMENT. ************************************************************
```

```
JOB DECK  FN2MOD

UNLOAD(NEWFTN)
STAGE(NEWFTN,MT,HY,E,PRE,
VSN=CY47)
REWIND(NEWFTN)
COPYP(NEWFTN,OLDPL)
COPYP(NEWFTN,NEWBIN)
UNLOAD(NEWFTN)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
RETURN(OLDPL,SCRAT,NEWBIN)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C FN2PSR
---EOS--- 7/8/9
*ID  CYCLEND
*1   CYCLE.2
 CYNR     DECMIC 47
---EOS--- 7/8/9
LIBRARY(NEWFTN,NEW=2000)
REWIND(FTN)
REPLACE(*,FTN)
FINISH.
---EOS--- 7/8/9
LIBRARY(FN2LEL,NEW=2000)
REWIND(FTN)
REPLACE(*,FTN)
FINISH.
LISTLIB(*,FN2LEL,N=1)
---EOI--- 6/7/8/9
```

9.1 REQUIREMENTS

The following materials are necessary to install FORTRAN Run 2.0.

9.1.1 TAPES

RN2REL	Described in part I, section 2.9

9.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (RN2MOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (RN2USR job)

9.1.3 DECKS

RN2GEN	Regenerates FORTRAN Run binary from RN2REL and catalogs RN2LEL (ID=PRDLIB)
RN2MOD	Creates a new RN2REL from the old RN2REL and MODPL, then generates the binary from the new RN2REL and catalogs RN2LEL (ID=PRDLIB)
RN2USR	Creates a new RN2REL from the old RN2REL and USERMODPL, then generates the binary from the new RN2REL and catalogs RN2LEL (ID=PRDLIB)
RN2INS	Catalogs binary from RN2REL as RN2LEL (ID=PRDLIB)

9.1.4 DEPENDENCIES

None.

9.2 CONFIGURATION PARAMETERS

The following installation parameters are defined in deck RUN.

Parameter	Released Default Value	Significance
IP. DMSZ	54000_8	If less than IP. MINMS is supplied for a RUN compilation, RUN raises the SCM field length to this value; cannot be set smaller than IP. MINMS.
IP. MINMS	45000_8	Minimum SCM field length required for a RUN compilation; cannot be set to less than 45000_8.
IP. PGSZ	58	The source listing from a RUN compilation contains IP. PGSZ lines per page; does not apply to intermixed COMPASS programs. This parameter is defined twice, in the 0, 0 and the 1, 0 overlays.

9.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

9.3.1 RN2INS

```
JOB DECK  RN2INS


RN2INS,CP70,MT01,T1000.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * INSTALL RUN FROM RELEASE TAPE                            *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(RN2REL,L=$RN2REL*SCP2.0$,R)
STAGE(RN2REL,MT,HY,E,PRF,
VSN=CY47)
REWIND(RN2REL)
COPYP(RN2REL,OLDPL)
COPYP(RN2REL,ZYX)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE  LIBEDT FORMAT LIBRARY OF RUN OVERLAY             *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG RN2LFL (USED BY NPRDLIB)                         *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(RN2LFL,RN2LFL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(RN2LFL)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
```

```
JOB DECK BEGINS

---EOS--- //G/G
LIBRARY(RNZLEL,NEW=2000)
REWIND(ZYX)
REPLACE(*,ZYX)
FINISH.
LISTLIB(*,RNZLEL,N=1)
---EOI--- G///R/9
```

9.3.2 RN2MOD

JOB DECK RN2MOD

```
RN2MOD,CP70,T1000,M101.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                   SCOPE 2.0 INTEGRATION ARH248           *
COMMENT. *                   4201 LEXINGTON AVE. N.                 *
COMMENT. *                   ST PAUL, MINN.    55112                *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
LABEL(RN2REL,L=$RN2REL*SCP2.0$,R)
STAGE(RN2REL,MT,HY,E,PRF,
VSN=CY45)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                       *
COMMENT. *                                                          *
COMMENT. ************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,8,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   UPDATE  RUN OLDPL, CREATE SEQUENTIAL NEWPL             *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P=RN2REL,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(RN2REL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL. *
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN  *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).  *
COMMENT. *                                                          *
```

```
JOB DECK  RN2MOD

COMMENT. ***************************************************************************
UPDATE(P=RANDPL,F,B,C=0,N,I=NULL)
RETURN(RANDPL)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. *  UPDATE NEWPL INSERTING CURRENT CYCLE INDICATOR                         *
COMMENT. *  AND PRODUCE COMPILE FILE                                               *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
UPDATE(P=NEWPL,C,F)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. *  ASSEMBLE TO CREATE RELOCATABLE BINARIES                                *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
COMPASS(I=COMPILE,B=RBIN,A,L=LIST)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. *  PRODUCE RUN OVERLAY ON FILE ZYX                                        *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
REWIND(RBIN)
LOAD(RBIN)
NOGO.
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. * CREATE  LIBEDT FORMAT LIBRARY OF RUN OVERLAY                           *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
LIBEDT(M)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. * CATALOG RN2LEL (USED BY NPROLIB)                                       *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
CATALOG(RN2LEL,RN2LEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(RN2LEL)
LABEL(NEWRUN,L=$RN2REL*SCP2.0$)
STAGE(NEWRUN,MT,HY,N,POST,
VSN=CY47)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. * CREATE NEW RN2REL TAPE...FIRST PARTITION - NEWPL,                       *
COMMENT. * SECOND PARTITION - RUN OVERLAY                                         *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
REWIND(NEWRUN,ZYX)
COPYP(NEWPL,NEWRUN)
```

```
JOB DECK  RN2MOD

COPYP(ZYX,NEWRUN)
COMMENT. ********************************************************************
COMMENT. *                                                                  *
COMMENT. * TEST THE NEW TAPE VIA UPDATE                                     *
COMMENT. *                                                                  *
COMMENT. ********************************************************************
UNLOAD(NEWRUN)
STAGE(NEWRUN,MT,HY,E,PRF,
VSN=CY47)
REWIND(NEWRUN)
COPYP(NEWRUN,OLDPL)
COPYP(NEWRUN,NEWBIN)
UNLOAD(NEWRUN)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
RETURN(OLDPL,SCRAT,NEWBIN)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C RN2PSR
---EOS--- 7/8/9
*ID  RN2CYC
*D   COPYR1.2
          DATA   10L CYCLE 47
---EOS--- 7/8/9
LIBRARY(RN2LEL,NEW=2000)
REWIND(ZYX)
REPLACE(*,ZYX)
FINISH.
LISTLIB(*,RN2LEL,N=1)
---EOI--- 6/7/8/9
```

FORTRAN OBJECT TIME ROUTINES 10

10.1 REQUIREMENTS

The following materials are necessary to install the FORTRAN object time routines.

10.1.1 TAPES

FCLOPL	SCOPE 3.4 release tape described in SCOPE 3.4 Installation Handbook
MODS3P4	SCOPE 3.4 modification tape described in SCOPE 3.4 Installation Handbook

10.1.2 PERMANENT FILES

USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (FTNLIBX job)
PRDLIB (ID=PRDLIB)	Necessary to compile part of FTNIOLIB

10.1.3 DECKS

FTNLIB7	Creates a new FCLOPL from the old FCLOPL and MODS3P4
FLIBGEN	Generates and catalogs the binary as FTNIOLIB (ID=PRDLIB), FTN object time routines, from an FCLOPL; requires use of PRDLIB (ID=PRDLIB)
RLIBGEN	Generates and catalogs the binary as RUNIOLIB (ID=PRDLIB), Run object time routines, from an FCLOPL
FTNLIBX	Creates a new FCLOPL from the old FCLOPL and USERMODPL

10.1.4 DEPENDENCIES

FORTRAN Extended must be installed before FCL at least to the extent that PRDLIB contains a FTN compiler updated to the current PSR summary level.

10.2 CONFIGURATION PARAMETERS

The following parameter is in common deck LIBMAC.

Parameter	Released Default Value	Significance
CALL	0	Indicates which library to assemble. 0 FTN object library 1 RUN object library

10.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

10.3.1 FTNLIB7

JOB DECK FTNLIB7

```
FTNLIB7,CP70,T1000,MT02.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                  SCOPE 2.0 INTEGRATION ARH248            *
COMMENT. *                  4201 LEXINGTON AVE. N.                  *
COMMENT. *                  ST PAUL, MINN.    55112                 *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * THIS JOB APPLIES MODS FROM THE SCOPE 3.4 MODTAPE (OR USERMODPL) TO  *
COMMENT. * THE FORTRAN COMMON OBJECT LIBRARY PRODUCING A NEWPL TO BE USED BY   *
COMMENT. * FLIBGEN AND/OR RLIBGEN TO GENERATE BINARIES                        *
COMMENT. *                                                          *
COMMENT. ************************************************************
FILE(MODS3P4,RT=S)
STAGE(MODS3P4,MT,HI,E,PRE)
UPDATE(P=MODS3P4,Q,*=/,C=UPIN,8,D)
UNLOAD(MODS3P4)
FILE(OLDPL,RT=S)
LABEL(OLDPL,L=$FTNLIBS*3P4$)
STAGE(OLDPL,MT,HI,E,PRE,
VSN=CY45)
LABEL(NEWPL,L=$FTNLIBS*3P4$)
FILE(NEWPL,RT=S)
STAGE(NEWPL,MT,HI,N,POST,
VSN=CY51)
UPDATE(P,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(OLDPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.*
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN      *
```

```
JOB DECK  FTNLTR7

COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).      *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
UPDATE(F=RANDPL,F,B,C=0,N,I=NULL)
RETURN(RANDPL)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * TEST NEW TAPE VIA UPDATE                                          *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
UNLOAD(NEWPL)
LABEL(NEWPL,L=$FTNL1BS*3P44)
FILE(NEWPL,RT=S)
STAGE(NEWPL,MT,HI,E,PRE,
VSN=CY51)
UPDATE(F=NEWPL,N=SCRAT,F,W,C=0,I=NULL)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- //8/9
//
// THE FOLLOWING EXTRACT FTN-RUN COMMON OBJECT LIBRARY MODS (IF ANY)
// FROM MODS3P4 (3.4 MODTAPE)
/C FCL
---EOI--- 6///8/9
```

10.3.2 FLIBGEN

JOB DECK FLIBGEN

```
FLIBGEN,CP70,T1000,MT01.
COMMENT. *SVDV,N7155,36R3D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO *
COMMENT. *                  SCOPE 2.0 INTEGRATION ARH248            *
COMMENT. *                  4201 LEXINGTON AVE. N.                  *
COMMENT. *                  ST PAUL, MINN.    55112                 *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * USING THE 3.4 RELEASE TAPE FOR THE FORTRAN COMMON OBJECT LIBRARY  *
COMMENT. * (FCL) OR AN UPDATED VERSION OF THAT TAPE PRODUCED BY FTNLIB7,     *
COMMENT. * FTNLIBX OR THE 6000 SCOPE 3.4 FCL MAINTAINENCE DECK , GENERATE   *
COMMENT. * THE FORTRAN SYSTEM LIBRARY (FORTRAN EXTENDED OBJECT LIBRARY)     *
COMMENT. *                                                          *
COMMENT. ************************************************************
FILE(OLDPL,RT=S)
LABEL(OLDPL,L=$FTNLIBS*3P4$)
STAGE(OLDPL,MT,HI,E,PRE,
VSN=CY51)
REWIND(OLDPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE OLDPL ADDING CYCLE INDICATOR AND PLACE SOURCE FOR FORTRAN  *
COMMENT. * OBJECT LIBRARY ON FTNSRC                                  *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P,F,C=FTNSRC)
UNLOAD(OLDPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * ASSEMBLE FTNSRC  (COMPILE USING LATEST FTN COMPILER)      *
COMMENT. *                                                          *
COMMENT. ************************************************************
COMPASS(I=FTNSRC,B=FTNLIB,A,L=LIST)
ATTACH(PRDLIB,PRDLIB,ID=PRDLIB,
```

```
JOB DECK  FLIBGEN

CY=51)
LIBRARY(PRDLIB)
FTN(I=FTNSRC,OPT=2,B=FTNLIB,A,L=LIST)
LIBRARY.
RETURN(FTNSRC)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF FORTRAN OBJECT TIME BINARIES         *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
REWIND(FTNLIB)
LIBEDT(M)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CATALOG FTNIOLIB (FORTRAN OBJECT LIBRARY)                            *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
CATALOG(FTNIO,FTNIOLIB,ID=PRDLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
RETURN(FTNIO)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
*IDENT CYCLNUM
*I MODLVL.2
 MODLVL   MICRO  1,, 51
 SUBLVL   MICRO  1,,
---EOS--- 7/8/9
LIBRARY(FTNIO,NEW=2000)
REPLACE(*,FTNLIB)
FINISH.
LISTLIB(*,FTNIO,N=1)
---EOI--- 6/7/8/9
```

10.3.3 RLIBGEN

JOB DECK RLIBGEN

```
RLIBGEN,CP70,T1000,MT01.
COMMENT. *SVDV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                  SCOPE 2.0 INTEGRATION ARH248            *
COMMENT. *                  4201 LEXINGTON AVE. N.                  *
COMMENT. *                  ST PAUL, MINN.    55112                 *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * USING THE 3.4 RELEASE TAPE FOR THE FORTRAN COMMON OBJECT LIBRARY  *
COMMENT. * (FCL) OR AN UPDATED VERSION OF THAT TAPE PRODUCED BY FTNLIB7,     *
COMMENT. * FTNLIBX OR THE 6000 SCOPE 3.4 FCL MAINTAINENCE DECK, GENERATE    *
COMMENT. * THE RUNLIB SYSTEM LIBRARY (FORTRAN RUN OBJECT LIBRARY)           *
COMMENT. *                                                          *
COMMENT. ************************************************************
FILE(OLDPL,RT=S)
LABEL(OLDPL,L=$FTNLIBS*3P4$)
STAGE(OLDPL,MT,HI,E,PRE,
VSN=CY51)
REWIND(OLDPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE OLDPL ADDING CYCLE INDICATOR AND PLACE SOURCE FOR RUNLIB  *
COMMENT. * OBJECT LIBRARY ON RUNSRC                                  *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P,F,C=RUNSRC)
UNLOAD(OLDPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * ASSEMBLE RUNSRC                                         *
COMMENT. *                                                          *
COMMENT. ************************************************************
COMPASS(I=RUNSRC,B=RUNLIB,A,L=LIST)
SKIPF(RUNSRC)
```

```
JOB DECK  RLIBGEN

COMPASS(I=RUNSRC,B=RUNLIB,A,L=LIST)
RETURN(RUNSRC)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF RUNLIB OBJECT TIME BINARIES          *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
REWIND(RUNLIB)
LIBEDT(M)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CATALOG RUNIOLIB (RUNLIB OBJECT LIBRARY)
COMMENT. *                                                                      *
COMMENT. ************************************************************************
CATALOG(RUNIO,RUNIOLIB,ID=PRDLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
RETURN(RUNIO)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
*IDENT CYCLNUM
*I MODLVL.2
 MODLVL    MICRO  1,, 51
 SUBLVL    MICRO  1,,
*D LIBMAC.70
 .CALL     EQU    1
---EOS--- 7/8/9
LIBRARY(RUNIO,NEW=2000)
REPLACE(*,RUNLIB)
FINISH.
LISTLIB(*,RUNIO,N=1)
---EOI--- 6/7/8/9
```

11.1 REQUIREMENTS

The following materials are necessary to install COBOL 1.0.

11.1.1 TAPES

COBREL	Described in part I, section 2.11

11.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (COBMOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (COBUSR job)
SRTLIB (ID=PRDLIB)	Necessary to satisfy externals in COBOL compiler

11.1.3 DECKS

COBGEN	Regenerates COBOL from COBREL and catalogs COBLEL and COBLIB7 (both jobs' ID=PRDLIB)
COBMOD	Creates a new COBREL from the old COBREL and MODPL, then generates the binary from the new COBREL and catalogs COBLEL and COBLIB7 (both jobs' ID=PRDLIB)
COBUSR	Creates a new COBREL from the old COBREL and USERMODPL, then generates the binary from the new COBREL and catalogs COBLEL and COBLIB7 (both jobs' ID=PRDLIB)
COBINS	Catalogs binary from COBREL as COBLEL (both jobs' ID=PRDLIB)

11.1.4 DEPENDENCIES

Sort/Merge must be installed before COBOL, or at least SRTLIB7 must be updated to the latest PSR summary level.

11.2 CONFIGURATION PARAMETERS

The following installation parameters are in common deck CBLTEXT, which also contains all of the COBOL assembly parameters.

Parameters	Released Default Value	Significance
IP. DOPT	0	If fatal errors are encountered in a COBOL compilation, COBOL aborts the LGO file only if the D parameter is or is not present on the COBOL control card, depending upon the value of this symbol. 0 Abort the LGO file only if the D parameter is specified. 1 Abort the LGO file only if the D parameter is not specified.
IP. LCMDF	100000_8	If less than IP. MNLCM is supplied for a COBOL compilation, COBOL raises the LCM field length to this value; cannot be set to less than IP. MNLCM.
IP. MNLCM	40000_8	Minimum LCM field length necessary for a COBOL compilation; cannot be set to less than 40000_8.
IP. MNSCM	40000_8	Minimum SCM field length necessary for a COBOL compilation; cannot be set to less than 40000_8.
IP. SCMDF	40000_8	If less than IP. MNSCM is supplied for a COBOL compilation, COBOL raises the SCM field length to this value; cannot be set to less than IP. MNSCM.

11.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

11.3.1 COBINS

JOB DECK COBINS

```
COBINS,CP70,MT01,T1000.
COMMENT. *SVDV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *           ( NO PSRS )                                    *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * INSTALL COBOL FROM RELEASE TAPE                          *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(COBREL,L=$COBREL*SCP2.0$)
STAGE(COBREL,MT,MY,E,PRE,
VSN=CY47)
REWIND(COBREL)
COPYP(COBREL,OLDPL)
COPYP(COBREL,COBOL)
COPYP(COBREL,COBOB)
REWIND(COBOL,COBOB)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF COBOL OVERLAY            *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG COBLEL (USED BY NPRDLIB)                        *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(COBLEL,COBLEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(COBLEL)
COMMENT. ************************************************************
```

```
JOB DECK  COBINS

COMMENT. *                                                                    *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF OBJECT TIME BINARIES               *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
LIBEDT(M)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * CATALOG COBLIB7 (COBOL OBJECT TIME LIBRARY)                        *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
CATALOG(COBLIB7,COBLIB7,ID=PRDLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(COBLIB7)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
LIBRARY(COBLEL,NEW=2000)
REPLACE(*,COBOL)
FINISH.
LISTLIB(*,COBLEL,N=1)
---EOS--- 7/8/9
LIBRARY(COBLIB7,NEW=2000)
REPLACE(*,COBOB)
FINISH.
LISTLIB(*,COBLIB7,N=1)
---EOI--- 6/7/8/9
```

11.3.2 COBMOD

JOB DECK COBMOD

```
COBMOD,CP70,I1000,MT01.
COMMENT. *SVDV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED                    *
COMMENT. *                                                                      *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA              *
COMMENT. *                                                                      *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)                 *
COMMENT. *                                                                      *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                                         *
COMMENT. *   STAGE HAS BEEN DEFINED                                             *
COMMENT. *                                                                      *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO             *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248                      *
COMMENT. *                    4201 LEXINGTON AVE. N.                            *
COMMENT. *                    ST PAUL, MINN.    55112                           *
COMMENT. *                                                                      *
COMMENT. *            ( NO PSRS )                                               *
COMMENT. ************************************************************************
COMMENT.
LABEL(COBREL,L=$COBREL*SCP2.0$)
STAGE(COBREL,MT,HY,E,PRF,
VSN=CY45)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                                  *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,8,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *   UPDATE COBOL OLDPL, CREATE SEQUENTIAL NEWPL                        *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
UPDATE(P=COBREL,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(COBREL)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED     *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL-  *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL. *
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN        *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).       *
COMMENT. *                                                                      *
```

```
JOB DECK  COBMOD

COMMENT. ************************************************************************
UPDATE(P=RANDPL,F,B,C=0,N,I=NULL)
RETURN(RANDPL)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *  UPDATE NEWPL INSERTING CURRENT CYCLE INDICATOR                      *
COMMENT. *  AND PRODUCE COMPILE FILE                                            *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
UPDATE(P=NEWPL,F,C)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * ASSEMBLE COBOL OBJECT TIME ROUTINES                                 *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
COMPASS(I,B=COBOB,A,L=LIST)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * ASSEMBLE COBOL COMPILER                                             *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
COMPASS(I,B=COBCP,A,L=LIST)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * BUILD OVERLAY ON FILE COBOL. SATISFY CERTAIN                         *
COMMENT. * EXTERNALS FROM SORT MERGE LIBRARY                                    *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
ATTACH(SRTLIBX,SRTLIB7,ID=PRDLIB)
LIBRARY(SRTLIBX)
LOAD(COBCP)
NOGO.
LIBRARY.
REWIND(COBOL,COBOB)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF COBOL OVERLAY                        *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
LIBEDT(M)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CATALOG COBLEL (USED BY NPRDLIB)                                    *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
CATALOG(COBLEL,COBLEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(COBLEL)
COMMENT. ************************************************************************
```

```
JOB DECK  COBMOD

COMMENT. *                                                                    *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF OBJECT TIME BINARIES               *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
LIBEDT(M)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * CATALOG COBLIB7 (COBOL OBJECT TIME LIBRARY)                        *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
CATALOG(COBLIB7,COBLIB7,ID=PRDLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(COBLIB7)
LABEL(NEWCOB,L=$COBREL*SCP2.0$)
STAGE(NEWCOB,MT,HY,N,POST,
VSN=CY47)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * CREATE NEW COBREL TAPE...FIRST PARTITION - NEWPL,                  *
COMMENT. * SECOND PARTITION - COBOL OVERLAY, THIRD PARTITION -                *
COMMENT. * COBOL OBJECT TIME BINARIES                                         *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
REWIND(NEWCOB,COBOL,COBCB)
COPYP(NEWPL,NEWCOB)
COPYP(COBOL,NEWCOB)
COPYP(COBOB,NEWCOB)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. *  TEST THE NEW TAPE VIA UPDATE                                      *
COMMENT. *                                                                    *
COMMENT. ********************************************************************
UNLOAD(NEWCOB)
STAGE(NEWCOB,MT,HY,E,PRF,
VSN=CY47)
REWIND(NEWCOB)
COPYP(NEWCOB,OLDPL)
COPYP(NEWCOB,NEWABS)
COPYP(NEWCOB,NEWREL)
UNLOAD(NEWCOB)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
RETURN(OLDPL,SCRAT,NEWABS,NEWREL)
PAUSE.  MP022 - TYPE X.GO TO CAT FILES FOR ABHOPS, BACK=TAPE
EXIT(C)
COMMENT. ********************************************************************
COMMENT. *                                                                    *
COMMENT. * RECATALOG COBPL, COBBIN AND CREATE COBREL BACKUP TAPE              **.*
COMMENT. *                                                                    *
COMMENT. ********************************************************************
```

```
JOB DECK  COMMOD

ATTACH(A,COBPL,PW=SCPCN,ID=S20CPLOPL,
CY=51)
PURGE(A)
EXIT(U)
RETURN(A)
ATTACH(A,COBBIN,ID=S20CPLOPL,PW=SCPCN,
CY=51)
PURGE(A)
EXIT(U)
RETURN(A)
CATALOG(NEWPL,COBPL,ID=S20CPLOPL,CN=SCPCN,MD=SCPMD,EX=SCPEX,
PW=SCPCN,
CY=51)
CATALOG(COBCP,COBBIN,ID=S20CPLOPL,CN=SCPCN,MD=SCPMD,EX=SCPEX,
PW=SCPCN,
CY=51)
LABEL(NEWCPY,L=$COBREL*SCP2.0$)
STAGE(NEWCPY,MT,HY,N,POST,
VSN=CY47)
REWIND(NEWPL,COBOL,COBOB,COBCP,NEWCPY)
COPYP(NEWPL,NEWCPY)
COPYP(COBOL,NEWCPY)
COPYP(COBOB,NEWCPY)
UNLOAD(NEWCPY)
STAGE(NEWCPY,MT,HY,E,PRE,
VSN=CY47)
REWIND(NEWCPY)
COPYP(NEWCPY,OLDPL)
COPYP(NEWCPY,NEWABS)
COPYP(NEWCPY,NEWREL)
UNLOAD(NEWCPY)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C COBPSP
---EOS--- 7/8/9
*ID CYCLENG
*D CY37.1
 EDITION  MICRO  1,,$CY47$
---EOS--- 7/8/9
LIBRARY(COBLEL,NEW=2000)
REPLACE(*,COBOL)
FINISH.
LISTLIB(*,COBLEL,N=1)
---EOS--- 7/8/9
LIBRARY(COBLIB7,NEW=2000)
REPLACE(*,COBOB)


FINISH.
LISTLIB(*,COBLIB7,N=1)
---EOI--- 6/7/8/9
```

12.1 REQUIREMENTS

The following materials are necessary to install Sort/Merge 1.0.

12.1.1 TAPES

SRTREL	Described in part I, section 2.12

12.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (SRTMOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (SRTUSR job)

12.1.3 DECKS

SRTGEN	Regenerates Sort/Merge binary from SRTREL and catalogs SRTLEL, SRTLIB7, and SRTMACS (all have ID=PRDLIB)
SRTMOD	Creates a new SRTREL from the old SRTREL and MODPL, then generates the binary from the new SRTREL and catalogs SRTLEL, SRTLIB7, and SRTMACS (all have ID=PRDLIB)
SRTUSR	Creates a new SRTREL from the old SRTREL and USERMODPL, then generates the binary from the new SRTREL and catalogs SRTLEL, SETLIB7, and SRTMACS (all have ID=PRDLIB)
SRTINS	Catalogs binary from SRTREL as SRTLEL, SRTLIB7, and SRTMACS (all have ID=PRDLIB)

12.1.4 DEPENDENCIES

None.

12.2 CONFIGURATION PARAMETERS

The following parameters are in deck SMSRTX and common deck SMCONCN.

Parameter	Released Default Value	Significance
IP. FLSD	40000B	The default field length (SCM) for the Sort/Merge directive version; refer to the IP. FLSM parameter.
IP. FLSM	14000B	The minimum field length (SCM) in which Sort/Merge executes for the directive call version. If the field length is less than IP. FLSM when the SORTMRG call is made, the default (IP. FLSD) field length is requested by Sort/Merge.
IP. IDMX	144B (100D)	The maximum number of names in the IDNAME and the SQNAME tables; that is, each table may contain this many names. IDNAME contains field (key) names and SQNAME contains sequence names. Attribute tables for these names are also set in length by this parameter.
IP. LCBM	25000B (10752D)	This is the IP. LCMB used if the user declares an LCMSB parameter in the range: $1 \leq$ LCMSB $\leq$ IP. LCBM This is the minimum IP. LCMB.
IP. LCMB	141520B (50000D)	Total LCM buffer area for record manager for all intermediate scratch files; that is, all files developed internally by Sort/Merge. The default (IP. LCMB) is used if the user does not declare the LCMSB parameter or if he declares LCMSB=0.
IP. ORDL	2	Minimum merge order for LCM resident intermediate merge files (sort strings).
IP. ORDM	6	Merge order; the number of files that are merged in one merge pass, although more files may be available for merging.
IP. PRGD	24000B	This is the default SCM used by the relocatable version of Sort/Merge; refer to the IP. PRGM parameter.
IP. PRGM	22000B	The minimum SCM required by the relocatable (macro callable) version of Sort/Merge. This parameter is the length of the Sort/Merge program plus a reasonable work area. The user may restrict available SCM within his field length by selecting the COMMON

Parameter	Released Default Value	Significance
		parameter within a Sort/Merge macro call sequence. Sort/Merge calculates available core with respect to the COMMON parameter. If the available SCM is less than this minimum, Sort/Merge requests a field length that provides for the default (IP.PRGD) SCM program and work area.

12.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

12.3.1 SRTINS

JOB DECK SRTINS

```
SRTINS,CP70,MT01,T1000.
COMMENT. *SVDV,07155,36R30,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * INSTALL SORT MERGE FROM RELEASE TAPE                      *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(SRTREL,L=$SRTREL*SCP2.0$)
STAGE(SRTREL,MT,HY,E,PRF,
VSN=CY45)
REWIND(SRTREL)
COPYP(SRTREL,OLDPL)
COPYP(SRTREL,SORTOV)
COPYP(SRTREL,RELB)
COPYP(SRTREL,SRTMACS)
REWIND(SORTOV)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF SORT OVERLAY              *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG SRTLEL (USED BY NPRDLIB)                         *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(SRTLEL,SRTLEL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
UNLOAD(SRTLEL)
```

```
JOB DECK  SRTINS

COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CREATE LINEDT FORMAT LIBRARY OF SORT OBJECT TIME BINARIES            *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
REWIND(RELB)
LIBEDT(M)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CATALOG SRTLIB7 (SORT/MERGE OBJECT TIME LIBRARY)                     *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
CATALOG(SRTLIB7,SRTLIB7,ID=PRDLIB,CN=SCPCN,MD=SCPMD,EX=SCPEX,PW=SCPCN,
CY=51)
UNLOAD(SRTLIB7)
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. * CATALOG SRTMACS                                                      *
COMMENT. *                                                                      *
COMMENT. ************************************************************************
CATALOG(SRTMACS,SRTMACS,ID=PRDLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
LIBRARY(SRTLEL,NEW=2000)
REPLACE(*,SORTOV)
FINISH.
LISTLIB(*,SRTLEL,N=1)
---EOS--- 7/8/9
LIBRARY(SRTLIB7,NEW=2000)
REPLACE(*,RELB)
FINISH.
LISTLIB(*,SRTLIB7,N=1)
---EOI--- 6///8/9
```

12.3.2 SRTMOD

JOB DECK SRTMOD

```
SRTMOD,CP7J,T1000,MT01.
COMMENT. *SVDV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                             *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO  *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
LABEL(SRTREL,L=%SRTREL*SCP2.0%)
STAGE(SRTREL,MT,HY,E,PRF,
VSN=CY45)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                       *
COMMENT. *                                                          *
COMMENT. ************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,B,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   UPDATE SORT/MERGE OLDPL, CREATE SEQUENTIAL NEWPL       *
COMMENT. *   AND COMPILE FILE FOR SORT OVERLAY                      *
COMMENT. *                                                          *
COMMENT. ************************************************************
UPDATE(P=SRTREL,F,C=0,N=RANDPL,I=UPIN)
UNLOAD(SRTREL)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED  *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL- *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.*
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN     *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).     *
```

```
JOB DECK  SRTMOD

COMMENT. *                                                                    *
COMMENT. ******************************************************************
UPDATE(P=RANDPL,F,M,N,C=OVLCOMP,I=NULL)
RETURN(RANDPL)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
COMMENT. *   PRODUCE A RELOCATABLE SORT COMPILE FILE FROM NEWPL             *
COMMENT. *                                                                  *
COMMENT. ******************************************************************
UPDATE(Q,P=NEWPL,C=RELCOMP)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
COMMENT. * ASSEMBLE SORT OVERLAY                                           *
COMMENT. *                                                                  *
COMMENT. ******************************************************************
COMPASS(I=OVLCOMP,B=OVLB,A,L=LIST)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
COMMENT. * ASSEMBLE  SORT OBJECT TIME ROUTINES                             *
COMMENT. *                                                                  *
COMMENT. ******************************************************************
COMPASS(I=RELCOMP,B=RELB,A,L=LIST)
RETURN(RELCOMP)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
COMMENT. * PRODUCE SORT OVERLAY ON FILE SORTOV                             *
COMMENT. * SCOPE 2.0 LOADER WILL ISSUE DIAGNOSTIC ON THIS NOGO              *
COMMENT. * STATEMENT - LD214  (REFERENCE TO UNDEFINED BLOCK OS.COG)         *
COMMENT. *                                                                  *
COMMENT. ******************************************************************
LOAD(OVLB)
NOGO.
REWIND(SORTOV)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF SORT OVERLAY                     *
COMMENT. *                                                                  *
COMMENT. ******************************************************************
LIBEDT(M)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
COMMENT. * CATALOG SRTLFL (USED BY NPRDLIB)                                *
COMMENT. *                                                                  *
COMMENT. ******************************************************************
CATALOG(SRTLFL,SRTLFL,ID=SCPSCR,CM=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=5))
UNLOAD(SRTLFL)
COMMENT. ******************************************************************
COMMENT. *                                                                  *
```

```
JOB DECK  SRTMOD

COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF SORT OBJECT TIME BINARIES          *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
REWIND(RELB)
LIBEDT(M)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * CATALOG SRTLIB7 (SORT/MERGE OBJECT TIME LIBRARY)                   *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
CATALOG(SRTLIB7,SRTLIB7,ID=PRDLIB,CN=SCPCN,MD=SCPMD,EX=SCPEX,PW=SCPCN,
CY=51)
UNLOAD(SRTLIB7)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * CATALOG SRTMACS                                                    *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
COPYP(OVLCOMP,SRTMACS)
CATALOG(SRTMACS,SRTMACS,ID=PRDLIB,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * CREATE NEW SRTREL TAPE...FIRST PARTITION- NEWPL,                   *
COMMENT. * SECOND PARTITION- SORT OVERLAY, THIRD PARTITION-                   *
COMMENT. * OBJECT TIME BINARIES, FOURTH PARTITION- SORT MACROS               *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
LABEL(NEWSRT,L=$SRTREL*SCP2.0$)
STAGE(NEWSRT,MT,HY,N,POST,
VSN=CY45)
REWIND(NEWSRT,RELB,SORTOV,SRTMACS)
COPYP(NEWPL,NEWSRT)
COPYP(SORTOV,NEWSRT)
COPYP(RELB,NEWSRT)
COPYP(SRTMACS,NEWSRT)
COMMENT. ************************************************************************
COMMENT. *                                                                    *
COMMENT. * TEST THE NEW TAPE VIA UPDATE                                       *
COMMENT. *                                                                    *
COMMENT. ************************************************************************
UNLOAD(NEWSRT)
STAGE(NEWSRT,MT,HY,E,PRE,
VSN=CY45)
REWIND(NEWSRT)
COPYP(NEWSRT,OLDPL)
COPYP(NEWSRT,NEWABS)
COPYP(NEWSRT,NEWREL)
COPYP(NEWSRT,NEWMAC)
```

```
JOB DECK  SRTMOD

UNLOAD(NEWSRT)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
RETURN(OLDPL,SCRAT,NEWARS,NEWREL,NEWMAC)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C SRTPSR
---EOS--- 7/8/9
*/ COMPILE CARDS FOR RELOCATABLE SORT
*IDENT SMRELOC                                          1/7
*INSERT SMSRTX.2                                        2/7
SMRELOC                                                 3/7
*C SMARITH.SMSRTX                                       4/7
*C SMMACP,SMCOLP                                        5/7
*C SMEND.SMCNVT                                         6/7
*C SMKYCD,SMSRTC,SMIPNS,SMSRTM                          7/7
---EOS--- 7/8/9
LIBRARY(SRTLEL,NEW=2000)
REPLACE(*,SORTOV)
FINISH.
LISTLIB(*,SRTLEL,N=1)
---EOS--- 7/8/9
LIBRARY(SRTLIB7,NEW=2000)
REPLACE(*,RELB)
FINISH.
LISTLIB(*,SRTLIB7,N=1)
---EOJ--- 6/7/8/9
```

13.1 REQUIREMENTS

The following materials are necessary to install the Diagnostic Control Package.

13.1.1 TAPES

DIAREL	Described in part I, section 2.13

13.1.2 PERMANENT FILES

MODPL (ID=S20OPSMOD)	Necessary for application of PSR updates only (DIAMOD job)
USERMODPL (ID=S20OPSMOD)	Necessary for application of local customer modifications only (DIAUSR job)

13.1.3 DECKS

DIAGEN	Regenerates DCP binary from DIAREL and catalogs DIALEL (ID=PRDLIB)
DIAMOD	Creates a new DIAREL from the old DIAREL and MODPL, then generates the binary from the new DIAREL and catalogs DIALEL (ID=PRDLIB)
DIAUSR	Creates a new DIAREL from the old DIAREL and USERMODPL, then generates the binary from the new DIAREL and catalogs DIALEL (ID=PRDLIB)
DIAINS	Catalogs binary from DIAREL tape and catalogs DIALEL (ID=PRDLIB)

13.1.4 DEPENDENCIES

None.

13.2 CONFIGURATION PARAMETERS

Parameter	Value	Significance
INTEGM	0	Location that indicates if integer multiply is installed.

Parameter	Value	Significance
		If location is 0, there is no integer multiply; if it is not 0, integer multiply installed. Cross reference: CT73

13.3 REPRESENTATIVE DECKS

The following decks are included to indicate the format of the released decks. They are decks released with the initial release of SCOPE 2.0 and may be modified in subsequent modifications of SCOPE 2.0.

13.3.1 DIAINS

```
JOB DECK  DIAINS

DIAINS,CP7,T1000,MT01.
COMMENT. *SVOV,07155,3683D,SDMILLER
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED        *
COMMENT. *                                                          *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA   *
COMMENT. *                                                          *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)     *
COMMENT. *                                                          *
COMMENT. *   STAGE HAS BEEN DEFINED                                 *
COMMENT. *                                                          *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248          *
COMMENT. *                    4201 LEXINGTON AVE. N.                *
COMMENT. *                    ST PAUL, MINN.    55112               *
COMMENT. *                                                          *
COMMENT. *            ( NO PSRS )                                   *
COMMENT. ************************************************************
COMMENT.
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * INSTALL DIAGNOSTIC LIBRARY FROM RELEASE TAPE             *
COMMENT. *                                                          *
COMMENT. ************************************************************
LABEL(DIAREL,L=$DIAREL*SCP2.0$)
STAGE(DIAREL,MT,HY,E,PRE,
VSN=CY51)
REWIND(DIAREL)
COPYP(DIAREL,OLDPL)
COPYP(DIAREL,DCP)
REWIND(DCP)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF DCP OVERLAY              *
COMMENT. *                                                          *
COMMENT. ************************************************************
LIBEDT(M)
COMMENT. ************************************************************
COMMENT. *                                                          *
COMMENT. * CATALOG DIALFL (USED BY NPRDLIB)                        *
COMMENT. *                                                          *
COMMENT. ************************************************************
CATALOG(DIALFL,DIALFL,ID=SCPSCR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
CY=NN)
UNLOAD(DIALFL)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
```

```
JOB DECK  DIAINS


EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
LIBRARY(DIALEL,NEW)
REPLACE(*,DCP)
FINISH.
LISTLIB(*,DIALEL,N=1)
---EOI--- 6//7/8/9
```

13.3.2 DIAMOD

```
JOB DECK  DIAMOD

DIAMOD,CP70,T1000,MT01.
COMMENT. *SVUV,07155,36830,SDMILLER
COMMENT.
COMMENT. ************************************************************************
COMMENT. *                                                                      *
COMMENT. *   THIS IS A SCOPE2.0 INSTALLATION DECK AS CREATED                    *
COMMENT. *                                                                      *
COMMENT. *   BY  SCOPE2.0  INTEGRATION  - ARDEN  HILLS, MINNESOTA              *
COMMENT. *                                                                      *
COMMENT. *   FOR PSR SUMMARY LEVEL 51 (SEE SYSDECK DESCRIPTION)                 *
COMMENT. *                                                                      *
COMMENT. *   MODTAPEMD HAS BEEN DEFINED                                         *
COMMENT. *   STAGE HAS BEEN DEFINED                                             *
COMMENT. *                                                                      *
COMMENT. *   PLEASE SEND SUGGESTED CORRECTIONS AND MODIFICATIONS TO             *
COMMENT. *                    SCOPE 2.0 INTEGRATION ARH248                      *
COMMENT. *                    4201 LEXINGTON AVE. N.                            *
COMMENT. *                    ST PAUL, MINN.    55112                           *
COMMENT. *                                                                      *
COMMENT. *           ( NO PSRS )                                                *
COMMENT. ************************************************************************
COMMENT.
LABEL(DIAREL,L=$DIAREL*SCP2.0$)
STAGE(DIAREL,MT,HY,E,PRF,
VSN=CY47)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. * UPDATE MODPL TO GET LATEST CDC MODS                                     *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
ATTACH(MODPL,MODPL,ID=S200PSMOD,
CY=51)
UPDATE(P=MODPL,Q,D,8,*=/,C=UPIN)
RETURN(MODPL)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. * UPDATE DIAGNOSTIC OLDPL, CREATE COMPILE FILE                           *
COMMENT. * PRODUCE A SEQUENTIAL NEWPL                                             *
COMMENT. *                                                                         *
COMMENT. ***************************************************************************
UPDATE(P=DIAREL,C=0,F,N=RANDPL,I=UPIN)
UNLOAD(DIAREL)
COMMENT. ***************************************************************************
COMMENT. *                                                                         *
COMMENT. *  DUE TO A DEFICIENCY IN UPDATE, TO INSURE THAT THE NEWPL CREATED        *
COMMENT. *  IS VALID, THE OLDPL IS MODIFIED CREATING A RANDOM TEMPORARY NEWPL-     *
COMMENT. *  RANDPL - WHICH IS THEN UPDATED CREATING A SEQUENTIAL NEWPL - NEWPL.    *
COMMENT. *  FAILURE TO DO THIS COULD RESULT IN LOSS OF DECK NAMES AND AN           *
COMMENT. *  IMPROPER ORDERING OF DECKS (WHEN DECKS ARE ADDED OR PURGED).          *
```

```
JOB DECK  DIAMOD

COMMENT. *                                                                    *
COMMENT. ******************************************************************
UPDATE(P=RANDPL,C,N,W,F,I=NULL)
RETURN(RANDPL)
COMMENT. ******************************************************************
COMMENT. *                                                                    *
COMMENT. * PRODUCE DIAGNOSTIC RELOCATABLE BINARIES                           *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
COMPASS(I,B=DIAGB,L=LIST)
RETURN(COMPILE)
REWIND(DIAGB)
LOAD(DIAGB)
NOGO.
REWIND(DCP)
COMMENT. ******************************************************************
COMMENT. *                                                                    *
COMMENT. * CREATE LIBEDT FORMAT LIBRARY OF DCP OVERLAY                        *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
LIBEDT(M)
COMMENT. ******************************************************************
COMMENT. *                                                                    *
COMMENT. * CATALOG DIALFL (USED BY NPRDLIB)                                   *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
CATALOG(DIALFL,DIALFL,ID=SCPSUR,CN=SCPCN,EX=SCPEX,MD=SCPMD,PW=SCPCN,
CY=51)
CY=NN)
UNLOAD(DIALFL)
COMMENT. ******************************************************************
COMMENT. *                                                                    *
COMMENT. * CREATE NEW DIAREL TAPE...FIRST PARTITION= NEWPL                    *
COMMENT. * SECOND PARTITION - ABSOLUTE DIAGNOSTIC BINARIES                    *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
LABEL(NEWDIA,L=$DIAREL*SCP2.0$)
STAGE(NEWDIA,MT,HY,N,POST,
VSN=CY51)
REWIND(NEWDIA,DCP)
COPYP(NEWPL,NEWDIA)
COPYP(DCP,NEWDIA)
COMMENT. ******************************************************************
COMMENT. *                                                                    *
COMMENT. * TEST NEW TAPE VIA UPDATE                                          *
COMMENT. *                                                                    *
COMMENT. ******************************************************************
UNLOAD(NEWDIA)
STAGE(NEWDIA,MT,HY,E,PRE,
```

```
JOB DECK  DIAMOD

VSN=CYS1)
REWIND(NEWDIA)
COPYP(NEWDIA,OLDPL)
COPYP(NEWDIA,NEWBIN)
UNLOAD(NEWDIA)
UPDATE(P,N=SCRAT,W,C=0,I=NULL)
RETURN(OLDPL,SCRAT,NEWBIN)
PAUSE.  MP000 - JOB COMPLETED SUCCESSFULLY... TYPE X.GO
EXIT.
PAUSE.  MP001 - JOB FAILED ... TYPE X.GO
---EOS--- 7/8/9
/C DIAPSR
---EOS--- 7/8/9
LIBRARY(DIALEL,NEW)
REPLACE(*,DCP)
FINISH.
LISTLIB(*,DIALEL,N=1)
---EOI--- 6/7/8/9
```

REQUIREMENTS 1

1.1 HARDWARE

1.1.1 MINIMUM CONFIGURATION

The minimum configuration that SCOPE 2.0 requires is:

One 7600 central computer system
One 7638 mass storage file
Six FLPPs
One MCU with card reader and CRT display
One 6000 station or one 7611-1 I/O Station

1.1.2 FCO LEVELS

The released version of SCOPE 2.0 and its product set were tested on a CDC® CYBER 70/Model 76 with the following FCOs installed.

Serial 7 (AA102A08)	Serial 12 (AA102A01)	
27240	24512	27817
28004	25648	31865
29050	25139	29720
27678	26669	
	27240	
	27008	
	28004	
	29030	
	26353	

1.2 MEMORY

Minimum memory requirements to use SCOPE 2.0 are:

32K words of SCM
256K words of LCM
8 million characters of mass storage

INSTALLATION MESSAGES 2

2.1 SCOPE 2.0 MAINTENANCE PACKAGE

The following messages may appear during the execution of SYSDECK jobs as comments on PAUSE cards.

Message	Significance	Action
MP000 JOB COMPLETED SUCCESSFULLY... TYPE X.GO	Job has completed successfully.	Note this fact on the SYSDECK worksheets. Type X.GO to obtain output.
MP001 JOB FAILED... TYPE X.GO	Job failed.	Note this fact on the SYSDECK worksheets. Type X.GO to obtain output. Determine and correct the cause of the error; rerun the job before proceeding with the next group of jobs.
MP002 NEW FILE CATALOGED SUCCESSFULLY... TYPE X.GO	A cycle of PRDLIB has been cataloged successfully.	NOTE this on worksheet. Type X.GO to continue.
MP003 OLD PRDLIB CYCLE EXISTS. TYPE X.GO TO REPLACE	The job OPRDLIB was run to create PRDLIB, but the file already exists.	Type X.GO to replace it; otherwise, type X.DROP.
MP004 NEW PRDLIB CYCLE EXISTS. TYPE X.GO TO REPLACE	The job NPRDLIB was run to create a new cycle of PRDLIB, but the cycle already was cataloged.	Type X.GO to replace it; otherwise, type X.DROP.
MP005 TYPE X.GO TO PURGE INACTIVE ID=PRDLIB FILES	The job DPIDPRD is running.	Type X.GO to begin a MO=3 DUMPF of inactive files cataloged with ID=PRDLIB. This eliminates files which have been replaced in this build. Type X.DROP to skip this step.

Message	Significance	Action
MP006 TYPE X. RERUN FOR MORE COPIES ELSE X. GO	Pause at completion of DUMPF by ID jobs.	Type X. RERUN to rerun the job in order to produce another copy of the DUMPF tape. Type X. GO to terminate the job.
MP007 TYPE X. GO TO PURGE CERTAIN INACTIVE FILES	Pause for operator action.	Type X. GO to begin a MO=3 DUMPF of inactive files cataloged with various IDs. Type X. DROP to skip the step.
MP008 TYPE X. GO TO PURGE INACTIVE ID=S20OPSOPL	Pause for operator action.	Type X. GO to begin a MO=3 DUMPF of inactive files cataloged with ID=S20OPSOPL. TYPE X. DROP to skip that step.
MP009 TYPE X. DROP AT THE MTS	Pause in verification job for 7611-2 station.	Type X. DROP at the 7611-2 station for the job.
MP010 TYPE X. GO... VERIFICATION GOOD IF MTS	Message appears at 7611-2 display. Test is successful.	Type X. GO to continue.
MP011 SET SENSE SW. X. ONSW, NN. THEN X. GO		Type X. ONSW, NN. Type X. GO.
MP012 VERIFY SENSE SW. THEN X. GO.		Verify that sense switch set in response to MP011 is on. Type X. GO to continue.

Message	Significance	Action
MP013 SET SENSE SW. X. OFFSW, NN. THEN X. GO		Type X. OFFSW, NN. to turn off the sense switch set in response to MP011. Type X. GO to continue.
MP014 VERIFY SENSE SW OFF. THEN X. GO		Verify that sense switch set in response to MP013 is off. Type X. GO to continue.
MP015 J-DISPLAY PLEASE. THEN X. GO		Bring up J display for job. Type X. GO to continue.
MP016 CHANGE JOB PRIORITY X. ENPR, NNNN. THEN X. GO		Type X. ENPR, nnnn. to change job priority (nnnn is new priority). Type X. GO to continue.
MP017 VERIFY NEW PRIORITY PLEASE. THEN X. GO		Verify that priority set in response to MP016 is correct. Type X. GO to continue.
MP018 CHANGE JOB TIME LIMIT X. ENTL, XXXX. THEN X. GO		Type X. ENTL, xxxx to change time limit (xxxx=new time limit); value of 20 is suggested. Type X. GO to continue.
MP019 VERIFY CARDS PUNCHED AT COMPLETION. X. GO	Job punches cards.	Verify that cards are punched. Type X. GO to continue.
MP020 ENTER - X. COMMENT. VERIFIED. THEN X. GO		Type X. COMMENT. Inspect J display to verify that message is displayed. Type X. GO to continue.
MP021 PLEASE RERUN JOB. X. RERUN.		Type X. RERUN. Verify that job is rerun.

Message	Significance	Action
MP022 TYPE X. GO TO CAT FILES FOR ARHOPS, BACK-TAPE	Message is only for CDC use.	

The following messages are from LISTER, the program that formats the DECK, MEMO, and WORKSHEETS.

MP101 NO PARAMETERS	LISTER was called with no parameters.	Specify at least CY and either DECK or MEMO.
MP102 BAD PUNCTUATION	Invalid punctuation.	Refer to MP120.
MP103 UNKNOWN PARAMETER	Keyword encountered which is not known to LISTER.	Refer to MP120.
MP104 VALUE OUT OF RANGE	The value of a parameter = value is either too high or too low. The following ranges are in effect.	Correct the value and rerun. Refer also to MP120.

Keyword	Usage	Low	High
CY	Cycle Number	1	200
PS	Page Size	30	60
TD	Title Depth	2	10
CD	Content Depth	3	10

MP105 NULL FIELD	Either a keyword or a parameter of a keyword is null. This is not legal.	Correct and rerun. Refer also to MP120.
MP106 CONFLICT IN PARAMETERS	Both DECK and MEMO were specified on the same call to LISTER.	Remove one and rerun. Refer also to MP120.
MP107 NEITHER DECK OR MEMO SPECIFIED		Specify input type and rerun. Refer also to MP120.

Message	Significance	Action
MP110 INVALID DECK SET	Unknown value of SET keyboard.	Refer to MP120.
MP111 NO CYCLE SPECIFIED	CY parameter is required for all calls to LISTER.	Refer to MP120.
MP120 ERROR IN PARAMETER XXX	This message is printed after all MP101 through MP111 messages. The value xxx indicates the keyword being processed at the time when the error was discovered. If xxx is ALL, the error concerns all of the parameters.	

2.2 SYSTEM MAINTENANCE MONITOR 3.0 (SMMB)

None.

2.3 7611-1 I/O STATION

Message	Significance
TAPE ERROR	There is an error on the deadstart tape.

2.4 7611-2 MAGNETIC TAPE STATION

The following tape error messages may appear during the 7611-2 Magnetic Tape Station tape deadstart process. To reinitialize the disk and read the deadstart tape from the beginning respond with:

Type GO
Press CR

Message	Significance
BAD TAPE FORMAT	The tape mounted is not in the correct format. (Record size may have exceeded 512 words.)
BLANK TAPE	There is no data written on the mounted tape.
RPE UNRECOVERED	The driver is unable to recover from a read parity error.
TAPE UNIT MALF	The tape unit and/or controller is not working properly.

2.5 6000 OR CYBER 70 SERIES SCOPE 3.4 STATION

None.

2.6 SCOPE 2.0

2.6.1 DEADSTART INITIALIZATION AND RECOVERY MESSAGES

If a fatal or nonfatal error condition occurs during deadstart of SCOPE 2.0, one of the following error messages is displayed at the MCU console. Continuation of the deadstart process depends upon the nature of the error.

Message	Significance	Action
DEADSTART EEA EXIT	Deadstart encountered an EEA condition. Possible problems may be: DS2 from the system library does not align with the DS1 from cards or disk, and a jump to an invalid address occurs. This could occur, for example, when a deadstart recovery is attempted with a system cycle different than the deadstart tape. DS2 from the system library is zero length indicating a problem during creation of the library.	To determine the cause of the error, examine the deadstart EEA exchange package at the SCM location T.EEAXPA (10460). If the PSD at location 10463 indicates an SCM or LCM parity error (in bits 46 and 47), the problem is a hardware problem. (An SCM or LCM parity error is also displayed.) If the problem is not a parity error, examine the P address (or the location before that address) from the EEA XP. These should normally indicate the cause of the crash.
DISK AREA EXHAUSTED	All the area on the designated system disk unit is allocated or flawed.	If recovery is attempted, do not recover the SFT (spooled files). If there are any permanent files, they can be recovered. To recover, clear the LCM locations 0-20000$_8$ using the KL command before attempting the recovery. When the request SYSTEM FILE TABLE RECOVERY Y/N is typed, reply with N. If the problem is still evident, a deadstart initialization is necessary.
DISK I/O ERROR Cchannel Uunit TRACKtrack SECTORsector	An irrecoverable I/O error occurred on the unit, track, and sectors specified.	Possible actions are: The CE could run diagnostics against the suspected disk area and/or punch a flaw card for the bad area and continue with deadstart initialization or deadstart recovery.

Message	Significance	Action
DS2 NOT IN LIBRARY	DS2 module cannot be located during a search of the system library directory.	Determine cause of fault, correct fault; reattempt deadstart initialization.
SIF LCM BUFFER RECOVERY ABORTED	Recovery of the SIF LCM buffers is aborted because the validation of pointers and tables in LCM indicated erroneous information.	Continue recovery by pressing carriage return.
ERROR	All deadstart reply entries are checked for format errors. When the reply is typed at the MCU console, this message appears at the end of the current display.	The format error is ignored; reenter the corrected reply.
ERROR TYPE-IN OR REREAD CARD TO RESUME READING, TYPE CARRIAGE RETURN	All deadstart reply entries are checked for format errors. When the reply is by card, the message appears with the current display.	Card reading stops. Type any number of entries; resume reading of cards by pressing CR.
INPUT FILE lfn NOT RECOVERABLE FILE IN TRANSIT CARRIAGE RETURN TO CONTINUE	Input file not completely transferred from station.	Press CR to continue deadstart; SFT entry is lost.
INPUT FILE lfn NOT RECOVERABLE ILLEGAL SFT ORDINAL CARRIAGE RETURN TO CONTINUE	SFT ordinal for this entry is invalid	SFT entry is lost; press CR to continue deadstart.
INPUT FILE lfn NOT RECOVERABLE INVALID FAT ADDRESS CARRIAGE RETURN TO CONTINUE	Disk address of FAT is invalid.	SFT entry is lost; continue deadstart by pressing CR.
INPUT FILE lfn NOT RECOVERABLE JOB NOT RE-RUNNABLE CARRIAGE RETURN TO CONTINUE	No rerunnable bit set for this input file.	SFT entry is lost; press CR to continue deadstart.

Message	Significance	Action
INPUT FILE lfn NOT RECOVERABLE NO STATION ID CARRIAGE RETURN TO CONTINUE	SFT entry has zero station ID.	SFT entry is lost; press CR to continue.
I/O ACTION TIME OUT	Indicates a deadstart hang. A tape or disk I/O device with which deadstart is attempting to communicate cannot accept I/O activity, such as not ready.	
INVALID RMS UNIT	When permanent files are being recovered, an illegal entry in the PFD or the FAT is found. This is usually a result of a disk area being inadvertently overwritten.	Dump the PFD from disk; redeadstart without permanent file recovery.
INVALID TIME DATE	The text of the time or date entry is invalid.	Press CR, then reenter date and time with correct text.
LABEL ERROR	Occurs during a deadstart initialization with permanent file recovery; a correct SCOPE 2.0 volume label group was not found at the specified address.	Deadstart initialize entering correct address of volume label group; or, deadstart initialize without permanent file recovery.
LIB BUFFERS EXCEEDED	The overlays and directories from the system library overflowed the number of library buffers allocated. The default number of library buffers is specified by IP. LIB or by a LIB card that may be included in the SCPSID deck.	If the problem occurs during a deadstart initialization, increase the specification on the LIB card by at least 108 buffers and deadstart initialize again. If the problem occurs during a deadstart recovery, specify the number of library buffers when the CHQ display occurs during the next recovery.

Message	Significance	Action
MORE THAN 8 RMS DEVICES	Occurs at deadstart initialization when an attempt is made to install more than eight mass storage units through the EST card. The system does not allow more than eight units.	Deadstart initialize assigning the allowed number of mass storage units with the EST replies.
NBR PFD PAGES CHANGE	Deadstart has been assembled with a systems text value for the number of PFD pages (NE. NP) that conflicts with the actual number of PFD pages or disk or the number indicated on a system disk.	Reassemble deadstart (ODS), and punch a new DS1 deck.
NO SYSTEM DEVICE	No system rotating mass storage device has been specified in the EST.	Specify system device and reattempt deadstart initialization or recovery.
OUTPUT FILE lfn NOT RECOVERABLE ILLEGAL SFT ORDINAL CARRIAGE RETURN TO CONTINUE	SFT ordinal for this entry is invalid.	SFT entry is lost; press CR to continue deadstart.
OUTPUT FILE lfn NOT RECOVERABLE INVALID FAT ADDRESS CARRIAGE RETURN TO CONTINUE	Disk address of FAT is invalid.	SFT entry is lost; continue deadstart by pressing CR.
OUTPUT FILE lfn NOT RECOVERABLE NO STATION ID CARRIAGE RETURN TO CONTINUE	SFT entry has zero station ID.	SFT entry is lost; press CR to continue.
OUTPUT FILE lfn NOT RECOVERABLE SPOOLED TO STATION CARRIAGE RETURN TO CONTINUE	Output file has already been transferred to station.	SFT entry is lost; press CR to continue deadstart.

Message	Significance	Action
PF CYCLE NOT FOUND	This situation occurs only during deadstart recovery; it indicates that the cycle of the system permanent file specified by the operator or by the default does not exist.	Repeat deadstart recovery using the correct cycle to the SYSTEM PFN AND CYCLE request. If subsequent attempts produce the error message, deadstart initialize.
PFD POINTERS FULL	This problem occurs only during a deadstart recovery from tape. The new system is to be entered in the next available entry of the PFD entry of the current or specified permanent file name; but all five cycle entries are full.	Repeat deadstart recovery and specify that the new system replace one of the five existing cycle entries.
SFT RECOVERY ABORTED	Recovery of the SFT was aborted because the validation of appropriate indicators, pointers, etc., indicate erroneous information. The SIF is lost.	If acceptable and no action is to be taken to get a copy of the SIF from dis, press CR.
SST LABEL NOT FOUND	The label operand specified cannot be located in the system text.	This is a fatal error indicating a faulty system library requiring recreation of the system library.
SST mod NOT FOUND	The specified SST module (OST) name cannot be located in the system library.	This is a fatal error indicating a faulty system library requiring recreation of the system library.
TAPE INPUT FAULTY	The driver transmitting the deadstart tape forwarded a status response indicating a tape error was detected, such as parity error. Deadstart automatically tries five times to have the tape contents transmitted correctly. If a fault still exists after the fifth attempt, this message is displayed.	Recreate the deadstart tape, and reattempt deadstart initialization.

Message	Significance	Action
TOO MANY FLAWS TK20	At least eight sectors must be available for the volume label group.	Reattempt deadstart initialization with fewer track 20 flaws.

2.6.2 HARDWARE ERROR PROCESSING MESSAGES

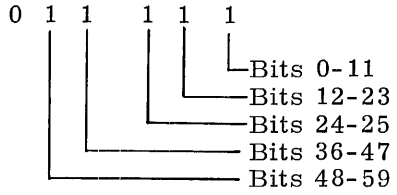
SC100
SCM TRANSIENT
PARITY address/bits

SCM parity error at specified location. The error does not recover upon rewriting and reading the specified location.

address — SCM parity error address

bits — Section error bits

When set, the bits indicate bad parity for the following portions of the SCM word.



If the address of the error is within the following areas, the system hangs.

1. 0 — 1777B
2. 10000B — beginning of user area (SCM address of File router job - FRJCB)

Restart the system by doing a deadstart recovery.

If the address of the error is within the field length of the currently active job, including the job supervisor, then parity error processing for the job is initiated. The job is either repressed or rerun (if possible).

If the address of the error is in the SCM I/O buffer area, 2000B - 7777B, or if none of the above, the error is ignored. No action to be taken.

SC101
LCM TRANSIENT
PARITY address/bits

LCM parity error at specified location. The parity error does not recover upon rewriting and reading the specified location.

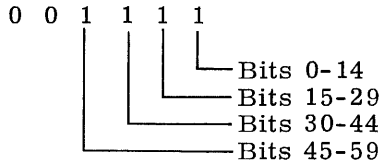
address — LCM address of parity error

bits — Section error bits

If the address of the error is less than the first system I/O buffer, the system hangs.

Restart the system by doing a deadstart recovery.

If the address is within the LCM field length of the currently active job, the parity error processing for the job is initiated.

Message	Significance	Action
	When set, the bits indicate bad parity for the portions of the LCM word:  	All other errors are ignored.
SC104 TAPE CHANNEL chan FLPP flpp ERROR status	chan CPU channel FLPP is driving flpp FLPP with error status FLPP status bits from SMM When set, the bits have the following meaning. Bit — Meaning 0 PPU has not been contacted by the MCU (SMM) in 77778 attempts (approximately 1 minute) 1 Parity error in stack 00 2 Parity error in stack 02 3 Parity error in stack 01 4 Parity error in stack 03 5 Error stop status 6 7638 disk driver (DSK) partner PPU time out 7 Illegal SMM call number Stack numbers are the lowest two bits of the address Then error processor automatically issues DP and CE commands to deadstart the FLPP and to clear parity. The operator must reload the FLPP program from cards as in the next column.	At the MCU card reader: Reload the FLPP program from cards using the following deck structure. Refer to part II, section 6.2.1 for a complete description of the deck. The EP cards are added. DPxx. KPxx. CExx. LPxx, 0. FLPP binary deck EPxx, 6, channel. EPxx, 7, channel. Initialize the flpp channel configuration by setting cells 6 and 7. xx flpp number channel numbers same as values on FC parameter on CHQ card (part II, section 6.3.3) RUxx, 100. 7_{8_9} At the system operator station: Turn on the channel so that the system issues further requests to the FLPP. Type ONCH, chan.

Message	Significance	Action
SC105 DISK CHANNEL chan FLPP flpp ERROR status	chan CPU channel FLPP is driving flpp FLPP with error status FLPP status bits from SMM. Refer to SC104 status description The error processor automatically issues DP and CE commands to deadstart the FLPP and to clear parity for both the failing FLPP and its partner FLPP. The operator must reload the FLPP program from cards as in the next column.	At the MCU card reader: Reload the FLPP program from cards using the deck structure in the SC104 action column. Reload both the failing FLPP and its partner FLPP. At the system operator station: Turn on channel so that the system can issue further requests to the FLPP. Type ONCH, chan. Turn on disk unit(s) connected to channel. Type ON ord. The E display indicates ON/OFF status as well as the EST ordinal for all online equipment.
SC106 STATION CHANNEL chan FLPP flpp ERROR status	chan Channel FLPP is driving flpp FLPP with error status FLPP status bits from SMM; refer to SC104 status information The error processor automatically issues DP and CE commands to deadstart the FLPP and to clear parity. The operator must reload the FLPP program from cards as in the next column.	At the MCU card reader: Reload the FLPP program from cards using the deck structure in the SC104 action column. At the system operator station: CPU- FLPP communication automatically resumes. Log in all stations connected to that FLPP.
SC107 MCU ERROR status	Indicates a FLPP 0 (the MCU FLPP) error. status Error status bits from SMM	Reload and reinitiate SMM in the MCU. Enter the CN command with parameters as set in deadstart SCPSID deck. If the error reoccurs, call customer engineering.

Message	Significance	Action
SC110 SCM RECURRENT PARITY address/bits	Hard memory parity error. address SCM address of error bits Section error bits; refer to SC100 description	Call customer engineer.
SC111 LCM RECURRENT PARITY address/bits	Hard memory parity error. address LCM address of error bits Section error bits; refer to SC101 description	Call customer engineer.

2.7 COMPASS 2.0

None.

2.8 FORTRAN EXTENDED 2.0

None.

2.9 FORTRAN RUN 2.0

None.

2.10 FORTRAN OBJECT TIME ROUTINES

None.

2.11 COBOL 1.0

None.

2.12 SORT/MERGE 1.0

None.

2.13 DIAGNOSTIC CONTROL PROGRAM

None.

NOTES AND CAUTIONS 3

3.1 SCOPE 2.0 MAINTENANCE PACKAGE

None.

3.2 SYSTEM MAINTENANCE MONITOR 3.0 (SMMB)

3.2.1 SYSTEM HANG

If the system is hung in monitor mode, type HC (and press CR) before trying any of the following:

Displaying LCM (LAD, LBD, LAM, etc.)

Clearing LCM (KL)

Taking a 7000 dump of LCM

3.2.2 SMMB and SMMD

The SMM Version 3.0 Operator's Guide describes the SMMB and SMMD Version 3.0 MCU monitor programs in detail. Refer to this manual also for details concerning SMM operating procedures and the interface between SMM and SCOPE 2.0.

SMMB is the MCU monitor program released with SCOPE 2.0; it does not require that the 857 disk pack be installed at the MCU.

The SMMD MCU monitor program is released through customer engineering to be installed on the MCU 857 disk pack by the customer engineers at an installation site. The SMMD monitor has all SMMB monitor capabilities plus additional features. The customer can use any of these additional features available under SMMD in the following situations.

The use of SMMD is restricted to less than ten percent of the available 857 storage capability. The remaining ninety percent of the 857 storage is reserved for customer engineering use.

The use of SMMD is for improved reliability, availability, and maintainability of the hardware.

The use of SMMD does not result in the system being dependent on 857 availability. The system must at all times be capable of operating with only the features provided by SMMB.

3.2.3 CN COMMAND

Following is a detailed description of the CN command that is required in the DS1 deck. The format is:

CN 10730, 10460, 5344, 10540.

This command directs the MCU to monitor the MCU/CPU call block for CPU calls and to store FLPP and SCM/LCM status in the communication area. The parameters specify the locations of the communication area, the EEA exchange package, and LCM addresses required by the SMM SY command. The last two parameters are optional. Whenever ORE or ORL changes are made, check the CN parameters for accuracy.

10730 — MCOVL is the location of the MCU's SCM data block. This block has the tag T. MCOVL and contains the areas for LCM I/O buffers, SCM/LCM parity error recording, and PPU error recording. These areas start at T. MCOVL (10700) + 30 in the released system. Therefore, MCOVL=10730.

10460 — EEAXP is the address of the EEA exchange package. SMM sets this address into EEA in the LCM I/O exchange package that is used for LCM displays etc. The package tag is T. EEAXPA and is equal to 10460 in the released system.

5344 — Optional parameter; add to ensure the proper functioning of the SY command.

SYKEY is the address to which SMM transfers the keyboard line when processing a SY command. The value of SYKEY is: (P. LCMCDCP)+W. LMCKJ1). Therefore, in the released system:

```
(P. LMCDCP)=5320
W. LMCKJ1=    24
-----------------
      SYKEY=5344
```

SYKEY is an address in the LCM table; it may change if the fixed tables in ORL are altered in length. W. LMCKJ1 is defined in CTEXT MCL and should not change. Check PLMCDCP for a change if ORL is modified. Installation parameter changes may alter ORL or ORE.

10540 — Optional parameter; add to ensure the proper functioning of the SY command.

SYEXCH is the address of the MCU interrupt exchange package called when executing a SY command. The package tag is IH. MCU; its value in the released system is 10540. This package is copied to SCM0000 before interrupting the CPU during processing of the SY command.

MCOVL, EEAXP, and SYEXCH does not change unless the exchange package areas are moved in ORE.

3.2.4 MCU CARD READER NOTES

If the MCU card reader fails to read a deck and there is a deadstart message displayed on the MCU console, press the clear key (CLR) on the keyboard to read the cards.

If the MCU card reader fails to read a deck after an aborted deadstart, type the following entry at the MCU keyboard to allow SMM to read the cards.

Type EP 0 5 0
Press CR

If the MCU (or any other card reader) fails to read a deck for no apparent reason, check the panel inside the lower left door of the card reader for any circuit breakers that are improperly positioned.

3.3 7611-1 I/O STATION

To refresh displays, type the RDON command and press CR.

3.3.1 PUNCH DEADSTART AND DUMP CARDS

To punch a new STATDS card, type the following in STATION mode:

PUNCH SCDC P80B

To punch a new STADDS card, type the following in STATION mode:

PUNCH DSC P80B

To punch cards necessary to dump the 7611-1 station, type any of the following pertinent commands in STATION mode.

PUNCH DDUMP1 P80B	To punch bootstrap card for PPU 1 (PRINTER1) loader
PUNCH DDUMP2 P80B	To punch bootstrap card for PPU2 (PRINTER2) loader
PUNCH DDPRINT P80B	To punch dump program deck

These cards (and assembly listings) may also be generated with the SYSDECK jobs STADCK and STADMP.

3.3.2 DUMP CPU AT 7611-1 I/O STATION

Dump the contents of the CPU memory at the 7611-1 I/O Station using the following procedure.

1. Read the dump card deck into the system through the MCU card reader in AUTO mode. Directions to obtain this deck are in part IV, section 3.3.1.

2. Deadstart the 7611-1 I/O Station if it is not active (part II, section 3.2).

3. Deadstart PPU5

   Type DS 5
   Press CR

4. In STATION mode, type CPU

   Press CR

5. In CENTRAL mode, type ON
   Press CR

   This command causes the 7611-1 I/O Station to begin communicating with the CPU. The message DEAD DUMP PROGRAM RUNNING IN THE CPU is displayed under PPU5 on the station display console.

6. Enter the range of the dump with the following command.

   Type DUMP mem sa la fname
   Press CR

mem	S for SCM; L for LCM
sa	Starting address
la	Last address
fname	Pseudo file name used temporarily during the dump procedure

   The range of core dumps should be limited to 100K maximum per file. The size of the station mass storage disk and the expansion encountered in converting from binary to line printer code requires the 100K maximum per file limit. No more than 300K should be at the station at one time.

3.3.3 DUMP THE 7611-1 I/O STATION

1. Deadstart the 7611-1 station using the bootstrap card for the printer to which the station is to be dumped. (Refer to part IV, section 3.3.1 for directions to obtain the bootstrap card needed.) If a dump of one of the printer drivers is desired, use the other printer for dumps. Deadstart procedures are in part II, section 3.

2. Read the dump card deck at the 7611-1 card reader in AUTO mode.

3. With the selected printer in READY mode, press the SINGLE SPACE switch n times. n is the number of the 7611-1 PPU that is to be dumped. Do not press SINGLE SPACE at all for a dump of PPU0.

4. Press the PAGE EJECT switch to dump the PPU selected.

5. Repeat steps 3 and 4 for each PPU dump desired.

6. Change the printer status to not ready to terminate a PPU dump in progress; press PAGE EJECT several times to remove the dump from the printer after taking dumps.

   The format of the dump is 9 columns wide. The first digit of the first column is the PPU number; the next 4 digits are the address of the first location of that row. The next 8 columns are the contents of the locations beginning with the address in column 1.

3.4 7611–2 MAGNETIC TAPE STATION

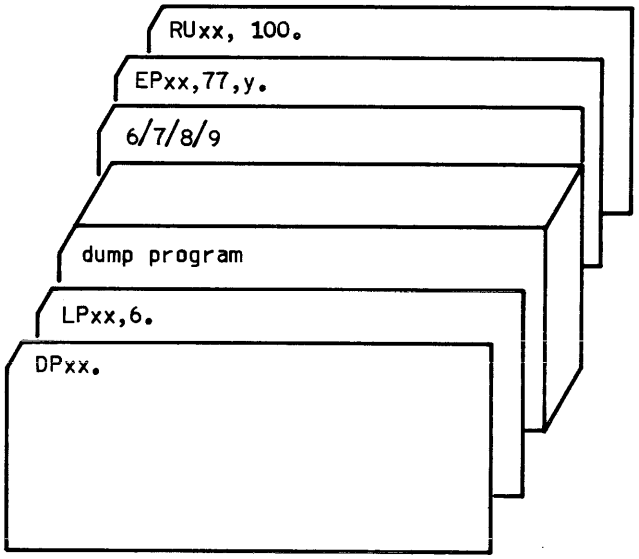
3.4.1 DUMP 7611-2 MAGNETIC TAPE STATION

The 7611-2 station dump program transfers the memory of the 7611-2 station PPU specified in the control cards into the FLPP so that it can then be dumped to the 6000 station or the 7611-1 station.

1. Punch the dump program from the MTSOPL using the following deck.

```
MTS, CM57000, T500, CP76.
COMMENT.
STAGE(MTSOPL, MT, PRE, HI, VSN=MTSOPL, ST=MTS)
REWIND(MTSOPL)
UPDATE(P=MTSOPL, Q)
REWIND(COMPILE)
COMPASS(I=COMPILE, B=PUNCHB, D)
7/8/9
*IDENT MODS
    installation changes
*COMPILE DMP
6/7/8/9
```

2. At the MCU card reader, read the following deck.



xx FLPP number
y Number of PPU that is to be dumped (0 to 5)

3. Dump the FLPP for each station PPU desired, using the dump procedures in the 7611-2 Magnetic Tape Station Operator's Guide.

   To dump the entire station, dump the FLPP first, then the XPP (PPU0), and then the other PPUs in any order.

3.4.2 UNIT ASSIGNMENTS

Unit assignments vary with assignments noted as follows.

Two 7629-2 controllers:

The first controller is on drivers 1 and 2 with logical units 0 through 7.
The second controller is on drivers 3 and 4 with logical units 10 through 17.

One 7629-2 controller:

This controller is on drivers 1 and 2 with logical units 0 through 7.

One 7629-1 controller:

This controller is on driver 1 with logical units 0 through 7.

One 7629-2 controller and one 7629-1 controller:

The 7629-2 controller is on drivers 1 and 2 with logical units 0 through 7.
The 7629-1 controller is on driver 3 with logical units 10 through 17.

Two 7629-1 controllers:

The first controller is on driver 1 with logical units 0 through 7.
The second controller is on driver 4 with logical units 10 through 17.

One 7629-2 and two 7629-1 controllers:

The 7629-2 controller is on drivers 1 and 2 with logical units 0 through 7.
The first 7629-1 controller is on driver 3 with logical units 10 through 13.
The second 7629-1 controller is on driver 4 with logical units 14 through 17.

or

The first 7629-1 controller is on driver 1 with logical units 0 through 3.
The second 7629-1 controller is on driver 2 with logical units 4 through 7.
The 7629-2 controller is on drivers 3 and 4 with logical units 10 through 17.

Three 7629-1 controllers:

The first 7629-1 controller is on driver 1 with logical units 0 through 7.
The second 7629-1 controller is on driver 2 with logical units 10 through 13.
The third 7629-1 controller is on driver 3 with logical units 14 through 17.

3.5 6000 OR CYBER 70 SERIES SCOPE 3.4 STATION

3.5.1 6000 STATION FLPP OPERATION

Occasionally attempts to communicate between the 6000 station and SCOPE 2.0 may fail because of a problem with the 7683 satellite coupler. Usually the FLPP for the 6000 station will display one of the following messages (each described in detail in the SCOPE 3.4 Operator's Reference Manual) if a coupler problem occurs.

TOO MANY WORDS FROM COUPLER
NO INPUT WD FLAG FROM COUPLER DURING TRANSFER
REC FLAG RECVD DURING BLOCK INPUT FROM COUPLER
NO REC FLAG FROM COUPLER
WORD COUNT FROM STATION TOO LARGE
END XMT, OUT WD FLAG TO COUPLER WONT DROP

If one of the preceding messages is displayed at the MCU, press CR. The FLPP attempts to communicate with the 6000 station. If the message reappears:

1. At the MCU:

   Type HT ppu.
   Press CR

2. Drop the station; at the 6000 console:

   Type X.DROP
   Press CR

   Type X.GO
   Press CR

3. Press MASTER CLEAR switch on the coupler (if the coupler has the software master clear FCO installed).

4. At the 6000 station:

   Type X.STATCCP.
   Press CR

5. AT the MCU:

   Type RUppu 102
   Press CR

If the situation persists with either no communication at all or the FLPP continuing to display the coupler error message, write an EOR on the coupler.

3.5.2 DUMP PROCEDURES

The detailed procedures to dump the SCM, LCM, and FLPPs are in the SCOPE 3.4 Station Operator's Reference Manual.

3.6 SCOPE 2.0

The SCOPE 2.0 operating system requires that SMMB be loaded and running in the MCU and that it be in CPU ON state in relation to the CPU (refer to the CN command in part IV, section 3.2.3). If this is not true, the SCOPE 2.0 operating system may hang, waiting for a SMM response, whenever any of the following conditions occur: SCM or LCM parity error; FLPP parity, error stop, or time-out error; or SMM request from on-line diagnostic overlays.

3.6.1 DUMP CONTENTS OF MASS STORAGE INTO LCM

During deadstart recovery (before typing the system source) the following command can be used to dump disk information into LCM starting at location 0.

The format of the command is:

DISK ^ cc ^ u ^ tttt, ss, no

cc	Two-digit channel number of disk
uu	One-digit unit number of disk
tttt	One to four-digit number of track to be dumped; 0 to 3777_8
ss	One to two-digit number of the first sector to be dumped; 0 to 47_8
no	One to two-digit number indicating total number of sectors to be dumped; 1 to 50_8
^	One space

Repeat this command until all necessary information is dumped. Each sector when dumped requires 1000_8 LCM locations.

Use any LCM station dump procedure to dump the contents of LCM.

3.6.2 PASSWRD CONTROL STATEMENT

The PASSWRD control statement enables a job with the proper system password to perform special privileged functions relating to activities such as system maintenance. The privileged status continues until the end of the job. Only one job can have privileged status at one time.

The format of the PASSWRD statement is:

```
PASSWRD (key)
```

key	Password key of 1 to 7 alphanumeric characters.

3.6.3 SYSLIBE CONTROL STATEMENT AND USE

Control Statement Format

The SYSLIBE control statement dynamically modifies the system libraries. A job using SYSLIBE to modify the libraries must have privileged status through use of the PASSWRD statement (part IV, section 3.6.2).

SYSLIBE adds or replaces a library in the system library table (SLT) for use by the relocatable loader in satisfying external symbols. The residence of the file being added or replaced is specified on the SYSLIBE control statement. The residence of an existing file in the SLT may be changed with SYSLIBE control statement.

The format of the SYSLIBE statement is:

```
SYSLIBE(libname=lfn, RES=r)
```

libname	Library name as it appears in the system library table (such as DBUGLIB, COBLIB, FTNLIB, and SRTLIB).
lfn	Name of a permanent file attached to the job as a local closed file written in library format and containing code for the library. The file must have read only permission. If no lfn is specified, the residence of the file already specified for the library is changed by the RES parameter.

r	Residence desired for the file	
	LCM	File is made resident in LCM
	RMS	File is resident in system mass storage

The job is aborted if the library name (libname) is not found in the system library table.

The job is aborted if the file specified by lfn is empty.

The file specified by lfn is no longer available to the user when the SYSLIBE function is completed. Only the loader can then access the file.

The specification of system libraries is not recovered by deadstart. Thus, the SYSLIBE statement must be executed after every deadstart initialization or recovery. The best way to accomplish this is to create a job that attaches the required files and executes a SYSLIBE statement for each of them. After an initial deadstart, the SYSLIB job (a sample of which is on SYSDECK) should be entered into the input queue, and given a priority of 7777B. Such a priority causes the job to return to the input queue with a priority of 0000B upon job termination. As long as the job has a priority of zero it does not begin execution. During a recovery this job is recovered and may be rerun by entering a priority of 7777B.

Use

To add system libraries to the SCOPE 2.0 system using the SYSLIBE procedure, use the following instructions.

1. Change or enter new libraries in the libraries section of OST. These symbols are defined with the TEQU macro. They must be in the same order and of the same length as the system library tables in ORL.

2. Insert the SLT entries in the libraries sections of ORL.

3. Reassemble the system.

The format of the SLT entry in ORL is:

```
T.SLTxxx    EQU     *
            VFD     56/nL libname
            VFD     4/1
            BSSZ    LE.SLT-1
```

The parameters are:

xxx	Three-character mnemonic for library
n	Length of name (in characters)
libname	Name of library

Sample ORL entry:

```
T.SLTCOB    EQU     *
            VFD     56/6L COBLIB
            VFD     4/1
            BSSZ    LE.SLT-1
```

Sample OST entry:

```
T.SLTCOB    TEQU        *,LE.SLT
```

Sample SYSLIBE control statement format:

```
ATTACH(COBLIB7,COBLIB7,ID=SCOPE 2)
SYSLIBE(COBLIB=COBLIB7,RES=RMS)
```

3.6.4 DUMPQ AND LOADQ CONTROL STATEMENTS

DUMPQ Description

DUMPQ is a SCOPE 2.0 utility that saves all or part of the input and/or output queue. The result of the DUMPQ operation is a W-format file called DPQFILE (described at the end of this section) that can be disposed in the following ways:

Staged tape	STAGE and/or LABEL statement or macro must precede DUMPQ control statement.
On-line tape	REQUEST and/or LABEL statement or macro must precede DUMPQ control statement.
Catalog	DPQFILE may be cataloged at either the 6000 station or at central.

If none of these are used, DPQFILE defaults to an unblocked mass storage file and is lost in job termination.

DPQFILE is always rewound when opened and is always closed and rewound at the conclusion of the dump.

All other files used by DUMPQ are opened and closed without rewinding.

Unless specified by a REQUEST statement, DPQFILE is assumed to have A0 allocation style and T0 transfer size.

DUMPQ Control Statement Format

```
DUMPQ(FT=type,ST=id,FC=forms,L=list,I=in,PF=pfn)
```

All parameters are optional and may appear in any order.

type — FT (file type) option that may have the following values.

omitted	Dump all files that fulfill the requirements of the other options.
I	Dump all input files.
O	Dump all output files; this includes all files of PR, P1, and P2 disposition.
PR	Dump only files of PR disposition.
P1	Dump only files of P1 disposition.

	P2	Dump only files of P2 disposition.
	P	Dump all punch files; this includes all the disposition mnemonics that follow.
	PU	Dump only files of PU disposition.
	PB	Dump only files of PB disposition.
	P8	Dump only files of P8 disposition.
	FR	Dump only files of FR disposition.
	FL	Dump only files of FL disposition.
	HR	Dump only files of HR disposition.
	HL	Dump only files of HL disposition.
	PT	Dump only files of PT disposition.

id — ST (station) option; may have the following values.

omitted	Dump files regardless of station id.
ggg(ttt)	Dump files from station ggg(ttt). ggg Station id ttt Terminal id

forms — FC (forms code) option; may have the following values.

omitted	Dump files regardless of forms code.
alphanum	Dump only those files of indicated forms code; alphanum is a maximum of 12 bits in length (2 display coded characters or from 1 to 4 octal digits).

Specification of the FC parameter overrides the FT specification unless FT=I, which is fatal.

list — L (list) option; may have the following values.

omitted	Full listing appears on file OUTPUT.
lfn_1	Full listing appears on file specified by lfn_1.
0	Partial listing appears on file OUTPUT.

The full list option includes the following information for each file dumped; file name, station id, file type, priority, forms code, allocation style, and file length. The partial list option contains only the file names dumped.

in — I (input) option; may have the following values.

omitted	Dump files regardless of file name.
I	Read file INPUT for list of file names to be selectively dumped from DPQFILE.
lfn_2	Read file lfn_2 for list of file names to be selectively dumped to DPQFILE.

The file names appearing on file INPUT or lfn_2 may have leading blanks, which are ignored. However, all blanks after the first nonblank character are converted to zeros. The file names may not exceed seven characters.

pfn — PFN (permanent file name) option; may have the following values.

omitted	If a system failure occurs during the DUMPQ, the result of the dump is lost.
pfn	A unique 1 to 7 character permanent file name. A permanent file is created and expanded as each file in an I/O queue is dumped.

DUMPQ Operating Procedures

When dumping input queues after DUMPQ begins execution, type JCB, 0. This ensures that input files are not executed during the dump.

To dump output queues, either log out the station to which the output files are to be sent or use the OFFST command to eliminate sending output files to the station.

When DUMPQ cannot find files to be dumped, the message DUMP - LOOKING FOR FILES - TYPE DROP TO DISCONTINUE appears. If the dump is complete, drop the DUMPQ job using the DROP statement. The job proceeds to an EXIT statement, if one is present, for further processing of DPQFILE.

The DUMPQ job should never be killed. This causes files that were to be dumped to remain in a locked status until another DUMPQ is attempted. That is, INPUT files are not executed and OUTPUT files are not spooled to the station.

LOADQ Description

LOADQ is a SCOPE 2.0 utility that restores all or part of the input and/or output queues. LOADQ requires as input a W-format file called DPQFILE.

DPQFILE is always rewound when opened and is always closed and rewound at the conclusion of the load. All other files used by LOADQ are opened and closed without rewind.

Unless specified in a REQUEST statement, DPQFILE is assumed to have A0 allocation style and T0 transfer.

LOADQ Control Statement Format

```
LOADQ(FT=type, ST=id, L=list, FC=forms, I=in)
```

All parameters are optional and may appear in any order.

type — FT (file type) option; may have the following values.

omitted	Load files regardless of file type.
I	Load all input files.
O	Load all output files; includes all files of PR, P1, and P2 disposition.
PR	Load only files of PR disposition.
P1	Load only files of P1 disposition.
P2	Load only files of P2 disposition.
P	Load all punch files; includes files whose disposition is one of the following.
PU	Load only files of PU disposition.
PB	Load only files of PB disposition.
P8	Load only files of P8 disposition.
FR	Load only files of FR disposition.
FL	Load only files of FL disposition.
HR	Load only files of HR disposition.
HL	Load only files of HL disposition.
PT	Load only files of PT disposition.

id — ST (station) option; may have the following values.

omitted	Load files regardless of station id.
ggg(ttt)	Dump files from station ggg(ttt) ggg Station id ttt Terminal id

list — L (list) option; may have the following values.

omitted	Full listing appears on file OUTPUT.
lfn_1	Full listing appears on file lfn_1.
0	Partial listing appears on file OUTPUT.

forms FC (forms code) option; may have the following values.

omitted	Load files regardless of forms code.
alphanum	Load only those files of indicated forms code; alphanum is a maximum of 12 bits in length (2 display coded characters or from 1 to 4 octal digits).

FC specification overrides the FT parameter.

in I (input) option; may have the following values.

omitted	Load files regardless of file name.
I	Read file INPUT for list of file names to be selectively loaded from DPQFILE.
lfn_2	Read file lfn_2 for list of file names to be selectively loaded from DPQFILE.

The file names appearing on file INPUT or lfn_2 may have leading blanks, which are ignored. However, all blanks after the first nonblank character are converted to zeros. The file names may not exceed seven characters.

DUMPQ Examples

The following is an example of a DUMPQ job that dumps the input files belonging to station ggg to an on-line tape.

```
SAVE,CM4000,T100,MT01.
REQUEST(DPQFILE,MT)
DUMPQ(FT=I,ST=GGG,L=0)
6/7/8/9
```

The following is an example of a DUMPQ job that dumps the output files that have P2 disposition and PK forms code belonging to station CDC. Following the dump, DPQFILE is cataloged at station CDC and at central.

```
SAVE,CM4000,T100.
DUMPQ(ST=CDC,FT=P2,FC=PK,L=LIST)
EXIT.
CATALOG(DPQFILE,OUTPUTQ,ST=CDC)
CATALOG(DPQFILE,OUTPUTQ)
6/7/8/9
```

LOADQ Examples

The following is an example of a LOADQ job that restores the input queue of station AAA from a staged tape.

```
RESTORE,CM4000,T100.
STAGE(DPQFILE,PRE,ST=AAA)
LOADQ(ST=AAA,FT=I)
6/7/8/9
```

The following is an example of a LOADQ job that selectively restores the output queue of station QED. DPQFILE is a blocked permanent file; FX0002L and FTNXOLG are the files to be loaded.

```
RESTORE,CM4000,T100.
FILE(DPQFILE,RT=W,BT=I)
ATTACH(DPQFILE,OUTPUTQ)
LOADQ(FT=O,ST=QED,L=LIST,I)
7/8/9
FX0002L
FTNXOLG
6/7/8/9
```

DUMPQ and LOADQ Examples

The following jobs save and restore both the SCOPE 2.0 input and output queues.

```
SAVE,CM4000,T100.                    dumps SCOPE 2.0 I/O queues
STAGE(DPQFILE,POST,VSN=DP1)
DUMPQ
6/7/8/9
RESTORE,CM4000,T100.                 restores SCOPE 2.0 I/O queues
STAGE(DPQFILE,PRE,VSN=DP1)
LOADQ.
6/7/8/9
```

DUMPQ and LOADQ Error Handling

When an error occurs while writing a file that is being dumped, DUMPQ pauses for operator intervention. If the operator chooses to continue, the file in error is discarded and dumping continues.

When an error occurs while reading DPQFILE, LOADQ determines if DPQFILE is W formatted and I blocked. If not, loading terminates at the point of the error. If so, LOADQ pauses for operator intervention. If the operator chooses to continue, LOADQ attempts to continue loading with the next file.

DUMPQ and LOADQ File Format

When a file is jumped to DPQFILE, the file is read with RT=U and written to DPQFILE with RT=W. Thus, the original W records are preserved and another set of W records is superimposed on all files dumped.

Each file dumped to DPQFILE is preceded by a W record that describes the file attributes. Following this header record is a set of W records of length 1000B that contains the data for the dumped file. Following the file data is an end-of-section and a trailing W record used for data verification.

Since DPQFILE consists of two levels of W records, the file itself cannot be directly printed. DPQFILE must be read by LOADQ, the output queue rebuilt, and the files sent to the station before proper results can be expected.

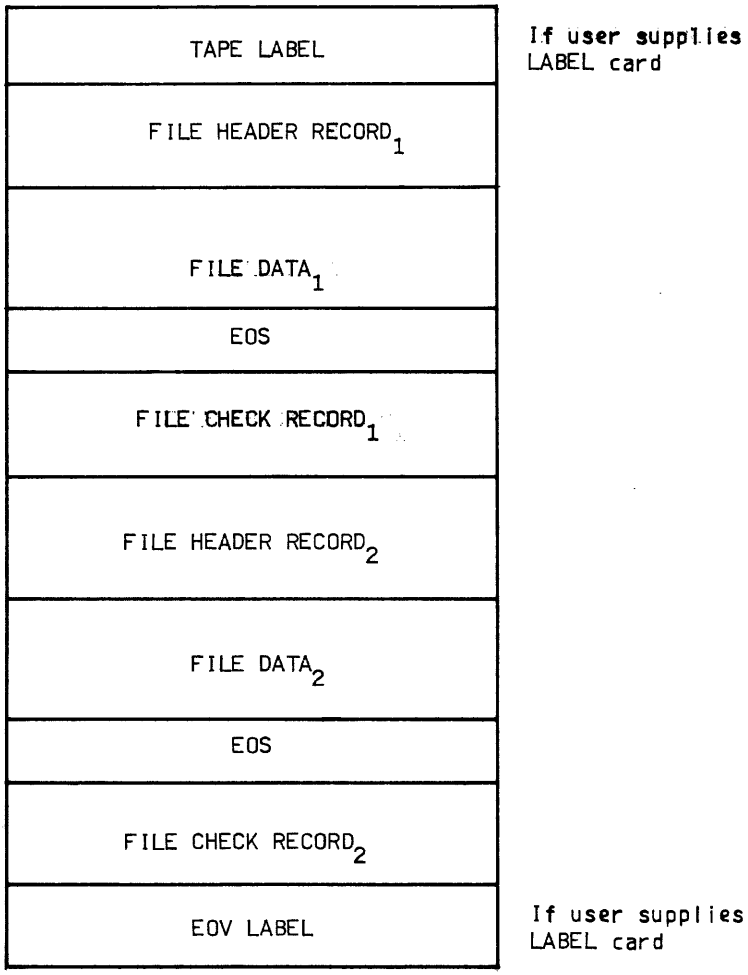


Figure IV-3.1. DUMPQ/LOADQ File Format

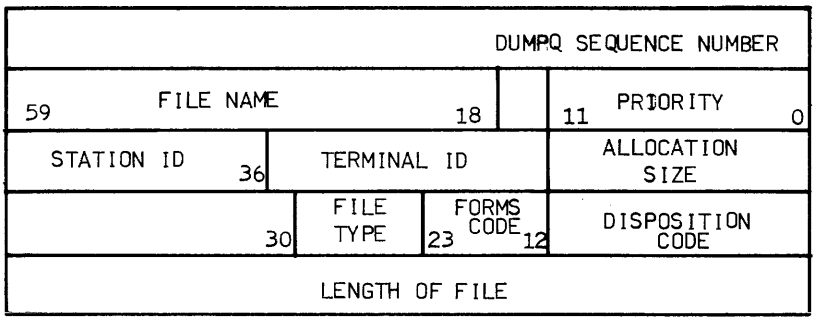


Figure IV-3.2. DUMPQ/LOADQ File Header Record Format

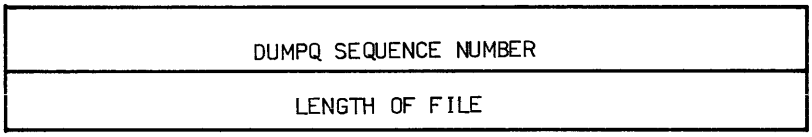


Figure IV-3.3. DUMPQ/LOADQ File Check Record Format

3.6.5 SYSTEM MASS STORAGE FORMAT

The following figure illustrates the contents (by track) of the system mass storage file device following system installation. Note that there are fixed areas in the beginning, middle, and end of the file that are reserved for preventive maintenance (diagnostic testing). The Permanent File Directory and Catalog (PFD and PFC) positions may vary and are located by position pointers in the volume label group. Likewise, the beginning of the system library file image may vary. The remainder of mass storage by sectors is indicated as allocatable through the Device Allocation Map or flawed (and not allocatable) through the Track Flaw Table.

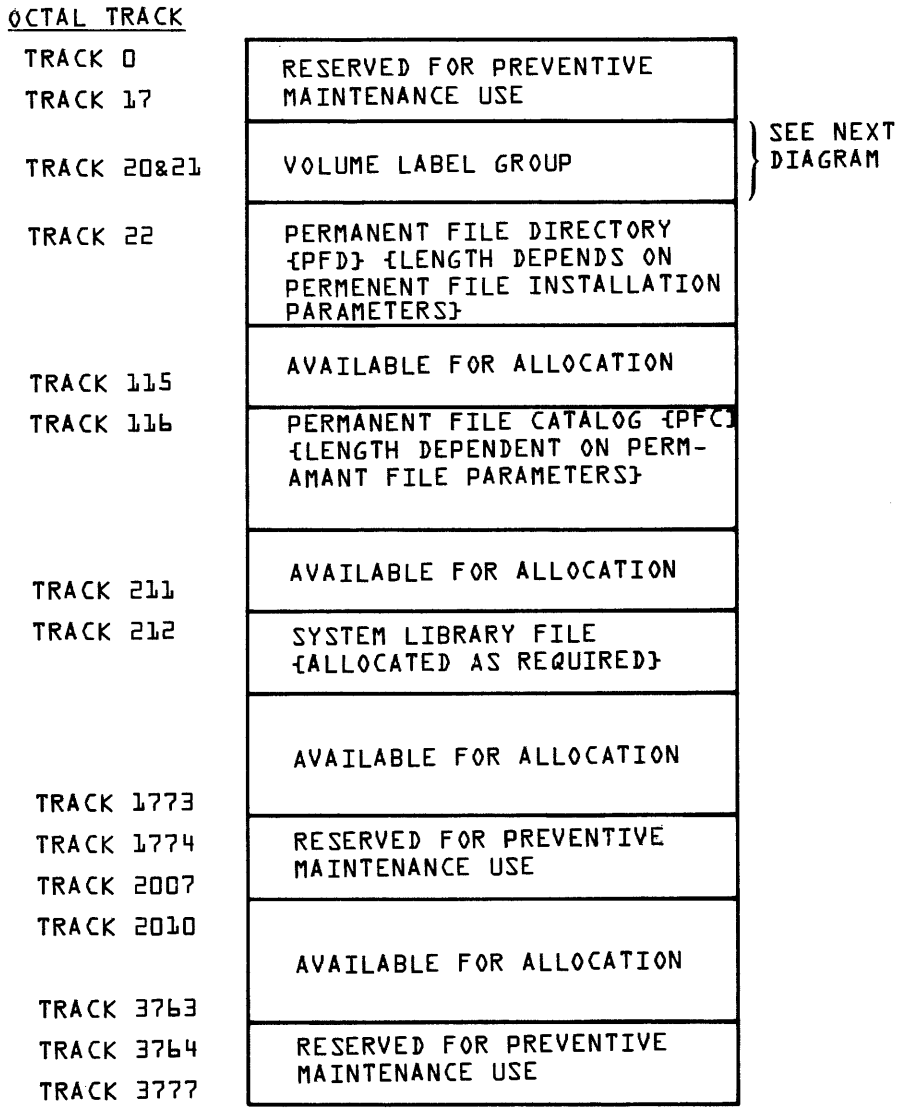


Figure IV-3.4. System Mass Storage File Allocation After Deadstart

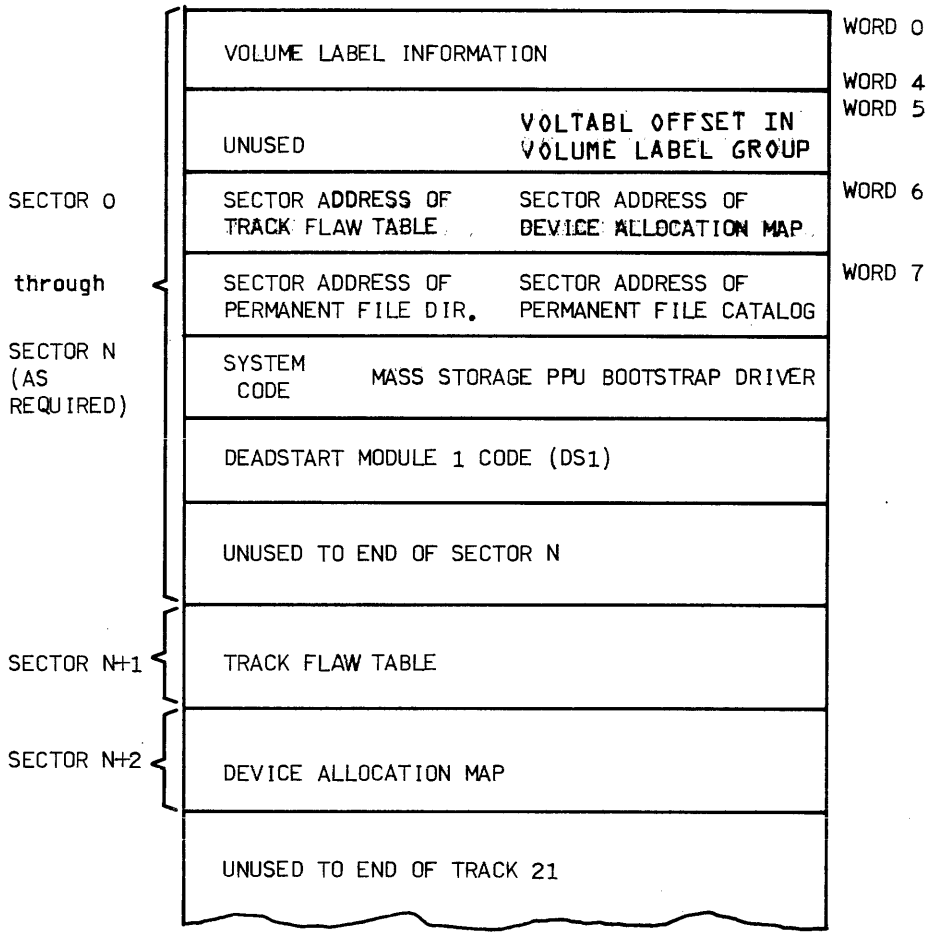


Figure IV-3.5. Volume Label Group (Track 20 and 21)

3.6.6 SYSTEM DEBUG AIDS

SCOPE 2.0 contains validity checks which are described as follows. However, they are not standard features and may be modified or deleted at any time if necessary.

The locations defined in LCM (ORL) that activate or deactivate the debug code can be set in any of the following ways:

By assembling ORL with the debug flag locations set when the system is generated.

By using a STORE command in the deadstart reply deck during deadstart initialization.

By typing a STORE command during deadstart recovery.

By manually setting the contents of the location from the MCU console with the EL command after deadstart.

SCOPE 2.0 validity checks are as follows.

1. The first L.ZERO (1000B) locations in LCM are defined as zero by the operating system (ORL). The job supervisor resident loader checks the first n words of LCM (n is specified by location T.LCMCHK) to ensure that no system overlay has written into low LCM. This check occurs before each job supervisor overlay is loaded for a CALL or GOTO; the check does not occur when a job supervisor overlay is reloaded following a RETURN. T.LCMCHK is described in part III, section 6.2.13.

2. All LCM buffers allocated to a job must subsequently be released by a job; otherwise, there is a buffer release problem that causes LCM buffers to be lost to the system. If the buffer check code detects a problem (such as a buffer allocated to a JCB and that job should not have buffers allocated) a system halt (crash) occurs through IH.DMP. The LCM location T.BUFCHK is a flag word indicating whether or not the buffer check code is to be executed. Refer to part III, section 6.2.13 for description of T.BUFCHK.

   T.BUFCHK also controls the execution of code to ensure that all buffers allocated to the file router are known to the file router, (that is, contained in PREs).

   If T.BUFCHK is set to 1 when deadstart completes, the file router buffer release code is activated. Setting T.BUFCHK to 0 inhibits execution of the file router buffer release checks. However, resetting T.BUFCHK to 1 when the system is executing does not activate the file router buffer release checks.

3. T.SYSABT in LCM specifies the conditions under which a system error results in a system halt (crash). In prereleases of SCOPE 2.0, the SCM location 13176B, (HALTFLG), was used in a similar way. T.SYSABT is described in part III, section 6.2.13.

4. A user program can dump the job supervisor LCM area by executing the DMPJSL control statement.

5. A user program can dump the JCB (job control block) and SFT (system file table entry) for the job by executing the DMPJT control statement. There are no parameters.

6. The deck queue manager (OE.DQM) halts the system on an illegal PRE (system table) chain.

7. The record manager (OS.CFL) halts the system on an illegal PRE (system table) address.

3.6.7 DESCRIPTION OF DEADSTART INITIALIZATION AND RECOVERY

SCOPE 2.0 deadstart and recovery are accomplished by one program. This program exists in the two modules DS1 and DS2. It defines the system equipment, initializes the central system and the FLPPs, and synchronizes the programs running in the system hardware. This section describes the functions of DS1 and DS2 during deadstart and recovery and is not part of the installation procedure.

DS1

The DS1 binary module is loaded into SCM by the SCPSID deck from the MCU card reader at system deadstart time, or it is loaded from the system mass storage device via a special bootstrap driver at system recovery time.

DS1 performs the following functions.

1. Configures SCOPE 2.0 according to system hardware.

2. Obtains the system library from tape (SCPLIB) that is mounted either on a 7611-1, 7611-2, 6000, or on-line tape unit (through a controller to a FLPP), with a common interface being used by DS1 to read the tape (tape format is that of a system library file copied to tape with W-format records and C blocking).

3. Establishes this system library as a permanent file on the system mass storage device and the system directory in LCM.

4. Locates the system library entry for DS2, reads DS2 from the system mass storage device to SCM, and passes control to DS2.

DS2

DS2 is a partition on the system library. After DS1 establishes the system library as a permanent file in system mass storage, it locates the system library entry for DS2, reads DS2 to SCM, and then passes control to DS2. The AUTO command module on the system library controls the execution of DS2.

DS2 performs the following functions during installation.

1. Interfaces with SMM (the MCU resident program) to interpret the replies in the deadstart reply deck (part of SCPSID deck) and any replies the operator enters through the MCU console

2. Constructs LCM and SCM resident programs

3. Establishes LCM system buffers

4. Builds overlay libraries and directories

5. Deadstarts and loads the FLPPs indicated by the CHQ card when applicable

6. Creates the device allocation maps (DAMs)

7. Creates or recovers (operator option) the permanent file directory (PFD), the permanent file catalog (PFC), and the track flaw tables for each mass storage device on the system disk

8. Formats and writes the volume label groups to system disk

9. Passes control to SCOPE 2.0

Recovery

The purpose of system recovery is to reestablish the operating system after system failure. Recovery of the system is accomplished by the two phases of execution DS1 and DS2. DS1 is given control after it is read from system mass storage by the special bootstrap driver supplied by SMM.

When the RSxx, addr, c, d command is typed, the system response is:

1. SMM transfers a system disk bootstrap driver from its own field length in the MCU to the FLPP specified by xx.

2. This bootstrap driver transfers the first sector of the deadstart volume label group from the disk address specified by addr to location 0 of the specified FLPP. It contains a recovery disk driver and part of DS1.

3. The bootstrap driver transfers control to the recovery disk driver program.

4. The recovery disk driver requests SMM to transfer the DS1 information from the FLPP to location 0 in SCM.

5. Then the recovery disk driver requests SMM to transfer the rest of the volume label group (which contains the rest of DS1 and the standard system disk driver) from system mass storage to SCM.

6. The deadstart disk driver requests SMM to drop a deadstart signal on the CPU. This initiates an exchange jump to location 0 which is the location of DS1.

During the recovery process, various displays may appear on the MCU console. These displays along with possible operator replies are listed in part II, section 6.3 in the order in which they occur. All replies must be entered through the MCU console during deadstart recovery.

Using these replies, one of up to five cycles of a system library permanent file may be indicated as the system to recover, or the system may be specified as residing on tape as during initialization. The system hardware may be defined as during deadstart. Permanent files and I/O queues may be reestablished at the option of the system operator.

The second part of recovery, DS2, functions the same as in deadstart.

Completion of system recovery leaves a running SCOPE 2.0 operating system in the CPU and FLPPs.

3.6.8 OVERLAY NAMING AND NUMBERING CONVENTIONS

To add an overlay to the system:

1. Find an available overlay number in the appropriate level, job supervisor, E1, E2, E3, IH.

2. In deck OST equate the overlay name to a five digit index having the following format.

x y zzz

zzz Overlay number; 0 through 777B

y Overlay level

Level Number	Level of System	Overlay Prefix
0	Job supervisor	OU., OS.
1	E1, executive level 1	OE.
2	E2, executive level 2	OE.
3	E3, executive level 3	OE.
7	Pseudo channel interrupt handler	OI.

x Entry point

0	Main overlay
1 through 7	Alternate entry points

OU. overlays can be called by both the user and system overlays. OS. overlays can be called by the system only. OU. overlay names must be the main overlay. A user cannot call an alternate entry point.

Overlays reserved for the user are as follows.

Job supervisor	740B through 777B
All others (E1, E2, E3, IH)	340B through 377B

3.6.9 RESTARTING THE ON-LINE TAPE FLPP

If an on-line tape job hangs the system because a unit drops ready status, it is possible to abort the job using the following procedure.

1. Ready the unit.

2. Restart the on-line tape FLPP at location 104B

   Type RUxx, 104 (xx is the FLPP number)
   Press CR

3.7 COMPASS 2.0

None.

3.8 FORTRAN EXTENDED 2.0

None.

3.9 FORTRAN RUN 2.0

None.

3.10 FORTRAN OBJECT TIME ROUTINES

None.

3.11 COBOL 1.0

None.

3.12 SORT/MERGE 1.0

None.

3.13 DIAGNOSTIC CONTROL PROGRAM

The job calling the diagnostic control program must have at least 10K of LCM. If more is allocated, DCP verifies the proper functioning of that LCM.

INDEX

COMMENT SHEET

MANUAL TITLE SCOPE 2.0 Installation Handbook

PUBLICATION NO. 60344000 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.

CUT ALONG LINE

AA3419 REV. 11/69 PRINTED IN U.S.A.

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

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Technical Publications Dept.
4201 North Lexington Avenue
Arden Hills, Minnesota 55112

CUT ALONG LINE

MD220

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.