CONTROL DATA
CORPORATION

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

SCOPE INSTALLATION HANDBOOK
MODELS 72, 73, 74 VERSION 3.4
6000 VERSION 3.4

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

REVISION RECORD

REVISION	DESCRIPTION
A	Original printing.
(8-28-72)	
B	Addition of a new section covering 241 Interactive Graphics System Version 2.1. Since section 28
(11-16-72)	contains only new material, revision bars appear only in the front matter.
C	This revision consists of corrections and additions to the original printing, but does not contain
(11-22-72)	changes necessitated by PSRs. Revision C is a complete reprint.

Publication No.
60307400

Additional copies of this manual may be obtained from the nearest Control Data Corporation sales office.

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Software Documentation
215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086

or use Comment Sheet in the back of this manual

PREFACE

Product testing was performed on SCOPE 3.4 as a unified product set. The new features of each product are discussed in appropriate sections of this document. General points of interest are:

The program libraries are numbered as follows.

		Section
PL1	SCOPE 3.4	1
PL2	COMPASS 3.0	2
PL3	RECORD MANAGER 1.0 (including IS 2.0, DA 1.0, and 8-bit Subroutines)	3,4,5,26
PL4	FORM 1.0	6
PL5	CE Diagnostics	7
PL6	Maintenance Tools	8
PL7	FORTRAN Extended 4.0 Compiler	9
PL8	FORTRAN Extended 4.0 Object Library	9
PL9	COBOL 4.0	10
PL10	SORT MERGE 4.0	11
PL11	QUERY UPDATE 1.0, QUIDDL 1.0	12,13
PL12	INTERCOM 4.1	14
PL13	PERT/TIME 1.2	15
PL14	SIMSCRIPT I.5 2.0	16
PL15	8231 IMPORT HS 1.0	17
PL16	APT 2.2	18
PL17	BASIC 2.0	19
PL18	ALGOL 3.0	20
PL19	SIMULA 1.0	21
PL20	1700 MSOS IMPORT HS 1.0	22
PL21	FORTRAN 2.3 (RUN)	23
--	1700 IMPORT HS 1.0	24
PL22	1700/274 IGS 2.0	25
PL23	MARS 2.1	27
PL24	241IGS 2.1	28

WARNING

The initial release of SCOPE 3.4 and product set supports 49K and larger central memory hardware configurations.

Installation of the full product set cannot be accomplished on 32K configurations and performance of that portion of the product set which can be installed is not guaranteed.

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undescribed parameters.

CONTENTS

PART I

PART II TABLE FORMATS

INTRODUCTION

1. Development and testing of SCOPE 3.4 was done on a variety of hardware configurations. The main configuration was a 131K 6600 which included the following major equipment. FCOs are listed in chronological order of installation:

<u>6613 Model D01</u>
20264B
20811
20252
21543
20738
20437
19964A
20880
20838
22888
21552
21553
23372B
24263A
25782
21828
26355
26379
24618
22095

<u>3624, Revision 23</u>
1
2
3
24 thru 30
33 thru 36
8954
9291
11471
13846
16445
19525
21538
21887
26865
19636

<u>6671 MUX</u>
19542
20250
20890
20930
22153
23853
21507

<u>6681 Model D01</u>
20922
22190
26904A
29677

<u>6684 (DC211) Model A01</u>
22994
26904A,02311B

<u>6639 Model B01/S.O. 10037 Model A03</u>
25311
26904

<u>3553 (FA705A) Model A02</u>
25999
26333
26574
26676

<u>6640 Model B01</u>
20742
21365
22366A
23426
23232
25408
24831
27751
22447A

While equipment designations and FCO levels are not provided, testing was also done on a 65K 6400 and selected CYBER models.

2. The full SCOPE 3.4 product set includes 11 system texts constructed from common decks included on the SCOPE and 6RM program libraries. Their source location and contents are shown below:

ACTCOM (SCOPE program library)
System action request macro prototypes

6RMCOM (Record Manager program library)
Record Manager user macro prototypes

CPSYS (SCOPE program library)
CPC IO macro prototypes

STATCOM (SCOPE program library)
7000 Station symbol definitions

IPARAMS (SCOPE program library)
Installation parameters

LMACOM (SCOPE program library)
Loader macro prototypes

PFCOM (SCOPE program library)
Permanent file macro prototypes

PPSYS (SCOPE program library)
SCOPE symbol definitions and PP macro prototypes

SCHCOM (SCOPE program library)
Integrated scheduler macro prototypes

RMCOM (Record Manager program library)
Record Manager internal macro prototypes

SISICOM (SCOPE program library)
SIS 1.0 macro prototypes

The following chart shows the combination of these common decks into the 11 system texts required for full utilization of the SCOPE 3.4 product set. All these texts are fixed in content except SYSTEXT; as released, SYSTEXT will contain 6RMCOM which contains IS 2.0 macro prototypes. At installation option, SYSTEXT may contain SISICOM and CPSYS in lieu of 6RMCOM.

The system texts are constructed as a part of the installation process discussed later in this document.

SCOPE 3.4 System Texts

Constituent COMDECKs

System Text Name	IPARAMS	STATCOM	PFCOM	LMACOM	SCHCOM	PPSYS	CPSYS	ACTCOM	6RMCOM	RMCOM	SISICOM
IPTEXT	X										
STATEXT		X									
PFMTEXT			X								
LDRTEXT				X							
SCHTEXT					X						
PPTEXT						X					
SCPTEXT						X	X	X			
TXT6RM								X		X	
IOTEXT								X	X		
CPCTEXT							X	X			X
SYSTEXT default								X	X		
SYSTEXT alternate							X	X			X

ALGTEXT and SMTEXT are installed when ALGOL and SORT/MERGE are installed. ALGTEXT is used only by ALGOL programs. COMPASS routines containing SORT/MERGE macros require specification of SMTEXT when assembled.

3. Representative times for the various provided installation/assembly decks are as follows when they are run on a 131K 6600 in conjunction with other batch processing:

DECK	Elapsed Minutes	CP Seconds
SCOPE1	60	1600
CMP1A	12	90
SIXRM1	12	163
IS1	15	215
DA1	7	60
BIT81	12	222
FORM1	8	166
CEDIAG1	15	223
SYMPL1	10	60
AIDS1	10	126
CMAINT	20	360
FTNLIB	8	140
COBOL1	40	630
SORT1	15	160
QU1	10	180
QUDDL1	10	80
INTCM1	55	800
PERT	5	45
SIMSC1	20	330
APT1	23	243
BASIC1	3	40
ALGOL1	8	180
SIMULA1	17	300
V23RUN1	10	150
MARS1	25	700

The above table does not include times for the Version 3.0 Object Library or any of the Editlib decks.

Installation time can be shortened perceptibly by utilizing the system's multiprogramming capability. Due to the interdependent relationships involving Record Manager and FORTRAN Extended, the jobs required for their installation must be run before any of their dependent products can be installed.

While not absolutely required for initial installation, maintenance mode installation involving introduction of PSR code dictates a prescribed order to ensure complete product set compatibility of relocatable binary and absolute binary overlays.

Each of the following steps contain one or more jobs that can be run concurrently; all jobs in a step must run to completion before the next step can begin:

A. Create FORTRAN Extended Compiler maintenance format tape using CINSTAL; assemble Record Manager using SIXRM1.

B. Replace Record Manager in the running system using SIXRM2; assemble FTN Compiler and FORM using CMAINT and FORM2.

C. Add FTN Compiler and FORM to the running system using FTNC2 and FORM2; assemble COBOL, SORT, and the 8-Bit Subroutines using COBOL1, SORT1, and BIT81.

D. Add COBOL, SORT, and the 8-Bit Subroutines to the running system using COBOL2, SORT2, and BIT82; assemble FTN Library and IS using FTN4LIB and IS1.

E. Add FTN Library and IS to the running system using LIB4 and IS2; assemble DA using DA1.

F. Add DA to the running system using DA2; assemble SCOPE and INTERCOM using SCOPE1 and INTCM1.

G. Add INTERCOM to the running system using INTCM2; assemble QU using QU1.

H. Add QU to the running system using QU2.

I. Create a configured deadstart tape using SCOPE2.

Deck AIDS1 must be run at step F or later but not before decks V23RUN1 and V23RUN2 have been run to install FORTRAN 23 (RUN). Deck AIDS2 may then be run at any convenient time.

Deck V3COB1 may be run at step C or later. Deck V3COB2 may then be run at any convenient time.

Deck SCOPE3 creates a configured deadstart tape from the running system. Execution of deck SCOPE3 at any time prior to step I will create an expanded though still unconfigured deadstart tape.

All other products may be installed at any point in the sequence outlined above.

4. All release tapes provided for SCOPE 3.4 and its product set are available in binary mode recorded in either 7- or 9-track format. The installation decks listed in this document and present on either the 7- or 9-track tapes are structured for use with the 7-track tapes.

A single tape is provided with the SCOPE release package which contains installation decks for the entire product set especially tailored for use at sites having only 9-track tape drives.

5. The installation decks provided create nine libraries. The following list defines contents of these libraries assuming all members of the SCOPE 3.4 product set are installed. The library names referenced should be considered reserved to CONTROL DATA CORPORATION.

The list is constructed in the order the products are installed for the Sunnyvale Development Center system. While listed here with all routines associated with each product shown in inclusive order, the PPLIB file of a deadstart tape is sorted in alphanumeric order by EDITLIB. The other deadstart tape library files are not sorted.

PPLIB CM Resident Library

Peripheral processor routines associated with SCOPE, INTERCOM, CE Diagnostics, Conversion Aids, and one PP routine associated with DA.

```
*************                    ***SCOPE***
*           *                       CED
* DEADSTART *                       TDR
*  RECORDS  *                       D
*           *                       DMT
*************                       CMR
***SCOPE***                         CCM
   CEA                              EST
***CE DIAGNOSTICS***                IRP
   CES                              5CP
   FC                               5CQ
   PM                               5CR
   EJT                              5CS
   MAP                              5CT
   CM6                              5CU
   MY1                              5CV
   ALS                              5CW
   FST                              P
   CT3                              STL
   CU1                              IRCP
   ALX                              MTR
                                    DSD
```

```
***********
*         *
* FPLIB   *
*         *
***********
*** SCOPE ***
    8X9   6000 RESIDENT OVERLAY
    8Y9   7000 STATION ROUTINE
    8XA   DSD CONSOLE COMMAND OVERLAY
    8XB   DSD CONSOLE COMMAND OVERLAY
    8XC   DSD CONSOLE COMMAND OVERLAY
    8XD   DSD CONSOLE COMMAND OVERLAY
    8XE   DSD CONSOLE COMMAND OVERLAY
    8XF   DSD CONSOLE COMMAND OVERLAY
    8XG   DSD CONSOLE COMMAND OVERLAY
    8XH   DSD CONSOLE COMMAND OVERLAY
    8XI   DSD CONSOLE COMMAND OVERLAY
    8XJ   DSD CONSOLE COMMAND OVERLAY
    8XK   DSD CONSOLE COMMAND OVERLAY
    8XL   DSD CONSOLE COMMAND OVERLAY
    8XM   DSD CONSOLE COMMAND OVERLAY
    8XN   DSD CONSOLE COMMAND OVERLAY
    8XO   DSD CONSOLE COMMAND OVERLAY
    8XP   DSD CONSOLE COMMAND OVERLAY
    8XQ   DSD CONSOLE COMMAND OVERLAY
    8XR   DSD CONSOLE COMMAND OVERLAY
    8YB   7000 STATION ROUTINE
    8YC   7000 STATION ROUTINE
    8YD   7000 STATION ROUTINE
    8YE   7000 STATION ROUTINE
    8YF   7000 STATION ROUTINE
    8YG   7000 STATION ROUTINE
    8YH   7000 STATION ROUTINE
    8YI   7000 STATION ROUTINE
    8DA   DSD DISPLAY DRIVER OVERLAY
    8DB   DSD DISPLAY DRIVER OVERLAY
    8DC   DSD DISPLAY DRIVER OVERLAY
    8DE   DSD DISPLAY DRIVER OVERLAY
    8DF   DSD DISPLAY DRIVER OVERLAY
    8DH   DSD DISPLAY DRIVER OVERLAY
    8DK   DSD DISPLAY DRIVER OVERLAY
    8DL   DSD DISPLAY DRIVER OVERLAY
    8DM   DSD DISPLAY DRIVER OVERLAY
    8DO   DSD DISPLAY DRIVER OVERLAY
    8DP   DSD DISPLAY DRIVER OVERLAY
    8DQ   DSD DISPLAY DRIVER OVERLAY
    8DR   DSD DISPLAY DRIVER OVERLAY
    8DS   DSD DISPLAY DRIVER OVERLAY
    8DT   DSD DISPLAY DRIVER OVERLAY
    8DX   DSD DISPLAY DRIVER OVERLAY
    8DY   DSD DISPLAY DRIVER OVERLAY
    8DZ   DSD DISPLAY DRIVER OVERLAY
    8EA   7000 STATION ROUTINE
    8EB   7000 STATION ROUTINE
    8EC   7000 STATION ROUTINE
    8EE   7000 STATION ROUTINE
    8EH   7000 STATION ROUTINE
    8EJ   7000 STATION ROUTINE
```

8EP	7000 STATION ROUTINE
8ES	7000 STATION ROUTINE
8EZ	7000 STATION ROUTINE
CIO	CIRCULAR INPUT/OUTPUT
MSG	ADD MESSAGE TO DAYFILE
1AJ	ADVANCE JOBS
7EC	GENERATE ECS BUFFERS
1PC	DROP PERMANENT FILE MASS STORAGE
PFC	PERMANENT FILE CATALOG
1PF	PERMANENT FILE QUEUE MANAGER
1FC	CATALOG FUNCTION PART II
PFA	PERMANENT FILE ATTACH
2FA	UTILITY I/L PROCESSOR PFA SEGMENT
LPF	LOAD PERMANENT FILES
PFP	PERMANENT FILE PURGE
PFE	PERMANENT FILE EXTEND
PFR	PERMANENT FILE RENAME
1FD	PFA DELAY OVERLAY
PFD	PERMANENT FILE DUMP
PFS	POSITION FUNCTION STORAGE ALLOCATION
DPF	DUMP PERMANENT FILES
TPF	TRANSFER PERMANENT FILES
1DU	DUMPF INITIALIZATION
TPT	TRANSFER PERMANENT FILE TABLES
1EJ	END OF JOB PROCESSOR
3DO	ALLOCATABLE DEVICE FILE OPEN
1S5	INTERFACE BETWEEN STACK PROCESSOR MGR. AND 1SP/3DO
1RN	RELEASE RECORD BLOCK CHAIN
4ES	ENTER STACK REQUESTS
1SP	STACK PROCESSOR MAIN PROGRAM
3SP	DRIVER OVERLAY FOR 6603-I
3SQ	DRIVER OVERLAY FOR 6638
3ST	DRIVER OVERLAY FOR 6603-II
3SR	DRIVER OVERLAY FOR 865
3SU	DRIVER OVERLAY FOR 814
3SV	DRIVER OVERLAY FOR 821
3SW	DRIVER OVERLAY FOR 841
3SS	DRIVER OVERLAY FOR 854
1EP	STACK PROCESSOR MAIN PROGRAM-ECS I/O BUFFERING
3EP	ECS DRIVER OVERLAY FOR 6603-I
3EQ	ECS DRIVER OVERLAY FOR 6638
3ET	ECS DRIVER OVERLAY FOR 6603-II
3ER	ECS DRIVER OVERLAY FOR 865
3EU	ECS DRIVER OVERLAY FOR 814
3EV	ECS DRIVER OVERLAY FOR 821
3EW	ECS DRIVER OVERLAY FOR 841
3ES	ECS DRIVER OVERLAY FOR 854
1DL	OVERLAY LOADER AND DAYFILE MESSAGE PROCESSOR FOR DSD
DIS	ROUTINE TO PROCESS CONTROL CARDS ENTERED AT CONSOLE
DMP	DUMP CM
1GM	DISPLAY GOOD MORNING
LOC	LOAD OCTAL CORRECTIONS
MEM	PROCESS MEMORY FUNCTION
REQ	REQUEST CARD PROCESSOR
3RQ	REQ OVERLAY CONTAINING 2TACOM
1RQ	REQ OVERLAY
2TA	TAPE ASSIGNMENT OVERLAY
ACE	CONTROL CARD READER

1IR	JANUS MAIN PROGRAM
1IS	JANUS MAIN PROGRAM
2IS	JANUS ROUTINE
1IQ	INITIATE JANUS CONTROL POINT
1IU	JANUS BACKSPACE PRINT NAME
1RP	END-OF-REEL SWAPPING
2RP	1RP OVERLAY
1CL	FILE CLOSE ROUTINE
2TC	CLOSE TAPE FILE
1TO	TAPE OPEN ROUTINE
6TO	OVERLAY TO 1TO
1OF	OPEN FILE ROUTINE (ALL FILES)
OFE	OPEN ROUTINE (DUMMY)
CLO	FILE CLOSE ROUTINE (DUMMY)
5DA	FAMILY DISK PACK LABEL PROCESSOR
1PK	SEQUENTIAL PACK CLOSE
3PK	SEQUENTIAL PACK INITIALIZATION
1DA	DISK PACK LABEL ROUTINE
EKG	FAMILY PACK END OF JOB PROCESSOR
SRB	EDITLIB ROUTINE TO COMPLETE DISK ADDRESS OF RECORD
1BT	TAPE/DISK BLANK LABELS
1DF	DUMP DAYFILE
1MF	MULTIFILE POSITION ROUTINE FOR ANSI LABELLED TAPES
1MH	DSD ≠MOTHER≠S HELPER≠
1DM	DEVICE QUEUE MANAGER
VSN	VISUAL SERIAL NUMBER
1LT	LOAD JOBS FROM TAPE
1MT	DRIVER FOR LONG RECORD STRANGER (L) TAPES FOR 7-TRACK TAPES
1P2	TAPE RECOVERY (WRITE DRIVER)
1P4	TAPE RECOVERY (POSITIONING DRIVER)
3PS	1P4 OVERLAY
1P3	TAPE RECOVER (VERIFICATION DRIVER)
3PO	1P3 OVERLAY
1P1	TAPE RECOVERY (POSITIONING DRIVER)
3PM	1P1 OVERLAY
1WI	SCOPE TAPE WRITE DRIVER FOR 7-TRACK TAPES
1RV	TAPE READ RECOVERY - INITIALIZATION
1NO	TAPE READ RECOVERY-NOISE RECORD-NOISE VERIFY
6NO	TAPE ERROR RECOVERY DEBUG SEGMENT
1N2	TAPE READ RECOVERY-NOISE RECORD-READ FORWARD 1
1N3	TAPE READ RECOVERY-NOISE RECORD-READ FORWARD 2
1R2	TAPE READ RECOVERY-TAPE PARITY ERROR REC 1
1R3	TAPE READ RECOVERY-TAPE PARITY ERROR REC 2
3R3	SEGMENT OF 1R3 LOADED WHEN GO OR DROP DECISION NECESSARY BY OPERATOR
1TF	FORWARD SKIP ROUTINE FOR TAPE
1WS	STRANGER (S) TAPE WRITE DRIVER
1NW	9-TRACK S-FORMAT TAPE WRITE DRIVER
8T3	SEGMENT FOR LOADING OF MMTC CONVERSION MEMORY
2TB	BACKWARD SKIP ROUTINE FOR TAPE
1CR	TAPE READ RECOVERY-WRITE CM FOR S-FORMAT 9-TRACK TAPES
1CT	TAPE READ RECOVERY-WRITE CM FOR SCOPE FORMAT 7-TRACK TAPES
1CS	TAPE READ RECOVERY-WRITE CM FOR S-FORMAT 7-TRACK TAPES
1RT	SCOPE TAPE READ DRIVER
1RS	STRANGER (S) TAPE READ DRIVER FOR 7-TRACK TAPES
1NR	9-TRACK S-FORMAT TAPE READ DRIVER
1W9	9-TRACK SCOPE FORMAT TAPE WRITE DRIVER
1C9	TAPE READ RECOVERY-WRITE CM FOR 9-TRACK SCOPE TAPES
1R9	9-TRACK SCOPE FORMAT TAPE READ DRIVER

1PL	PLOTTER PROGRAM (DUMMY)
1SX	PROCESS STACK PROCESSOR ERRORS
1TD	DUMP OUTPUT FILE TO TAPE
1IB	INITIATE A BATCH JOB
1SI	SWAPIN OR ROLLIN A JOB
6SI	OVERLAY TO PROCESS PARITY ERROR FOR 1SI
1SO	SWAPOUT OR ROLLOUT A JOB
7T1	ANSI/DISPLAY CODE CONVERSION TABLE FOR MMTC MEMORY
7T2	EBCDIC/DISPLAY CODE CONVERSION TABLE FOR MMTC MEMORY
2LP	ON-LINE PRINTER DRIVER
2PC	ON-LINE CARD PUNCH DRIVER
2RC	ON-LINE CARD READER DRIVER
2TJ	TRANSLATE JOB CARD
4LB	TAPE LABEL PROCESSOR (ANSI)
6ER	4LB OVERLAY
6EW	4LB OVERLAY
6LM	LOAD FIELD NAME MESSAGES
6LC	LOAD CONVERSION TABLE INTO MMTC
6L1	4LB INLAY TO CONVERT PRU COUNT
6L2	BCD CONVERSION TABLE INLAY FOR 4LB
6L3	4LB INLAY TO CHECK THAT PROPER CONVERSION TABLE IS IN THE MMTC
6L4	4LB INLAY FOR DEBUG MESSAGE WRITER
6L5	4LB INLAY TO FORMAT THE LABEL INFORMATION
6L7	4LB INLAY TO PACK AND WRITE LABEL TO TAPES TABLE
4LC	3000 TYPE LABEL PROCESSOR
6CR	3000 LABEL PROCESSOR READ FUNCTION CODE OVERLAY
6CW	3000 LABEL PROCESSOR WRITE FUNCTION CODE OVERLAY
6WM	DAYFILE MESSAGES FOR I/O REQUESTS
7W1	OVERLAY TO 6WM
7W2	OVERLAY TO 6WM
CEM	CENTRAL ERROR MANAGER
1TS	TAPE SAMPLER
XRQ	RESTORE QUEUE
XDQ	DUMP QUEUE
DSP	PROCESS DISPOSE FUNCTION
CKP	TAPE CHECKPOINT
CY1	RESET FNT FOR RESTART
RST	RESTORE CONTROL POINT AREA FOR RESTART
1RC	RELOAD CORE FOR RESTART
RPV	PROCESS REPRIEVE FUNCTION
STS	STATUS ROUTINE
JDP	PROCESS JOB DEPENDENCY
LDL	LOADER UTILITY
LDV	ABSOLUTE OVERLAY LOADER
LDW	ABSOLUTE OVERLAY LOADER
MDI	MOVE SYSTEM DIRECTORY (EDITLIB USE)
6MD	DUMMY EDITLIB OVERLAY
TDS	TERMINATE DEADSTART
PRM	PERMISSION CODE PROCESSOR FOR PERMANENT FILES
RWE	CHECK IF INTERCOM CONTROL POINT
CON	CONNECT FILE NAME TO TTY
EPF	SEND PERMANENT FILE AUDIT INFORMATION

*** INTERCOM ***

1ZF	MULTIPLEXOR DRIVER
8ZF	MULTIPLEXOR DRIVER
9ZF	MULTIPLEXOR DRIVER
1XP	HIGH SPEED EXPORT PROCESSOR
2XP	PROCESS SPECIAL DIRECTIVES

3XP	PROCESS OUTPUT DATA STREAM
5XP	OUTPUT BANNER TO TERMINAL
6XP	OUTPUT LACED CARD
4XP	PROCESS INPUT DATA STREAM
1XG	GRAPHICS INPUT/OUTPUT PROCESSOR
7XP	END PROCESSING
1LX	LCC EXPORT PROCESSOR
3LX	OVERLAY TO 1LX
4LX	OVERLAY TO 1LX
1WB	INTERCOM V4 WIDE BAND DRIVER
0ZZ	INTERCOM LCC DRIVER INITIATOR
1ZZ	INTERCOM LCC DRIVER
9ZZ	INTERCOM LCC DRIVER
1CI	COMMON COMMUNICATIONS INTERFACE
3CU	ASSIGN NEW USER TABLE
3CI	USER TABLE PROCESSOR
3CT	USER TABLE PROCESSOR
3CX	COMMAND PROCESSOR FOR 1CI
3CF	CLEAN UP PHASE OF 1CI
1I1	STARTS INTERCOM AT CONTROL POINT
3TT	READ/WRITE FOR REMOTE TERMINAL
3T1	READ SEGMENT OF 3TT
3T2	WRITE SEGMENT OF 3TT
1BR	INTERCOM VR4 BUFFER MANAGER
1QP	MUJ PROCESSOR
1QM	1QP OVERLAY
1FT	INTERCOM VR4 LOW SPEED EXPORT PROCESSOR
8FT	INPUT FILE TRANSMISSION
9FT	OUTPUT FILE TRANSMISSION
1PJ	PROCESS JOB CARD
1ID	SEND DAYFILE MESSAGE TO INTERCOM TERMINAL
1IM	SENDS MESSAGES TO TERMINALS FROM PP ROUTINES
1DS	H DISPLAY GENERATOR FOR INTERCOM VR4
T76	INTERCOM 4.1 7000 DISPLAY GENERATOR
TBL	INTERCOM VR4 TABLE TRANSMITTER
FNT	MODIFIES VR4 FNT ENTRY FOR BATCH AND DROPO
IUP	INITIATE USER PROGRAM
IAP	INITIATE ANOTHER PROGRAM
MES	MESSAGE TRANSFER ROUTINE
2ME	OVERLAY TO MES
3ME	OVERLAY TO MES
MUJ	MULTI-USER JOB INITIALIZATION
MAC	MUJ ACCOUNTING
FAD	FILE ATTACH/DETACH FOR MUJ
GBJ	BEGINS GRAPHICS MODE
GEJ	ENDS GRAPHICS MODE
1GJ	UPDATE IGS QUEUE WHEN IGS DATA STREAMS DEFINED
2GJ	FORMAT SCOPE ERROR MESSAGES FOR 274 IGS
1GR	274 IGS RECOVERY

*** SDA ***

MSD

*** AIDS ***

VSM
1VG
SPY
SPZ

*** CE DIAG ***

MTT
9CA
9CB
9CC
9CD
9CE
9CF
9CG
LPT
DF4
9GR
9GQ
9GS
9GU
9GV
9GW
9GX
9GY
9GZ
9G0
9G1
9G2
9G3
9G4
9G5
9G6
9G8
9G9
DF7
9K9
9KA
9K8
9KB
9KC
9KD
9KE
9KF
9K6
9KG
9K5
9KH
9K4
9K3
9KI
9K2
9K1
9K0
9K7
9KJ
9KZ
9KY
9KX
9KK
9KW
9KL
9KM
9KN
9KO
9KP
9KQ
9KR
DF8
8CA
8CB
8CC
8CD
8CE
8CF
8CG
8CH
8CI
8CJ
8CK
8CL
8CM
8CN
8CO
8CP
8CQ
8CR
8CS
8CT
8CU
LP1
9IY
9I7
9IZ
9I8
9I0
9I9
CP1
CR1
APR
IEF
RMS
DLE

NUCLEUS CM Resident Library

Routines callable by control card, including all compiler (0,0) overlays. COMPASS 3.0, LOADER, and EDITLIB non-0,0 overlays are included in NUCLEUS.

```
**********
*        *
* NUCLEUS*
*        *
**********
*** SCOPE ***
    PFCOP
    LOADPF
    TRANSPF
    DUMPF
    REQUEST
    RESTART
    COPYXS
    SEGBILD
    SEGRES
    LOAD
    LOADC
    LOADM
    LOADU
    LOADUC
    LOADUM
    UCLOAD
    LIBRARY
    LOADO
    LOAD01
    LOAD02
    LOAD03
    COMBINE
    XXXRESQ
    XXXDMPQ
    EDITLIB
    EDITSYS
    EDITUSR
    COPYBCD
    LABEL
    COPYL
    LISTMF
    CFC
    IORANDM
    IO
    COMPARE
    CHEKPT
    TRAP
    TRAPPER
    SETCORE
    BKSP
    COPY
    COPYBF
    COPYCF
    COPYCR
    COPYBR
    COPYN
    COPYSBF
    REWIND
    UNLOAD
    RETURN
    UPDATE
    SKIPB
    SKIPF
    TRANSR
    TRANSF
    DISPOSE
    RECOVR
    SYS.RM
    AUDIT
    DMPECS
    PFDUMP
    LDRTEXT
    PFMTEXT
    SCHTEXT
    PFTEXT
    SCPTEXT
    CPCTEXT
    STATEXT
    IFTEXT
*** CE DIAG ***
    CEDIAG
    CERMS
    CEFAP
    EC2
    CM6
    MY1
    ALS
    FST
    CT3
    CU1
    ALX
*** 6RM ***
    SYSTEXT
    IOTEXT
    TXT6RM
    FILE
*** FORTRAN ***
    FTNMAC
    FTN
*** COBOL ***
    COBOL
    COPYCL
*** SORT/MERGE ***
    SORTMRG
    SMTEXT
*** INTERCOM ***
    LOGIN
    CONNECT
    SITUATE
    DISCONT
    BATCH
    Q
    PAGE
    SEND
    LOGOUT
    CONVERT
    BRESEQ
    TEACH
    FILES
    STORE
    DISCARD
    FETCH
    XEQ
    PASSWRD
    ASSETS
    READ
    ERRORS
    TESTLP
    EDITOR
*** SIS ***
    ESTMATE
    SISTAT
*** SDA ***
    KYAN
    DCREATE
*** 8 BIT ROUTINES ***
    COPY8P
*** FORM ***
    FORM
*** QUERY/UPDATE ***
    QU
    QU1
    QU2
*** QUIDDL ***
    QUIDDL
*** COMPASS ***
    COMPASS
    COMP2$
*** RUN ***
    RUN
*** SYMPL ***
    SYMPL
*** AIDS ***
    SIF
    CPSPY$
    DOCK
    WHEN
    BETO6RM
    PRNTSPY
    CONVPF
    RANCONV
    SIFT
    DATAR
    SORT
    LEE
*** BASIC ***
    BASIC
*** ALGOL ***
    ALGOL
    ALGTEXT
*** SIMULA ***
    SIMULA
*** SIMSCRIPT ***
    SIMS
*** APT ***
    APT
*** MARS ***
    MARS
```

SYSOVL   CM or Disk Resident Library

Overlays other than (0,0) for the SCOPE 3.4 product set. This library is the central source for all system overlay calls.

```
**********
*        *
*  SYSOVL*
*        *
**********
*** FORTRAN ***
    LSTPRO$
    PS1CTL$
    CLOSE2$
    FTNMSG$
    FASS15$
    FASS14$
*** COBOL ***
    COBOL10
    COBOL11
    COBOL12
    COBOL13
    COBOL14
    COBOL15
    COBOL16
    COBOL20
    COBOL21
    COBOL22
    COBOL23
    COBOL24
    COBOL30
*** SORT/MERGE ***
    SORT10
    SORT11
    SORT12
    SORT20
    SORT30
    SORT31
    SORT33
    SORT32
    SORT40
*** QU ***
    Q6000
    Q7000
*** INTERCOM ***
    ERFTN
    ERCOB
    ERRUN
*** QUDDL ***
    QUDDL10
    QUDDL20
*** RUN ***
    RUN1
    Q8DIAGP

*** SYMPL ***
    SYMP10
    SYMP15
    SYMP16
    SYMP17
    SYMP13
    SYMP30
    SYMP31
    SYMP32
    SYMP40
    SYMP50
    SYMP51
    SYMP52
*** ALGOL ***
    ALG0
    ALG1
    ALG2
    ALG3
    ALG4
    ALG5
*** SIMULA ***
    SIM0
    SIM1
    SIM2
    SIM3
    SIM4
    SIM5
*** AFT ***
    SECTN1
    OVER11
    OVER12
    OVER13
    OVER14
    OVER15
    OVER16
    OVER17
    OVER18
    OVER19
    CALSEC2
    ARLM2
    POCKET
    RLDSR
    PCLCON
    CALSEC3
    SECTN4

*** MARS ***
    LDSV70
    LAQDE
    DEFSUB
    LAQLO
    SOLINK2
    SRTMRG
    SOLINK3
    SOLINK4
    UF30
    UF39
    UF31
    UF32
    UF33
    UF34
    UF35
    UF36
    UF37
    UF38
    SOLINK9
    RES00
    LAQRE
    RES01
    RES06
    RES015
    RES017
    RES020
    RES013
    RES02
    RES03
    RES010
    RES04
    RES011
    RES012
    MASSUP
    MASS51
    MASS52
    MASS53
    MASS54
    RPG60
    RPG61
    RPG62
    RPG63
```

Object time routines for FORTRAN Extended 4.0

```
**********
*        *
* FORTRAN*
*        *
**********
*** FORTRAN ***        READEC      DSQRT=      MASK
    BACKSP=            REMARK      DTOD*       OR
    BUFIN=             SLITE       DTOD=       SHIFT
    BUFIC=             SSWTCH      DTOI*       ABS
    BUFOUT=            WRITEC      DTOI=       AIMAG
    BUSY               ACOS        DTOX*       AINT
    CLCSMS             ASIN        DTOX=       AMAX0
    CONDIS             ACOSIN=     DTOZ*       AMAX1
    DECODE=            ALOG=       DTOZ=       AMIN0
    ENCODE=            ALOG        EXF         AMIN1
    ENDFIL=            ALOG10      EXP=        AMOD
    EOF                ATAN        HYPERB=     CMPLX
    FORSYS=            ATAN=       ITOD*       CONJG
    FTNBIN             ATAN2       ITOD=       COUNT
    GETFIT=            ATAN2=      ITOJ*       DABS
    INFB=              CABS        ITOJ=       DBLE
    INFC=              CABS=       ITOX*       DIM
    IOCHEC             CCOS        ITOX=       DMAX1
    KODER=             CCOS=       ITOZ*       DMIN1
    KRAKER=            CEXP        ITOZ=       DSIGN
    LABEL              CEXP=       RANF        FLOAT
    LENGTH             CLOG        RANSET      IDIM
    NAMIN=             CLOG=       SIN         INT
    NAMOUT=            COS         SINCOS=     ISIGN
    OPENMS             COS=SIN     SQRT        MAX0
    OUTB=              CSIN        SQRT=       MAX1
    OUTC=              CSIN=       TAN         MIN0
    RANMS=             CSQRT       TAN=        MIN1
    READMS             CSQRT=      TANH        MOD
    REWIND=            DATAN       TANH=       REAL
    STINDX             DATAN2      XTOD*       BUGARR
    SYSTEM             DATAN=      XTOD=       BUGCLL
    UNIT               DEXP        XTOI*       BUGCTL
    WRITMS             DEXP=       XTOI=       BUGFUN
    XRCL               DLNLOG=     XTOY*       BUGGTA
    CLOCK=             DLOG        XTOY=       BUGSTO
    DISPLA             DLOG10      XTOZ*       BUGTRC
    DUMP               DMOD        XTOZ=       BUGTRT
    GOTOER=            DMOD=       ZTOI*       DEGFIT=
    LEGVAR             DCOS        ZTOI=       FTNERR=
    MOVLEV             DSIN        AND         TRACEX
    OVERLAY            DSNCOS=     COMPL
    PAUSE=             DSQRT       LOCF
```

COBOL   Disk Resident Library

Object time routines for COBOL 4.0 and SORT/MERGE 4.0

```
**********
*        *
*  COBOL *
*        *
**********
*** COBOL ***
    DDSUBMV
    DDCKETL
    DDCKLA
    DDWALAB
    DDOPWA
    DDSQWA
    DDRAWA
    DDFINIS
    DDCOBIC
    DDLABLS
    DDSQLAB
    DDMLLIT
    DDCMLIT
    DDSBLIT
    DDZPAR
    DDSQIO
    DDSTIO
    DDDAIO
    DDISIO
    DDUPCNT
    DDADVAN
    DDUSE
    DDEXP
    DDANCM
    IFNUMA
    DDIFALP
    DDALFC1
    DDSUBSC
    DDSBSC2
    DDSBSER
    DDSBSC6
    DDSBSC7
    DDSOL
    DDSORT
    DDDSPLY
    DDXCEPT
    DDXCPT
    DDDADD
    DDFIVED
    CCNSTAN
    DDTRUEL
    DDZONE
    DDZN3A
    DDZN5A
    DDZN6A
    DDZNTAB
    STRPN
    DDSTPTB
    DDZN7A
    DDSTP1A
    DDSTP2A
    DDSTRP3
    DDDATCN
    DDTENDP
    DDTENS
    DDTNTHS
    DDCVBD
    DBN1SA
    DDBN
    DBN6SA
    DBN6SS
    BN1DSA
    DBN1DS
    BN6DSA
    DBN6DS
    DB1DDA
    DBN1DD
    DB6DDA
    DBN6DD
    DBN1SB
    BN1SBA
    DDED
    AOPTN
    DDEDAL
    DDEDIT
    DDEDALP
    DDEOP
    DDEXAMC
    DDMOVE
    DDBCDCM
    DDSCM
    CONV1
    DDPAGE
    DDLNCT
    DDSPACE
    DDINITL
    DDPRINT
    DDRGEN
    DDBEGRP
    DDENDRP
*** SORT/MERGE ***
    DSMCON
    TSMCON
    MSMCON
    MACPRO
    SOCHKR
    ENDPRO
    EXTRACT
    KYCPL
    KEYCOD
    BUFALL
    TRNSRT
    TSC
    TMC
    NEXRCM
    FMC
    FMIP
    FMOP
    FSRTGET
    MRGCON
    DDC
    MRGDSN
    SRTCON
    SRTPUT
    SRTGET
```

RUN2P3 Disk Resident Library

Object time routines for RUN 2.3

```
**********
*        *
*  RUN2P3*
*        *
**********
***SCOPE***
    R23RCVR             FTNBIN
*** RUN ***             IOINT
    RUNCALL             INPUTN
    ACGOER              INPUTS
    DBLE                LENGTH
    EXP                 OUTPTN
    IBAIEX              OUTPTS
    INITMS              OVERFL
    LEGVAR              OVERLAY
    LOCF                PAUSE
    SINCOS              RANF
    SNGL                RBAIEX
    SQRT                RBAREX
    SYSTEM              READEC
    TAN                 READMS
    XRCL                REMARK
    ALNLOG              BACKSP
    ASINCOS             BUFFEI
    ATAN                BUFFEO
    ATAN2               CONDIS
    CABS                ENDFIL
    CBAIEX              IOCHEC
    CCOS                IFENDF
    CEXP                INPUTB
    CLOG                INPUTC
    CSIN                IOCHEK
    CSQRT               KODER
    DABS                KRAKER
    DATAN               OUTPTB
    DBADEX              OUTPTC
    DBAIEX              REWINM
    DEXP                SLITE
    DISPLA              SLITET
    DLNLOG              SSWTCH
    DMOD                START
    DSIGN               SIS=TIM
    DSINCOS             TANH
    DSQRT               TIME
    DUMP                WRITEC
    DVCHK               WRITMS
```

SYSIO   Disk Resident Library

SCOPE system I/O routines, 6000 Record Manager (6RM), including the Direct Access (DA) and Indexed Sequential (IS) modules, other independent I/O routines available to the product set members plus the SYMPL and FORM object library routines.

```
***********
*         *
* SYSIO   *
*         *
***********
*** 6RM ***    LXER.SQ        GETPAR         SD$SEEK                    T8.NCH
    LBUF.SQ    ENDF.SQ        NEWCONS        SPACE                      T8.MVB
    SQ.RM      WEOR.SQ        FTNMAIN        SQUEEZE                    T8.C6.
    WA.RM      WTMK.SQ        READCRD        SEARCH                     T8.CN6P
    IS.RM      DLT.RM         PRINTER        SD$TRC                     T8.CN6T
    DA.RM      DLT.SQ         NRER1          FILEDA                     T8.CN6C
    CIO.RM     DLT.WA         DEFAULT        FILEIS                     T8.CNC6
    ERR.RM     SKFL.RM        ACCESS         FILESQ                     T8.CNT6
    MOVE.RM    SKFF.SQ        ACCPRCK        FILEWA                     T8.CNA
    CHWR.SQ    SKFL.SQ        CHKFET         FITCOM                 *** FORM ***
    MCT.RM     SKFL.WA        DELETE         ERRPROC                    FM$BDEF
    MEM6RM     GETN.RM        FORCEW         OPNCLS                     FM$CRCR
    CHEK.RM    GETN.SQ        INSERT         GET                        FM$BCON
    OPEN.RM    GETN.WA        FTNCALL        PUT                        FM$QAL
    PDF.RM     REPL.RM        OPENNEW        REPLC                      FM$REF
    OPEN.SQ    REPL.SQ        OPENOLD        DLTE                       FM$PRT
    OPEX.SQ    REPL.WA        REPLACE        GETN                       FM$SCAN
    OPEN.WA    SEEK.RM        REPOS          SEEKF                      FM$BDIA
    OPEN.IS    SEEK.SQ        SEEK           SKIP                       FM$SEQ
    OPEN.DA    SEEK.WA        SETBLKD        REWND                      FM$CON
    PUT.RM     SKBL.RM        SETBLKI        GETP                       FM$FORM
    RLEQ.RM    SKBL.SQ        SETCOLL        PUTP                       FM$CCNV
    PUT.SQ     SKFP.SQ        SETERR         SCANCWA                    FM$TMOV
    WAR.SQ     SKBL.WA        SETKEY     *** 8 BIT ROUTINES ***         FM$KOD
    PUT.WA     MSG.RM         STFETF$        T8.ERR                     FM$XX3
    CLSF.RM    R6RM.RM        STKEYF$        XPACK                      FM$TFX3
    CLSF.SQ *** SIS ***       TERMNAT        XPANC                      FM$TFK3
    FLSH.SQ    ACCMMON    *** SDA ***        XMOVE                      FM$TFL3
    CLSV.RM    BUFALOC        BINDSDA        XCOMP                      FM$TBT3
    CLSV.SQ    DATM           CLOSE          TDCRACK                    FM$TXX6
    CLSF.WA    DIAGNOS        SD$DEL         T8.HXTB                    FM$TFX6
    CLSV.WA    FINDIT         D.DIAGN        BDPTAB                     FM$TFK6
    CLSF.IS    INDEXM         SD$FIND        T8.6TAB                    FM$TFL6
    CLSF.DA    IOMGR          SD$GETN        XFILE                      FM$TBT6
    REW.RM     PPCALL         SDAHASH        XREAD                  *** RUN2P3 ***
    REW.SQ     CEOI6RM        SD$ADD         XWRITE                     GETBA
    GET.RM     WRIT6RM        D.INREC        T8.CHK                     SIO$
    GET.SQ     REGSAV         SD$IC          T8.PARS                *** SYMPL ***
    Z.SQ       SAAM.IS        SD$KEY         T8.CCM                     SYMHBS$
    R.SQ       SISCLSE        RECHOM         T8.TXT                     SYMBSW$
    W.SQ       SISKEY         D.OPN          T8.CNC                     SYMIO
    DT.SQ      SISOPEN        OVFM           T8.CNT                     SYMSM$
    FSU.SQ     SISRPV         SD$RPLC        T8.CN6                     SYMSC$
    BTRT.SQ    SISSEEK        RETRV          T8.NNA
    GET.WA     SISSKIP        SD$RLCV        T8.TSTC
    PUTL.SQ    SIZES          SDARPVD        T8.TSTT
    GETL.SQ    KWAKER         SAAM.DA        T8.TST6
```

SYSMISC Disk Resident Library

Object time routines for BASIC, ALGOL, SIMULA and SIMSCRIPT. Additionally, the FTN 3.0, COBOL 3.0 and SORT/MERGE 3.0 object time routines appear in SYSMISC if they are installed.

```
**********
*        *
*SYSMISC *
*        *
**********
***SCOPE***
   F30RCVR
*** BASIC ***
   BASOGEN
   BASEGEN
   BASIGEN
   BASIINP
   BASATRI
   BASAATN
   BASALEP
   BASARST
   BASMCPR
*** ALGOL ***
   ALGORUN
   ALGLB00
   ALGLB01
   ALGLB02
   ALGLB03
   ALGLB04
   ALGLB05
   ALGLB06
   ALGLB07
   ALGLB10
*** SIMULA ***
   SIMURUN
   SIMLB00
   SIMLB01
   SIMLB02
   SIMLB03
   SIMLB05
*** SIMSCRIPT ***
   XXINIT
   XXERR
   ERRLP
   KXXDAT
   XXCNVH1
   XXGARG1
   XXT1
   XXBSR
   XXSKR
   XXREW1
   XXBSF
   XXOPEN
   XXWEF1
   XXINP1
   XXLINK1
   XXMEM1
   XXOUTP1
   XXCNWH1
   GRAND
   PRTIME
   XXMAINS
   XXTMR
   CRADS
   GETPUT
   FILREMV
   XXSTEP
   XXBIN
   XXBOUT
*** FTN3P0 ***
   SYSTEM$
   ACGOER$
   BACKSP$
   BUFFEI$
   BUFFEO$
   BUGARR$
   BUGCLL$
   BUGCTL$
   BUGFUN$
   BUGGTA$
   BUGSTO$
   BUGTRC$
   BUGTRT$
   CONDIS$
   DBGFET$
   DUMP$
   ENDFIL$
   FTNBIN$
   FTNERR$
   IFENCF$
   INITMS$
   INPUTB$
   INPUTC$
   INPUTN$
   INPUTS$
   IOCHEC$
   IOCHEK$
   KODER$
   KRAKER$
   LENGTH$
   MASK$
   OUTPTB$
   OUTPTC$
   OUTPTN$
   OUTPTS$
   READEC$
   READMS$
   REWINM$
   TRACEX$
   WRITEC$
   WRITMS$
   ABS$
   IABS$
   DABS$
   AINT$
   INT$
   IDINT$
   AMOD$
   MOD$
   AMAX0$
   AMAX1$
   MAX0$
   MAX1$
   DMAX1$
   AMIN0$
   AMIN1$
   MIN0$
   MIN1$
   DMIN1$
   FLOAT$
   IFIX$
   ISIGN$
   SIGN$
   DSIGN$
   DIM$
   IDIM$
   SNGL$
   REAL$
   AIMAG$
   DBLE$
   CMPLX$
   CONJG$
   SHIFT$
   ACOS$
   ALNLOGE
   ALOG$
   ALOG10$
   AND$
   ASIN$
   ASNCOSE
   ATAN$
   ATANE
   ATAN2$
   ATAN2E
   CABS$
   CABSE
   CCOS$
   CCOSE
   CEXP$
   CEXPE
   CLOG$
   CLOGE
   COMPL$
   COS$
   COSESIN
   CSIN$
   CSINE
   CSQRT$
   CSQRTE
   DATAN$
   DATAN2$
   DATANE
   DCOS$
   DEXP$
   DEXPE
   DLNLOGE
   DLOG$
   DLOG10$
   DMOD$
   DMODE
   DSIN$
   DSNCOSE
   DSQRT$
   DSQRTE
   DTOD$
   DTODE
   DTOI$
   DTOIE
   DTOX$
   DTOXE
   DTOZ$
   DTOZE
   DISPLA$
   EXP$
   EXPE
   HYPERBE
   ITOD$
   ITODE
   ITOJ$
   ITOJE
   ITOX$
   ITOXE
   ITOZ$
   ITOZE
   LABEL$
   LEGVAR$
   LOCF$
   OR$
   OVERLA$
   PAUSE$
   RANDOME
   RANF$
   RANGET$
   RANSET$
   REMARK$
   SEGMEN$
   SIN$
   SINCOSE
   SLITE$
   SLITET$
   SQRT$
   SQRTE
   SSWTCH$
   TAN$
   TANE
   TANH$
   TANHE
   TIME$
   XTOD$
   XTODE
   XTOI$
   XTOIE
   XTOY$
   XTOYE
   XTOZ$
   XTOZE
   ZTOI$
   ZTOIE
*** COBOL3P0 ***
   DDSUBMV
   DDCOBIO
   DDEXP
   DDANCM
   DDSUBSC
   DDSOL
   DDSORT
   FINIS
   DDDSPLY
   DDXCEPT
   DDDADD
   DDFIVED
   DDTRUBL
   DDZONE
   DDSTRP
   DDDATCN
   DDTENDP
   DDTENS
   DDTNTHS
   DDCVEC
   DDBN
   DDED
   DDEDAL
   DDEDIT
   DDEDALP
   DDEOF
   DDEXAMO
   DDMOVIO
   DDBCDCM
   CONV1
   DDPAGE
   DDLNCT
   DDSPPR
   DDSPACE
   DDINITL
   DDPRINT
   DDRGEN
   DDBEGRP
   DDENDRP
*** SORTV3P0 ***
   DECK10
   DECK12
```

IGS274   Disk Resident Library

INTERCOM 4.1 routines associated with the Interactive Graphics 274 terminal subsystem.

```
**********
*        *
* IGS274 *
*        *
**********
***INTERCOM***
     AELBUT
     GFONTA
     GFONTN
     GIABRT
     GIBRD
     GIBWRT
     GIESZ
     GICONF
     GICOPY
     GIFID
     GIFSID
     GILKID
     GIMAC
     GIMESG
     GIMOVE
     GIPLCT
     GITIMM
     GUARC
     MARK
     IEG
     GUARCG
     GULINE
     GUPNTS
     DMINIT
     DMFLSH
     DMGTED
     DMRLED
     DMGET
     DMUTIL
     DMDMP
     AEXEC
     CCNSCLC
     ERASER
     GIANE
     GIANS
     GIBUT
     GICLR
     GIDISP
     GIEBUT
     GIMASK
     GUAN
     MACBYT
     GURSET
     GUTIL
     GVALID
     SEGENR
```

SCOPE 3.4 DOCUMENTS

CDC CYBER 70 and 6000 Series computer systems hardware and software information is available in the following documents:

Hardware Documents	Publication Number
CDC CYBER 70/Model 72 Systems Description and Programming Information (vol.1) (RM)	60347000 C
CDC CYBER 70/Model 73 Systems Description and Programming Information (vol.1) (RM)	60347200 A
CDC CYBER 70/Model 74 Systems Description and Programming Information (vol.1) (RM)	60347400 D
CDC CYBER 70/Models 72, 73, and 74 Instruction Descriptions (vol.2) (RM)	60347300 C
CDC CYBER 70 Computer Systems-7030 Extended Core Storage (RM)	60347100 D
CDC CYBER 70/Models 72, 73, and 74 and 6000 Series Computer Systems I/O Specifications (RM)	60342500 C
7600 Computer System (RM)	60258200 C
7600 Computer System Hardware Features (RM)	60258500 B
7618-1, 7628-2, and 7629-1/2 Magnetic Tape Controller (RM)	60341600 B
7681 Data Channel Converter (RM)	60356000 A
7683-1 Remote Channel Coupler (RM)	60296200 C
7611-10 Buffer Controller (RM)	60275000 B
6000 SCOPE On-line Diagnostics (GIM)	60309800 A
6000 Series Computer Systems I/O Specifications (RM)	60100000 N
6000 Series Computer Systems Instruction Card	60164500 C
6000 Series Computer Systems Instruction Codes	60141900 D
6000 Extended Core Storage (RM)	60225100 C
6682-A and 6683-A Satellite Coupler (RM)	60334300 A
6681-B/C/D and 6684-A Data Channel Converter (RM)	60334400 A
6676-A Teletype Multiplexer (RM)	38706000 B
6676-B/C Teletype Multiplexer (RM)	38707800 D
6673/6674 Data Set Controller (RM)	60334500 A
6671-A/B and 6671-1 Data Set Controller (RM)	60334600 A
6642-1 Distributive Data Path Operations/Programming (RM)	60376300 A
6641-A Data Set Controller (RM)	60334200 A
6639-A/B Disk File Controller (RM)	60334100 A
6622-A/B Magnetic Tape Controller (RM)	60333600 A
6603-A/B/C Disk File Controller (RM)	60334000 A
6602-A/B/C/D/E and 6612-A/B/C/D/E/F/G Console Display Controller (RM)	60333900 A

Hardware Documents	Publication Number
362X-A/B Magnetic Tape Controller (RM)	60331500 A
3555-1 Line Printer Controller (RM)	60231300 A
3553-1/2 Mass Storage Controller (RM)	60278500 A
3518 and 3528 Magnetic Tape Controller (RM)	60287600 D
3447-A/B Card Reader Controller (RM)	60332300 A
3446-A/B and 3644-A Card Punch Controller (RM)	60332100 A
3436-A and 3637-A Drum Storage Controller (RM)	60333100 A
3243-A/B Magnetic Tape Controller (RM)	60331600 A
3234-A/B Mass Storage Controller (RM)	60333300 A
3232-A/B Disk Drive Controller (RM)	60165300 01
3231-A Disk Pack Controller (RM)	60110900 C
3228-A/B and 3229-A/B Magnetic Tape Controller (RM)	60331700 A
3000 Series Computer Systems Peripheral Equipment (vols. 1, 2) (RM)	60108800 U
733-10 High Speed Batch Subsystem Functional Characteristics and Programming Manual (RM)	60329500 01
732 Medium Speed Batch Subsystem Functional Characteristics and Programming Manual (RM)	60348100 01
720 Digital Communications Terminal Control Unit (RM)	36814300
713 Conversational Display Terminal Hardware Programming Manual (RM)	62033400 02
713 Conversational Display Terminal (OG)	62037900 00
712 Printer Terminal Hardware Programming Manual (RM)	62073900
712 Printer Terminal (OG)	62049400 00
711 CRT Display Terminal Hardware Programming Manual (RM)	62022700 02
711 CRT Display Terminal (OG)	62034100 01
595 Print Cartridge (RM)	44855800 C
512 Line Printer (RM)	44980100 C
505-A/B High Speed Line Printer (RM)	40820800 A
501-C/D High Speed Line Printer (RM)	40807200
501-A/B High Speed Line Printer (RM)	40805000
417-1 Card Punch Controller (RM)	60283600 B
405 High Speed Card Reader (RM)	40809300 E
364-4/5 Communications Multiplexer (RM)	41612700 A
364-3 Communications Multiplexer (RM)	41612500 A
364-1/2 Communications Multiplexer (RM)	41610900 A
304-1 Communications Multiplexer (RM)	13796400 A
275 Digigraphic Buffer Memory (RM)	60181300 A
274 Digigraphic Console, Theory of Operation (RM)	60279100 A
273 Digigraphic Console (RM)	60188600 A
271 Digigraphics Controller (RM)	60181400 A
250-2 Display Subsystem (RM)	82129000 A
222-A/B Line Printer Controller (RM)	47966700 E
200 User Terminal (RM)	82128000 B

Software Documents	Publication Number
ALGOL 3 (RM)	60329000 B
APT 2 (GIM)	60172100
APT 2 (RM)	60174500 C
APT 2 (IN)	60193100
BASIC Language 2 (RM)	60306200 C
COBOL 4 (GIM)†	60327800 A
COBOL 4 (RM)	60384100 A
COBOL 4 (IN)†	60328400 A
COMPASS 3 (GIM)†	60343400 A
COMPASS 3 (RM)†	60360900 A
COMPASS 3 (IN)	60361000 A
COMPASS 3 (Instruction Cards)†	60361700 A
EXPORT/IMPORT 8231 HS 1 (RM)	60189100 A
1700 IMPORT HS 1 (RM)	60235400 D
1700 MSOS IMPORT HS 1 (RM)	60305700 A
1700 MSOS IMPORT HS 1 (OG)	60359800 A
FORM 1 (RM)	60307000 D
FORTRAN Ext. 4 (GIM)†	60327900 A
FORTRAN Ext. 4 (RM)†	60305600 B
FORTRAN Ext. 4 (IN)†	60357900 A
FORTRAN Ext. DEBUG (UG)†	60329400 B
FORTRAN 2.3 (RM)	60174900 F
FORTRAN 2.3 (IN)	60189500 E
SIFT (PSB)	60358400 A
Math Science Library (8 vols) (RM)*†	60327500 A
Math Science Library Usage Information*	60329200 A
Math Science Library (GIM)†	60328900 A
240 Series Interactive Graphics System (RM)	17307300 D
274 Interactive Graphics System 2 (RM)	60358800 A
274 Interactive Graphics System 2 (OG)	60359800 A
INTERCOM 4 (RM)	60307100 B
INTERCOM 4 Multi-User Job Capability (PSB)*	60358300 B
LOADER (RM)†	60344200 B
Multi-Access Retrieval System 2 (MARS VI) for Partial Inversion (RM)	17313100 A
Multi-Access Retrieval System 2 (MARS VI)	17313000 A

*Limited distribution; available only through Software Manufacturing and Distribution Section, Sunnyvale, Ca. 94086.

†Combined 6000/7000 Series manual.

Software Documents	Publication Number
PERT/TIME 1 (GIM)	60133300
PERT/TIME 1 (RM)	60133600
Programming Reference Aids	60158600
QUERY UPDATE 1 (RM)	60307500 D
QUIDDL 1 (RM)	60327900 D
Record Manager 1 (RM)	60307300 B
SCOPE 3.4 (RM)	60307200 D
SCOPE 3.4 (SPRM)	60306500 B
SCOPE 3.4 (OG)	60327300 C
SCOPE 3.4 (IH)	60307400 B
SCOPE 3.3/3.4 Conversion Aids (PSB)	60358200 B
SIMSCRIPT 2 (GIM)	60173500 A
SIMSCRIPT 2 (RM)	60178300 C
SIMULA 1 (GIM)	60251900 A
SIMULA 1 (RM)	60234000 E
SIMULA 1 (IN)	60235100 A
SORT/MERGE 4 (GIM)	60342400 A
SORT/MERGE 4 (RM)	60343900 B
SYMPL 1 (RM)	60328000 A
UPDATE (RM)†	60342500 B
UPDATE (IN)†	60283100 A
UPDATE (OG)	60281600 A
8-BIT Subroutines (RM)	60359400 01

GIM	General Information Manual
IH	Installation Handbook
IN	Instant
OG	Operator's Guide
PSB	Programming Systems Bulletin
RM	Reference Manual
SPRM	System Programmers Reference Manual
UG	User's Guide

*Limited distribution; available only through Software Manufacturing and Distribution Section, Sunnyvale, Ca. 94086.

†Combined 6000/7000 Series manual.

RELEASE DESCRIPTION

SCOPE 3.4 is a revised and extended version of the SCOPE 3 operating system.

HARDWARE CONFIGURATION

The minimum hardware configuration consists of:

1	6000/CDC CYBER 70 Series computer
24	million characters of mass storage on any combination of the following: 854 disk drive 865 drum 814, 6603, or 6638 disk files with standard option 10037-A 821 data file 841 multiple disk drive
1	405 card reader with controller
1	415 card punch with controller
1	501 or 512 printer with controller
2	magnetic tapes from any of the following: 604, 607, 657-2, 657-3, or 657-4, 659-2, 659-3, 659-4.

RELEASE MATERIALS

Release materials for the basic SCOPE 3.4 operating system package consist of the following:

PL1 SCOPE 3.4 program library
PL2 COMPASS 3.0 program library
PL3 6RM 1.0/IS 2.0/DA 1.0 program library
PL4 FORM 1.0 program library
PL5 CE Diagnostics program library
Unconfigured deadstart tape

The SCOPE Program Library Tape contains the source programs for all routines comprising the SCOPE 3.4 operating system. An itemization of a complete SCOPE 3.4 product set deadstart tape appears in the preface of this document. The unconfigured deadstart tape contains only the products SCOPE, COMPASS, 6RM, and SYMPL.

Required supplements to this package are:

PL6 SYMPL 1.0/Maintenance Tools program library
PL7 FORTRAN Extended 4.0 Compiler program library
PL8 FORTRAN Extended 4.0 Object Routines program library

NEW FEATURES

15 Control Points

An installation can select from 1 to 15 control points at system assembly time.

20 Peripheral Processor Support

SCOPE 3.4 will support the hardware capability to have 20 PP's and 24 channels.

Tape Scheduling

Tape Scheduling (in SCOPE 3.3 known as automatic tape assignment) provides more control of system resources. The installation can tailor the system to its needs. The following capabilities are provided:

Preview of a queue of tape-dependent jobs

Dynamic tape drive status determination

Scheduling of tape-dependent jobs based on tape availability

Use of VSN information to accurately identify tapes

Handling of fixed priority jobs

Deadlock prevention

Loader

The SCOPE 3.4 loader directives and control cards allow the full use of alternate and user libraries. Expanded use of overlay structures allows up to 4095 levels of overlays.

The new loader maintains all existing features of the replaced CP and PP loaders, except segmentation. The following features have been added:

Selection of alternate system libraries or user libraries to satisfy externals

Selective load of programs from a file

Greater load map flexibility

Presetting of core

Saving of core image of loaded program

Capability of not satisying selected externals

Expanded user call capability

Multiple entry points to overlays

EDITLIB

EDITLIB has been rewritten to accommodate the new loader library structures, and it includes the capability to generate and modify user libraries.

ECS I/O Buffering

Through REQUEST card or macro declaration, ECS may be used to block the user's input/output records into large ECS buffers thereby reducing the number of accesses to rotating mass storage devices.

ECS is divided into:

System area: system information, flaw tables

Paged allocatable area: I/O buffering area, resident library, swap files

Direct access area: user area, as in SCOPE 3.3

Integrated Scheduler

The job scheduler for SCOPE 3.4 handles the normal batch jobs and in addition, controls the scheduling of all other jobs submitted to the system through remote facilities. The scheduler assigns a number, the job descriptor table (JDT) ordinal, to each job in the queue rather than just the control point number. Based on a set of job class characteristics, set by the installation, jobs may be swapped in and out to resolve resource conflicts or provide additional core for processing of high priority jobs; any job waiting for some external event (terminal I/O, tape assignment, PF availability, etc.) also will be swapped or rolled out. Through the use of job priority computation and job swapping, use of central memory and the central processor is improved. Jobs requiring high priority consideration will be placed in an express queue.

ANSI Label Support

Additional user header and trailer labels conform to the ANSI specifications.

SCOPE 3.4 conforms to an ANSI label standard which supersedes the ANSI standard followed by SCOPE 3.3. Under the new standard, density of the label data is the same as that of subsequent data.

Tapes with labels created under previous systems must be identified as Z labels if label and data densities are not the same. SCOPE 3.3 and 3.4 U labels are not identical. Refer to Appendix E of the SCOPE 3.4 Reference Manual.

Private Pack Support

The disk pack utilization has been expanded in the form of Sequential Packs. A multi-pack file no longer is restricted by the number of disk drives available. The only requirement is that the pack currently in process be mounted. The multi-file capability is not available on Sequential Packs.

Support of SCOPE Format on 659 Transports

This feature allows such capabilities as having the deadstart tape created at 800 or 1600 bpi on a 9-track tape.

STATUS Request

A new macro provides the user with information concerning the remaining resources in the system. This information will include file characteristics and unused rotating mass storage.

Extended Error Processing

An additional capability is provided in the area of error status. By setting a bit in the FET, the user can receive additional detail on the nature of any error condition.

Permanent Files

The PF utilities now read/write non-stop to disk and tape.

A new PF utility, TRANSPF, allows the transfer of permanent files and/or the permanent file tables from one public mass storage device to another.

The PF routines re-use RBTC space.

ID hashing is used to search the permanent file directory (PFD).

Automatic archiving and retrieval of permanent files from tape is provided.

Automatic swap-out is provided for jobs waiting for access to files, APF space, or the PF utilities.

Two functions are new. SETP makes it possible to pre-position files at attach time, and ALTER allows end of information for a permanent file to be set to a current position.

PFDUMP

PFDUMP is a Fortran program which calls a PP program PFD, to dump the permanent file tables (PFD and RBTC). The PP program asks for two operator n.GO responses before completion. If an installation does not want this program as part of their running system due to fear of the loss of security to their permanent file system, they should delete it from their program library.

Deadstart and Recovery

Expanded recovery capabilities and reliability

RMS label initialization independent of permanent files

Option to logically turn off a CPU at deadstart time

Exchange jump package dump

Recovery of reconfigured CMR

CORRECTIONS

All eligible PSR corrective code published through Summary 312 has been added to the SCOPE 3.4 program library.

NOTES AND CAUTIONS

To prevent degradation in system performance, RMS devices used for SYSTEM, PFD and/or PF residency should be placed on double ranked channels (24B-33B) in machines having more than 10 PRU's and 12 channels.

Under SCOPE 3.4, it is no longer necessary or permissible to create an EST or RBR entry for ECS.

Sequential pack files (DP or 2DP on REQUEST card) cannot be ECS I/O buffered.

CAUTION: The MMTC controller, operating at 1600 bpi in conversion mode, initiates a memory reference to controller memory cell 377 on detecting the first character of the postamble. Consequently, the flag bit always must be set in this cell; otherwise memory flag bit errors will occur. This problem is resolved when ECO/FCO CA26461(3518) and CA26462(3528) are installed.

When SCOPE 3.4 is run on a 6500 or 6700 using IP.XJ=1 or 2, FCO CA23065 must be installed (to prevent both CPU's from being in monitor mode simultaneously).

If the OUTPUT file is rewound but no other action is performed on the file, the OUTPUT file will be evicted; a skip to EOI is not performed prior to writing the job dayfile on OUTPUT.

An installation defined limit is placed on the amount of mass storage that can be allocated to a job; jobs exceeding this limit will be terminated. The LIMIT control card can be used, however, for a job that is expected to exceed the installation limit.

To free disk space following a disk overflow the alternatives are:

KILL the job at one or more control points

Allow jobs at control points to loop until disk space becomes available as other jobs leave the print queue

A combination of the above

If permanent files are to be recovered, allocatable devices that are to be unavailable should be turned off at deadstart time, rather than having their ESTs zeroed out.

Mass storage accounting is modified slightly. Accounting of mass storage accesses for CM read/write are based on the values assigned by the macro ACCOUNT in CMR. The range of values acceptable to the ACCOUNT macro is 0 to 32. 1SP/1EP rounds down the ACCOUNT values to a power of 2; resulting in effective values of 0, 1, 2, 4, 8, 16, or 32. These values give accounting of 0, .25, .5, 1, 2, 4, or 8 milliseconds per PRU transferred.

Permanent files cannot reside in ECS.

Assignment of long stranger (L) tapes is prevented if SCOPE is assembled with IP.ECSB set to one (via code present in REQ under identifier SC40035). Installations with ECS active must avoid any use of L tapes, as 1MT (the L tape driver) could be locked out of central memory by an ECS transfer, resulting in numerous lost data errors or a system hang.

The COPY utilities do not support the random file features of SCOPE 3.4.

If a reverse function (SKIPB, BKSP, etc.) reaches beginning of reel, the operation is considered complete, as are READSKP and SKIPF when they reach end-of-reel if UP is on. If UP is off, these operations automatically go to the next reel. The number of skips not completed is returned in the FET extension if extended error processing is selected.

If two users submit dependent job strings with the same dependency identifiers and the same job name, SCOPE cannot distinguish between them. Therefore, the installation must prevent duplication by scanning job names for identical dependency identifiers.

If DISPOSE of a file is executed prior to the end of a job but the job continues to use that file name, two files are created and there is no way for the operator to differentiate between them for such operations as EVICT or ENPR.

INSTALLATION PROCEDURES

Installation of SCOPE 3.4 requires customizing to conform to the site's hardware and software specifications as follows:

Selection of general installation parameters within IPARAMS

Selection of tape processing installation parameters within CIOCOM

Choice of tape scheduling option

Configuration of CMR

Consideration of scheduling parameters

Determination of deadstart installation parameters

Selection of permanent file installation parameters

Selection of ECS parameters

Construction of a deadstart tape

General installation parameters related to SCOPE are defined within the COMDECK IPARAMS. IPARAMS is listed in the routine CMR. Other installation parameters are described elsewhere in this and other sections of this document. Assigned (default) values and descriptions are listed below. Changes to the default values listed below should be made at IPARAMS.15. The first parenthetical value is the default value as set on the released program library. Additional parenthetical values, where given, have also been tested.

The default values of the IPARAMS configuration parameters are defined with the CEQU or CMICRO macros, so that an installation can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications should precede them.

Symbols can be defined be EQU or CEQU except for IP.SYSL1, IP.VER and IP.SYSE, which are micros and must be defined by CMICRO.

The following list constitutes the extent of installation changeable symbols in IPARAMS. Certain symbols present in IPARAMS in SCOPE 3.3 have been made installation invariant in SCOPE 3.4.

IP.CMU (0)

If nonzero, Compare/Move Unit hardware is present.

IP.CP (6)

If set to 6, all BCD cards will be punched as if by an 026. If set to 9, all BCD cards will be punched as if by an 029. This setting must agree with the mode set by IP.CR.

IP.CR (69D)

If set to 6, all BCD cards will be read as if punched by an 026. If set to 9, all BCD cards will be read as if punched by an 029. If set to 69, all BCD cards will be read as if punched by an 026; however, if a job card or a 7/8/9 end-of-record card has 29 punched in columns 79-80, all following BCD cards in that job will be read as if punched by an 029, until a following 7/8/9 end-of-record card changes the mode again. If set to 96, the inverse is true: 029 is default and job and EOR cards may switch to 026. The card reader routine, 2RC, treats all level 17 EOR cards as end-of-file for compatibility with JANUS.

IP.CSET (IP.C63)

Defines character set to be used throughout the system; it must be set to one of the following values:

IP.C63 63-character set, same as SCOPE 3.3 and earlier systems.

IP.C64.1 CDC standard 64-character set.

IP.C64.2 64-character set where many special symbols have a different definition and graphic than the 63-character set.

The CEQU statements for IP.C63, IP.C64.1, and IP.C64.2 must not be altered by the user.

Each of the three character sets are described in detail in Appendix A of the SCOPE 3.4 Reference Manual, Publication No. 60307200.

IP.CSET must have a constant value as all products to be added to the deadstart tape are assembled.

IP.DSP (PEABCD)

Macro representing a string of two-character mnemonics; each represents special characteristics for a disposition of output files. Example: IP.DSP CMICRO, (1P2P3P) indicates special mnemonics for one-part, two-part, or three-part paper.

IP.ECSB (0) (1)

If zero, the ECS extensions code is not assembled. If non-zero, the ECS extension code is assembled; and the ECS installation parameters are activated.

IP.GP250 (0)

If non-zero, the graphic display package is part of the system.

IP.IMUL (0)

If nonzero, Integer Multiply hardware is present.

IP.IQD (6)

Input queue delay. The lower 6 bits of the input queue priority are incremented by one every 2**IP.IQD seconds (0 to 11). See Scheduling Parameters.

IP.IQPW (3)

Input queue priority weight. The effective input queue priority is (P*(2**IP.IQPW))+A. P= job card priority, A = age addend (0 to 6). See Scheduling Parameters.

IP.LVF (70B)

Lowest fixed priority. Normally, it should be greater than IP.MPR. Since a fixed priority does not age, it is normally higher than can be specified on a job card. Thus it can be created only by operator action.

IP.MAP (10B) (2)

Default loader MAP option. 0 = MAP(OFF), 1 = (MAP=S),
2 = (MAP=B) (corresponding to SCOPE 3.3 MAP (PART)),
4 = (MAP=E), 10B = (MAP=X) (corresponding to SCOPE 3.3 MAP (ON))
S - Loader statistics and error messages only
B - S Option plus block names, addresses and lengths
E - B Option plus entry point list
X - E option plus a cross reference list of external references

IP.MCPU (1)

Installation option to define maximum number of CPUs to be used by system. The value 1 will produce the most efficient code for use on a single CPU. The system will run on a 6500, but it will use only one CPU. The value 2 will produce an MTR which will run on a 6500 or 6700 using both CPUs or on a 6400 or 6600 using one CPU.

IP.MECS (0) (730B) (40B)

Maximum number 0 to 7777 (octal) of 1000 word (octal) blocks of ECS direct access that may be assigned in response to a job card EC parameter. This value determines whether sections of code are to be assembled within the system to handle ECS allocation.

IP.MFL (140000B) (300000B)

Maximum amount of central memory field length that may be assigned to a user job. A user cannot request more than IP.MFL field length on job card or with MEM or RFL. Each installation must set IP.MFL less than (machine FL-CMR size-RBT area size-2*IP.POSFL*100B). If value is too large, a job swapped out waiting for a large field length might be locked out and never be swapped in, even if the machine is idle.

IP.MMS (100B)

Maximum mass storage limit that may be specified by PRUs/100 (octal) on a LIMIT card.

IP.MPPU (10D)

The maximum number (7-20) of peripheral processors in the configuration of any of the CMRs on the deadstart tape.

IP.MPR (20B)

Maximum priority a user can specify on his job card. Range (1 to 70B). Normally, it should be less than IP.LVF.

IP.MSCT (0)

Maximum decimal number of messages (1 to 4095) that may be entered into the dayfile by a single job. Only messages sent through MSG are counted. If zero, no maximum will be considered.

IP.MTL (77777B)

Maximum CP time limit in seconds, 1 to 77777 (octal), that may be assigned to a job.

IP.NDFS (1) (2)

Number of dayfile copies on output. Up to 4095 decimal may be specified.

IP.NJFL (20B)

FL/100B assigned to batch jobs when first assigned to a control point. Range (1 to IP.MFL). The default value allows satisfaction of job setup utilities.

IP.OQD (10B)

Determines delay before incrementing priority of a job in the output queue.

IP.POSFL (5)

Field length/100B reserved for use by 1SO for requesting positive field length. Positive field length is not available to user jobs and can be considered part of CMR. Positive field length is allocated internal to the system for swapout use only. Range (4 to 10B).

IP.SECS (0)

Default number of direct access ECS blocks (1000 octal words) to be assigned to a job if not declared on job card; range zero to IP.MECS.

IP.SFL (50000B)

Default central memory field length (octal) to be assigned to a job if not declared; 100 to IP.MFL.

IP.SMS (0)

If non-zero, the default mass storage PRU limit a job can use, divided by 100 (octal). All jobs therefore proceed as if a LIMIT card with value IP.SMS were in the job deck. Refer to the LIMIT card in the SCOPE Reference Manual.

IP.SPR (10B)

Default priority given to a job if no priority specified on job card. Range 1 to IP.MPR.

IP.STL (100B)

Default time limit in octal seconds (1 to IP.MTL) to be assigned to a job if not declared on the job card.

IP.TCPUB (4)

The number of time units that should be accumulated on CPUA for the equivalent of 4 time units on CPUB. The default value should be used on a 6500; for a 6700, the value 9 should be used to indicate CPUA to be 2.25 times as fast as CPUB.

IP.TYPE (6600)

Determines the type of central processor to be used by the system (6600 (CDC CYBER Model 74) or 6400 (CDC CYBER Model 72)) for generation of optimal code. Acceptable values are 6400 and 6600.

IP.XJ (-1) (1)

Values (pertaining to routine CMR only)

0 Computer does not have the central exchange feature; central monitor is simulated.

1 Make use of central exchange jump feature.

2 Make use of central exchange jump feature including the MAN instruction.

-1 Central exchange jump feature is not to be used. An exchange jump protection program is included to protect the system against an accidental execution of an exchange jump instruction.

When multiple CMR's are assembled, any of which are assembled with a value of 0 or -1, MTR must be assembled with a 0 or -1 value.

IP.YMD (MDY)

Micro which shows format of date to be typed in at deadstart. The six possible permutations of the letters MDY constitute the range of this parameter.

The IPARAMS common deck also contains symbols IP.ILCMD, IP.IUSED, IP.1M1, IP.1WB, and IP.1ZZ. The INTERCOM 4.1 section contains a description of these symbols.

The default values of the CIOCOM configuration parameters are defined with the CEQU or CMICRO macros, so that an installation can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications should precede them.

Symbols can be defined by EQU or CEQU except for IP.SYSL1, IP.VER and IP.SYSE, which are micros and must be defined by CMICRO.

Installation parameters specifically oriented to tape processing are defined within the COMDECK CIOCOM. CIOCOM is listed in the routine CMR. Changes to default values should be made at CIOCOM.6. Assigned (default) values, other tested values, and descriptions are as follows:

The following constitutes the extent of installation changeable symbols in CIOCOM. Certain symbols present in CIOCOM in SCOPE 3.3 have been made installation invariant in SCOPE 3.4.

IP.CBKSP (0)

If one, controlled backspace is available in all controllers; if zero, it is not installed.

IP.NBCD (0)

9-track default conversion mode (0=ANSI, 1=EBCDIC)

IP.NBRK (0)

If zero, system noise records are used in write recovery; if one, they are not used.

IP.NDEN (2)

Density for label and data on 1/2 inch 9-track tape, if not declared on REQUEST or LABEL card.

IP.NOISE (3)

Maximum decimal number of 12-bit bytes in a noise record on 7-track S and L tapes or 9-track Conversion Mode (S-format) magnetic tape. A record less than or equal to IP.NOISE is discarded.

IP.NOIS9 (17D)

Maximum decimal number of 8-bit bytes in a noise record for packed mode on 9-track tapes. A record less than or equal to IP.NOIS9 is discarded. Default (17D) is the ANSI standard.

IP.RCYC (3R000)

Retention cycle (0-999) for calculating tape label expiration date when no retention cycle is given; 999 indicates permanent retention. The address field of the symbol definition should contain 3Rxxx where xxx defines retention cycle; leading zeros need not be written.

IP.RPE1 (12D)

Total decimal number of read parity retries on a single record (must be less than 60).

IP.RPE2 (8)

Decimal number of read parity retries accomplished by backspacing over the previous three records then reading forward in an attempt to recover (IP.RPE2 must be less than IP.RPE1.)

IP.TDEN (0) (2)

Density for both label and data on 1/2-inch 7-track magnetic tape if not declared on LABEL or REQUEST card: 0=556 bpi, 1=200 bpi, 2=800 bpi.

IP.TRYS (10D)

The number of unsuccessful attempts to locate the last good record, in excess of the minimum before declaring parity error irrecoverable. The minimum is equal to zero if IP.NBRK=1 or number of skips done + 2 if IP.NBRK=0. This value should not exceed 62D.

IP.TSG (2617B) (6017B)

Tape scheduling options as follows:

Name	Bit		
S.AUTO	1	1	Enable automatic tape assignment according to LABEL or VSN specification
S.URES	1	1	Enable job scheduling based on unit availablility
S.PRES	2	1	Enable pre-staging features (the VSN preview of the P display)
S.2LBP	3	0	Only ANSI labels will be accepted and written
		1	Two label formats (ANSI and 3000) are defined
S.AUNR	4	1	Allow auto assign to not ready unit
S.AUUL	5	1	Automatic unloading of tapes if necessary to make automatic assignment
S.SCUL	6	1	Write-enabled, unlabeled tapes will be considered as usable for automatic assignment as scratch tapes
S.SCEL	7	1	Write enabled expired labeled tapes will automatically be considered for assignment as scratch tapes
S.SCBL	8	1	Write-enabled blank labeled tapes will automatically be considered as scratch tapes
S.PREA	9	0	Give warning if tape job has no VSN information
		1	Preabort such jobs
S.OCJI	10	0	Job initiation is based on tape drive availablility; total demand cannot exceed number of drives logically.
		1	Job initiation allows tape drive overcommitment,
S.UEOJ	11	1	Unless specified otherwise on REQUEST or LABEL cards, all tapes will be unloaded at end of job
S.PSON	12	1	Prestaging feature (bit 2) set on at deadstart time. This is equivalent to the STAGE ON typein
	13-16		Reserved for CDC
S.NOOR	17		Operator cannot override VSN card
S.DBUG	18		Enable label debug code (4LB,4LC)
	19-20		Reserved for CDC

Note: Use of bit 4 assumes that the basic auto-assign option (bit 0) is on. Likewise, bit 5 assumes that both bits 0 and 4 are on. Assembly errors result when setting of these bits is inconsistent.

Cross Reference Listing

The following cross-reference listing shows the SCOPE routines that reference each IPARAMS and CIOCOM symbol.

*******										
IPARAMS										
*******										
IP.CMU										
IP.CF	1IR									
IP.CR	1IR									
IP.CSET	1RN	1IR	1MT	1P2	1P4	1P3	1P1	1WI	1RV	1NO
	1N2	1N3	1R2	1R3	1TF	1WS	1NW	8T3	2TB	1CR
	1CT	1CS	1RT	1RS	1NR	1W9	1C9	1R9	7T1	7T2
	2LP	2PC	2RC	4LB	6LC	4LC				
IP.DSF										
IP.ECSE	CEA	CED	STL	IRCP	CMR	DSD	1AJ	7EC	1EJ	1SF
	REQ	1SI	1SO	CEM	TDS	LOAD	LOADM	LCADU	LOADLM	UCLOAD
IP.GP250	CMR									
IP.ILCMD	DSD									
IP.IPUL										
IP.IGC	1RN									
IP.IGPW	1IB									
IP.IUSID	CMR									
IP.LVF	1PF	1RN	1MH	1IB	2TJ	TDS				
IP.MAF	1AJ	1IB								
IP.MCPL	CED	CMR	MTR	DSD	1EJ	1SP	DIS	1IB	1SO	2TJ
	RPV									
IP.MECS	CMR	MTR	DSD	CIS	1IB	2TJ				
IP.MFL	1AJ	MEM	1SO	2TJ	TDS					
IP.MMS	1AJ									
IP.MPPU	CMR	MTR								
IP.MPR	2TJ	DSP								
IP.MSCT	MSG	1IB								
IP.MTL	2TJ									
IP.NDFS	1EJ									
IP.NJFL	CMR									
IP.OGC	1RN									
IP.PCSFL	MTR	1SO								
IP.SECS	2TJ									
IP.SFL	1PF	1SI	2TJ	TDS						
IP.SMS	1IB	1SI								
IP.SPR	2TJ									
IP.STL	2TJ									
IP.TCPUB	CMR									
IP.TYPE	CMR									
IP.XJ	CMR	MTR								
IP.YMC										
IP.1M1	DSD									
IP.1WE	DSD									
IP.1ZZ	DSD									
******										
CICCOM										
******										
IP.CBKSP	1MT	1WI	1WS	1NW	1W9					
IP.NECC	1EJ	REQ	6L3	1TS						
IP.NBRK	1MT	1P2	1P3	1P1	6BW					
IP.NCEN	REQ									
IP.NOISE	CED	1MT	1P1	1NO	1N2	1N3	1R2	1R3	1TF	1WS
	1NW	2TB	1RS	1NR						
IP.NOIS9	1P1	1NO	1N2	1N3	1R2	1R3	1TF	1NW	2TB	1NR
IP.RCYC	4LB	4LC								
IP.RPE1	1RN	1MT	1P2	1P4	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW	8T3	2TB	1CR	1CT
	1CS	1RT	1RS	1NR	1W9	1C9	1R9	4LB	4LC	
IP.RPE2	1RN	1MT	1P2	1P4	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW	8T3	2TB	1CR	1CT
	1CS	1RT	1RS	1NR	1W9	1C9	1R9	4LB	4LC	
IP.TDEN	CED	REQ	1RQ							
IP.TSG	CED	CMR	DSD	CIO	1PF	1EJ	3DD	1RN	4ES	REQ
	2TA	1RP	1CL	2TC	1TO	1OP	1PK	3PK	1BT	1MF
	1MH	1LT	1MT	1P2	1P4	1P3	1P1	1WI	1RV	1NO
	1N2	1N3	1R2	1R3	1TF	1WS	1NW	8T3	2TB	1CR
	1CT	1CS	1RT	1RS	1NR	1W9	1C9	1R9	1IB	2TJ
	4LB	4LC	6WM	1TS	REQUEST					

Tape Scheduling options that may be selected by the installation are implemented by the use of conditionally assembled code. The bits in IP.TSG are tested at assembly time to determine the exact nature of the programs that comprise tape scheduling. For example, bit S.SCBL in IP.TSG governs the automatic scratch status of blank labeled tapes. If the bit is on, blank labeled tapes will be considered scratch without operator intervention; if the bit is off, scratch status will not be granted automatically.

The bits in IP.TSG can be divided into the 3 general catagories of automatic assignment bits, pre-staging and overcommitment bits, and miscellaneous bits.

Automatic Assignment Bits

The installation can select 4 levels of automatic assignment by setting the 3 bits S.AUTO, S.AUNR, and S.AUUL:

No automatic assignment (all 3 bits off)

Basic automatic assignment (bit S.AUTO on and others off).

Auto assign to not ready unit (bits S.AUTO and S.AUNR on).

Auto assign to not ready unit with unload (bits S.AUTO, S.AUNR, and S.AUUL on).

With only S.AUTO set on, a specific tape will be assigned automatically when the specific tape is mounted. This level of automatic tape assignment is the basic and most generally useful. In the not ready modes, a search is made for the specific tape requested. When it is found, it is assigned, and the job continues. If it is not found, a not-ready tape unit is selected and the operator is instructed to mount the specific tape on the exact unit selected. If no not-ready tapes are available and the unload option is selected a ready tape will be selected and unloaded and the operator will be instructed to mount the required tape on that unit.

Automatic Scratch Status

Three other bits are related to automatic assignment. They are bits S.SCUL, S.SCEL, and S.SCBL. When set, each bit determines a specific type of tape to be considered automatically as a scratch tape. If all three bits are off, the only tapes treated as scratch are those specifically designated by the operator with the command SCRuu (where uu is the EST ordinal).

A job specifies *MT or VSN = SCRATCH in the request for a scratch tape. If any automatic assignment is turned on (bit S.AUTO is set), the system will try to assign a scratch tape automatically to the job. The tape must be mounted on a ready unit with a write ring in place, it must also be designated as scratch as described above, and it must meet these qualifications:

Tapes designated as scratch by the operator

Unlabeled tapes if bit S.SCUL is on

Tapes with expired labels if bit S.SCEL is on

Tapes with blank labels if bit S.SCBL is on

Pre-Staging Bits

Unit Reservation:

Bit S.URES controls the necessity of jobcard tape parameters, without which overcommitment and deadlock prevention are meaningless and pre-staging will not function.

Pre-Staging:

The prestaging option is assembled if bit S.PRES is set. If this option is on, a pre-staging buffer is assembled in CMR, its length is N.VRNBUF*6 (release value gives a 171B word buffer). Installations can change the symbol N.VRNBUF in CMR to change the size of the buffer.

If bit S.PSON is on, it sets up CMR as if STAGEON had been typed after a normal deadstart. Deadstart Recovery preserves the current setting of the STAGEON/STAGEOFF switch.

If bit S.PREA is set, a job that specifies tapes on the job card but has no VSN specifications in the job will be pre-aborted.

Overcommitment:

Bit S.OCJI determines whether or not tape drives will be overcommitted. If the bit is off, the total number of tape drives required by all jobs executing at a given time (as determined by job card tape parameters) cannot exceed the total number of tape drives at the installation. If bit S.OCJI is on, tape drives are overcommited; the total tape requirements of executing jobs can exceed the total number of tape drives at the installation. Deadlock is prevented by an algorithm calculated each time a tape is assigned.

Miscellaneous Bits

Two Label Processors:

If in additon to the ANSI label processor, 4LB, CDC 3000 (Y) labels are to be processed, bit S.2LBP should be on to allow use of the alternate label processor, 4LC.

EOJ Tape Unload:

Bit S.UEOJ causes 1EJ to unload non-scratch tapes at end of job. If any problems are encountered trying to unload the tape, such as tape not ready, the unload attempt will be ignored. This differs from the SAVE (SV on REQUEST card or X=SV on LABEL card) unload processed by 1EJ; 1EJ issues a message that problems exist and continues trying until the operator types in GOuu.

Operator Cannot Override VSN:

With bit S.NOOR off, the operator can assign a tape with a VSN different from the VSN specified by the job; however when this bit is on, a different VSN, is not allowed.

Label Debug:

Bit S.DBUG controls debug code in 4LB and 4LC; use of this bit is not the normal mode of operation. This debug code will produce many messages which show the calls to and returns from the label processors. Such messages may cause other more informative messages to be overwritten.

Option Dependencies (IP.TSG)

The two figures below show dependent bits. Each bit name shown cannot be turned on (or turning it on will have no effect) unless all bit names below it are on. The automatic assignment bits are independent of the Pre-staging and Overcommitment bits and vice versa.

Miscellaneous bits (S.2LBP,S.UEOJ, S.NOOR, and S.DBUG) are independent of each other and of the bits shown below:

Auto Assign Dependencies:

<table>
<tr><td>S.AUUL</td><td></td><td></td><td></td></tr>
<tr><td>S.AUNR</td><td>S.SCUL</td><td>S.SCEL</td><td>S.SCBL</td></tr>
<tr><td colspan="4">S.AUTO</td></tr>
</table>

Pre-Stage and Overcommitment Dependencies:

<table>
<tr><td>S.PSON</td><td>S.PREA</td><td></td><td></td></tr>
<tr><td colspan="2">S.PRES</td><td></td><td>S.OCJI</td></tr>
<tr><td colspan="4">S.URES</td></tr>
</table>

For example, to set bit S.AUUL on, bits S.AUNR and S.AUTO must be on; not having S.AUTO on will cause assembly errors. S.PREA is dependent on S.PRES and S.URES but not on S.OCJI.

CMR CONFIGURATION PARAMETERS

The default values of the CMR configuration parameters are defined with the CEQU or CMICRO macros, so that an installation can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications should precede them.

Symbols can be defined by EQU or CEQU except for IP.SYSL1, IP.VER and IP.SYSE, which are micros and must be defined by CMICRO.

All the CMR configuration parameters are grouped together near the beginning of CMR. Changes should be made at CMR.964.

General parameters should be tailored to suit the needs of each installation; default values are shown in parentheses:

L.EST	(40B)	Length of equipment status table (≤ 77B)
L.CST	(50B)	Length of channel status table.
L.RPT	(10B)	Length of removable pack table.
L.INS	(0)	Length of installation table. Size, definition and usage of an installation table is completely controlled by the individual site. No SCOPE 3.4 product set program makes reference to the installation table.
L.FNT	(2200B)	Length of file name table
L.RMS	(10B)	Length of the C.E. RMS diagnostic table
L.SEQ	(10B)	Length of diagnostic sequencer table
LE.DFB00	(400B)	Size of system dayfile buffer
LE.DFBXX	(100B)	Size of control point dayfile buffer
LE.CERFB	(40B)	Size of hardware error file buffer
N.CP	(15D)	Number of control points (1 to 15D)
N.DEVICE	(3)	Number of controllers for allocatable devices; 1 for each 821 (2 units), 1 for each 814 (2 units) 1 for each 6603 unit, 1 or 2 for each 6638 unit, and 1 for each drum controller (which may drive more than one drum) and each disk pack controller (which may drive more than one disk pack unit). A 6638 with standard option 10037A is two controllers with one unit each (both 0).
N.RBR	(3)	Number of record block reservation tables; normally 1 for each 6638 unit, 1 for each drum unit, 1 for each 854 disk pack unit, and 2 for each 6603 (one each for inner and outer zones), 8 for each 821, 1 for each 814 unit, and 1 for each 841 disk pack unit.
N.RQS	(40D)	Number of request stack entries

N.VRNBUF	(20D)	Number of entries in VSN buffer. Each entry is 6 words long and represents one line of job VSN information in the P display.
IP.SYSL1	(SCOPE 3.4)	System label (up to 20 characters); the first character must be blank.
IP.VER	(RELEASE)	System version identifier (up to 10 characters)
IP.SYSE	(07/30/72)	System generation date (up to 10 characters)

Parameters set to configure CMR for the permanent files are:

N.ESD	(32D)	Number of entries per sub-directory
N.RBTC	(20D)	PRU number (octal) in RBTC
N.SD	(11D)	Number of sub-directories
L.APF	(30D)	Length of APF table (2 word entries)

INTERCOM parameters:

L.ITABL	(19)	Length of INTERCOM table

Scheduler parameters: (See page I-1-50)

L.ECSSWP (3)

Swap to ECS mask: The value of this symbol determines which jobs get swapped to ECS. It is a mask of bits; each bit set defines a set of jobs that will be swapped to ECS. If all bits are off (L.ECSSWP=0), only MUJ jobs and INTERCOM jobs swapped out on a quantum go to ECS. In addition, the following bit settings cause the particular types of jobs to be swapped to ECS.

Bit 0	INTERCOM job at end of a command
Bit 1	Batch jobs swapped out on a quantum
Bit 2	Swap all INTERCOM jobs to ECS
Bit 3	Swap all batch jobs to ECS.

L.SCHJCA	(20B)	Length of Job Control Area. Must be a multiple of 8. (Needs to be changed only if new classes are added.)
L.SCHJDT	(400B)	Length of Job Descriptor Table. Must be a multiple of 8.

Establishing a CMR for an installation requires inserting information about the CMR configuration parameters and tailoring the EST, DST, RBR, and RPT tables. With 3.4, up to eight different CMR configurations, each with unique EST, DST, RBR and RPT tables, may be placed on the deadstart tape.

Equipment Table Configurations

Four equipment tables must be tailored for each installation: EST (equipment status table) DST (device status table), RBR (record block reservation table), and RPT (removable pack table) table. The instructions for creating the EST entry differ between allocatable and non-allocatable devices.

The EST may be altered from the console. DST, RBR, and RPT entries may not be altered from the console; as CMR changes in size with the addition or deletion of entries.

The EST may be tailored to any configuration by using the macros described in this section. Its size may be greater than or equal to the number of hardware units present in the configuration. However, it may not exceed 77(octal) since an EST ordinal must be no more than six bits. Since the first word of the EST may not be used, the first equipment ordinal is 01.

Table Structures

The CMR tables are defined by the TABLE macro.

The sequence of the macro calls defines the sequence of the tables generated in CMR. This sequence can be altered by an installation, but the following constraints must be observed:

1. Origins of EST, CST, FNT and DAT must be located under 10000 (octal).
2. The RQS and DST table must be located under 20000 (octal).

RMS Devices:

The EDST macro causes a one-word EST entry to be constructed where the macro statement occurs. If no EDST macro with the same channel 1, channel 2, and controller has been assembled, a DST and corresponding DAT entry will be constructed. Numeric arguments must not have leading zeros. EDST macro entries should be made at CMR.2108. The macro format is:

```
name    EDST  type,chan1,chan2,contr,unit,sysres,pf,onoff,priv.
```

name — Any combination of up to 6 alphanumeric characters

type — Two-letter hardware code:

AA	6603 disk	AF	814 disk
AB	6638 disk (standard option 10037A required)	AL	821 disk
AC	6603-II disk	AM	841 disk pack
AD	865 drum	AP	854 disk pack unit

An EDST macro entry is not permissible for ECS in SCOPE 3.4

chan1 — Primary access channel for device (assumed decimal)

chan2 — Secondary access channel for device (decimal). Only one channel can be assigned to each mass storage controller.

contr — Controller number for device

unit — Unit number (decimal) for device

sysres — If non-blank, system will be deadstarted onto this device

pf — If non-blank, permanent files may reside on this device

onoff — If non-blank, device is off initially; otherwise, device is on initially.

priv — If non-blank and type is AM or AP; the device will be made a private device.

To avoid degradation in system throughput, the system resident device should be on a channel separate from any other equipment.

Equipment (controller) numbers for 814, 821, 854, 841, and 865 devices must be 4, 5, 6, or 7.

Disk pack entries assembled into the EST for the installation CMR should be designated as logically free, so the operator can make them public or private as required. Disk packs which are to be used as private devices must be so configured when the CMR is assembled. Operator defined private packs will cause a system hang at first reference.

CMR Mass Storage Flaw processing

Input to the FLAW macro in CMR is a 4-digit (octal) PB (physical block) number. Do not append B to the number. The first PB is number zero. Flaws must be entered in ascending orders. CMR sets flaw bit status of all RB's necessary to cover the PB specified. FLAW macro entries should be inserted at CMR.2375. Given the address of a bad spot in controller address format, the PB containing the bad spot can be computed as follows:

For the 6638, 6603 and 865, no formula is required to compute the PB from a physical address. All that is needed is to decompose the physical address into the 12 bit value described below.

```
6638          bit 0      =   0 if sector plus head group is even
                         =   1 if sector plus head group is odd
              bits 1-5   =   head group
              bits 6-10  =   cylinder
              bit 11     =   0

              Example:  bad spot at position (cylinder) 1, head group 30
                        decimal sector 9 corresponds to PB 175(octal) so
                        6638  FLAW  0175

6603          bit 0      =   0 if sector number is even
                         =   1 if sector number is odd
              bits 1, 2, 10  form the 3-bit head group number
                             bit 1 = 0 if head group even; 1 if odd
                             bit 10 = 0 if outer zone; 1 if inner zone
              bits 3-9   =   head position
              bit 11     =   0

              Example:  bad spot at position 108, head group 4;
                        sector 6 corresponds to PB 3540(octal) so
                        6603  FLAW  3540

865 drum      bits 0, 1  =   head subgroup
              bits 2-7   =   head group
              bit 8      =   0 if sector is even: 1 if odd sector/3 is odd
              bits 9-11  =   zeros

              Example:  bad spot at head group 60 (decimal), head subgroup 3;
                        sector 119 corresponds to PB 763(octal)
```

A formula is required to convert an 814, 821, 841, or 854 physical address into a PB. The formula and examples are as follows:

```
814           PB = cylinder*22+N/372+11*E

              where N bits 0-4 are sector number +
                      bits 5-11 are track.
              Remainder of N/372 is discarded
              E = 0 if N/3 is even and 1 if N/3 is odd

              + the last 4 sectors of each cylinder are not used;
              no PB for track 127, sectors 28-31.

              Example:  bad spot at cylinder 59, track 85, sector 9
                        PB = 59*22+(2729/372)+11=1298+8+11
                             = 1317 = 2445(octal) so 814  FLAW  2445
```

821 PB = cylinder*4+stack*2+E
where E = 0 if sector is even
= 1 if sector is odd
in controller address first byte, stack is bit 2^0, cylinder is 2^1-2^9. Sector is in address second byte bits 2^0 thru 2^5

Example: bad spot at controller address 0017 0002 (octal)
corresponds to PB 36 (octal) so 821 FLAW 36

841 PB = cylinder * 5 + N + (track + 2N)/8
where N = 0 if even sector
N = 2 if odd sector
Remainder of division by 8 is discarded

Example: bad spot at cylinder 4 track 8; sector 10
corresponds to PB 25 (octal)

854 PB = cylinder*13+N/24+M where N bits 0-3 = sectors
bits 4-7 = track

M computed as follows:
Let I = remainder of N/24
R = remainder of I/6
M = 0 if R < 3
M = 6 if R ≥ 3 and I < 15
M = 7 if R ≥ 3 and I ≥ 15

*sectors 12 thru 15 of track 9 on every cylinder are unused.

EXAMPLE: cylinder 7, track 3, sector 15 (decimal): N/24=63/24=2+15/24
I = 15 15/6=2+3/6, R=3. M=7
PB = 7*13+2+7=91+9=100=144 octal.

Deadstart

The input to the FLAW type-in during deadstart is an RB number, the first RB being number 1. When a parity error is encountered during system operation, the RB number, RBR ordinal, and corresponding address is recorded in the system dayfile.

RBR Macro for RMS Devices

The first header word of each RBR contains a 6-bit allocation style and a default bit which were parameters to the RBR macro when CMR was assembled.

Several RBR's can be generated for a single unit, each describing a unique area of the device. If RBR's have a unique allocation style for each RB size, this allocation style can be used to direct a file to the RBR with the desired RB size: If the default bit is set for an RBR, files with no specific allocation style may be assigned to that RBR.

One single equipment can be represented by any number of RBR tables (minimum of two for 6603) except for removable devices, which can have only one RBR per unit using the default RB size. Each RBR refers to a different area of the equipment. An RBR table is a single bit string of variable length, up to a maximum of 4095 bits, except for the first RBR of a system or PFD device, which must not exceed 2048 bits. Within an RBR table, one bit represents one RB and the number of PRU's per RB is constant throughout this table. RBR's are added to CMR by inserting RBR macro statements into CMR at CMR.2380. These statements have the form:

name RBR count,prurb,alloc,default

name must agree with name in associated EDST statement for same device

count — Number of available record blocks in this RBR. The counts for all the RBRs on one unit must be set so that they do not exceed the size of the device. The number of RBs required to describe one unit of a device using RB size = PB size is shown in the following table.

prurb — Number of PRU's per record block. If not specified, the PB size will be used. The record block size may not be less than 1/32 of the PB size or larger than 32 times the PB size.

Device	Mnemonic	Standard RB (PB) in PRU	Corres. RBR Size in Bits
6603 disk	AA, AC	50 inner	1024*
		64 outer	1024
6638 disk	AB	50	2048
865 drum	AD	21	512
814 disk	AF	62	2816
821 disk	AL	320**	2048
841 pack	AM	56	1000
854 pack	AP	4	2600

*2 RBR's required

**The RB size for the 821 must not exceed 40 PRUs at time of release.

alloc — Specifies allocation style: A value, range 0-77B used to specify RMS areas having particular RB sizes.

default — If non-zero, files with no allocation style specified will be allowed on this device

Note: On previous versions of SCOPE, scratch and system files could be assigned to an RBR with allocation style zero without the default bit set. On SCOPE 3.4, the same RBR must have the default bit set if scratch and system files are to be assigned to the device.

EXAMPLE: An RBR for the 6638 with PRURB=50 requires a count of 2048 to describe one unit. If PRURB is set to 100, the count must be reduced to 1024. If two RBRs are defined for one unit of the 6638, both using PRURB=50, the sum of the counts must not exceed 2048.

The following table reflects file assignment to an RBR depending on allocation style and the default bit setting in the RBR header.

Y	file may be assigned to this RBR
N	file may not be assigned to this RBR
d	default bit
AL	allocation style
PDE=1	file request specified: PF, device type, or EST ordinal. The RBR in question has already passed these tests.

RBR Default Bit and Allocation Setting

Requested Values	d=0, AL=0	d=0, AL=x	d=1, AL=0	d=1, AL=x
PDE=0 AL =0	N	N	Y	Y
PDE=1 AL =0	Y	N	Y	Y
PDE=0 AL =x	N	Y	N	Y
PDE=1 AL= x	N	Y	N	Y
PDE=0 AL= z	N	N	N	N
PDE=1 AL =z	N	N	N	N

With allocation style, installations can differentiate record block sizes.

Non-Allocatable Devices

The macro used for creating an EST entry for non-allocatable devices depends on whether the device is a 3000 series or a 6000 series device (the console, multiplexer, 6671 Data Set controller, 6676 Data Set controller).

Macro for 6000 Series Equipment:

An EST entry for 6000 series equipment is added by insertion into CMR at the appropriate EST ordinal at CMR.2108 as follows (number base is octal for this macro):

```
h  6000  chan,sync,unit,on/off,ipoint
```

h	Two letter hardware type code DS Display console. See section I-14 for definition of INTERCOM multiplexer types
chan	Access channel number
sync	Synchronizer or equipment number
unit	Not used-retained for compatibility with previous versions
on/off	Switch, on=0, off=1
ipoint	Pointer to INTERCOM MUX subtable

Example:

DS 6000 10,7 Creates an entry for a display console on channel 10, synchronizer 7. Parameters are assumed to be octal.

Macro for 3000 Series Equipment:

An EST entry for 3000 series equipment is added by insertion into CMR at the appropriate EST ordinal at CMR.2108 as follows (number base is octal for this macro):

h	3000	channels B and A, channels D and C, equipment, MMTC, unit, on/off

h
- CR Card Reader (405)
- LP Line Printer (501)
- LQ Line Printer (512)
- MT Magnetic Tape 607,657
- NT Magnetic Tape 659
- CP Card Punch (415)

channel A

If one of the channels is channel zero, it must be described as such in the channel A field. (See II-1-26)

The high order bit of the channel A field has special meaning for magnetic tape equipment:

1) The high order bit set to zero indicates that all tape channels are accessed through 6681 data channel converters or through 6684 converters described and used as 6681's.

2) The high order bit set to one indicates that all tape channels are accessed through either all 6684-I converters or through all 6684-II converters.

3) If IPARAMS symbol IP.CSET is equated to IP.C63, the data channel converters used must be all 6681's, all 6684-I's or 6684-II's used as 6681's as in 1) above.

   If IPARAMS symbol IP.CSET is equated to either IP.C64.1 or IP.C64.2, The data channel converters, used must be all 6681's, all 6684-II's or all 6684-I's used as 6681's as in 1) above.

channel A is the primary channel over which the equipment being defined will be accessed. Only magnetic tape equipment definitions may have channels B, C, and D specified.

channel B	First alternate channel for access to tape transports
channel C	Second alternate channel for access to tape transports
channel D	Third alternate channel for access to tape transports
equipment	Equipment number for device
MMTC	Set to 1 indicates presence of a 3518 or 3528 type magnetic tape controller. Set to 0 indicates magnetic tape controller other than a 3518 or 3528.
unit	Unit number for device
on/off	Switch, on=0, off=1

Examples:

MT 3000 1213,,5,,1	Creates an entry for a 7-track magnetic tape on channels 12 and 13 (with a 6681 converter on both channels) equipment 5, unit 1
MT 3000 1253,,5,,1	Creates an entry for a 7-track magnetic tape on channels 12 and 13 (with a 6684 converter on both channels) equipment 5, unit 1

Equipment numbers of 7- and 9-track magnetic tape controllers must be 4, 5, 6, or 7.

Each 6000 channel can have only one 6681 or 6684 channel converter. This restriction does not exclude the use of a 6000 type controller on the same channel with the 6681 or 6684 converter.

1.. Example modifications to CMR for installation of the following equipment:

6603 disk on channel 0, controller 1, unit 0, which is to be the system resident device and may have resident permanent files

3000 card punch on channel 11, equipment 4

3000 card reader on channel 12, equipment 4

6000 console on channel 10, synchronizer 7

3000 printer on channel 11, equipment 6

Four magnetic tapes on channel 13, 6681 converter, equipment 5, units 0, 1, 2, and 3

865 drum on channel 11, controller 5, unit 0

Three 854 disk pack units on channel 12, controller 6, unit 0, 1, and 2; drums and disk packs are to be initially OFF in the EST.

```
*INSERT CMR.964
N.DEVICE     EQU     3
N.RBR        EQU     6
*INSERT  CMR.2108
6603         EDST    AA,0,,1,0,1,PF
             BSSZ    1          (Standardizes EST ordinals)
CP           3000    11,,4
CR           3000    12,,4
DS           6000    10,7
LP           3000    11,,6
             BSSZ    1          (Standardizes EST ordinals)
MMTC         SET     0
TUNIT        SET     0
             DUP     4,2
MT           3000    13,,5,MMTC,TUNIT
TUNIT        SET     TUNIT+1
865          EDST    AD,11B,,5,0,,,1
854A         EDST    AP,12B,,6,0,,,1
854B         EDST    AP,12B,,6,1,,,1
854C         EDST    AP,12B,,6,2,,,1
*INSERT  CMR.2380
6603         RBR     1024,,,D
6603         RBR     1024,,,D
865          RBR     512,,,D
854A         RBR     2600,,,D
854B         RBR     2600,,,D
854C         RBR     2600,,,D
```

2. Example CMR modifications for installation of the following equipment:

6638 disk; first half on channel 0, controller 1, unit 0, to be the system resident and permanent file directory device; second half on channel 1, controller 2, unit 0

3000 card punch on channel 5, equipment 4

3000 card reader on channel 12, equipment 4

6000 console on channel 10, synchronizer 7

Two 3000 printers on channel 11, equipments 6 and 7

Sixteen magnetic tape units on channels 5, 11, 12 and 13, 6681 converter, equipment number 5, units 0 through 17B

Three 9-track magnetic tape units on channel 7, 6681 converter, equipment number 7, units 0, 1, 2

```
*INSERT CMR.964
N.DEVICE   EQU    2
N.RBR      EQU    2
*INSERT  CMR.2108
6638A      EDST   AB,0,,1,0,1,PF
6638B      EDST   AB,1,,2,0,,1
CP         3000   5,,4
CR         3000   12,,4
DS         6000   10,7
LP         3000   11,,6
LP         3000   11,,7
TNUM       SET    0
MMTC       SET    0
           DUP    16,2
MT         3000   1213B,1105B,5,MMTC,TNUM
TNUM       SET    TNUM+1
MMTC       SET    1
TUNIT      SET    0
           DUP    3,2
NT         3000   7,,7,MMTC,TUNIT
TUNIT      SET    TUNIT+1
*INSERT   CMR.2380
6638A      RBR    2048,,,D
6638B      RBR    2048,,,D
*INSERT CMR.2375
6638A      FLAW   0320
```

DEADSTART PANEL SETTINGS

By setting the toggle switches on the Dead Start Panel, the operator establishes a program that is loaded into memory and executed when the deadstart switch is activated. Required settings are determined by the system configuration and tape records. Channel, equipment, and unit numbers of the magnetic tape drive holding the deadstart tape must be specified according to actual connection and settings on the controller and unit.

Bits 6-8 of either word 3 or 5 indicate the record containing configuration data that will be loaded during deadstart. If only one CMR exists, these bits should be set to zero. The other CMR configurations can be accessed by setting the bits to an octal value that is one less than the configuration number. For example, the seventh CMR is obtained by setting these bits to 6.

The PPO save indicator (word 7) relates to dumping contents of that peripheral processor as the first deadstart action. Unless this bit is 0, the original contents of PPO cannot be dumped because it will be overwritten by the deadstart dump program. When the bit is not set, the contents of PPO will be copied to central memory and dumped from there when the dump routine is executed.

Two Dead Start Panel settings are shown. The first figure shows the standard setting for SCOPE; it assumes that the deadstart tape unit is on channel 0, 12 or 13. The setting in the second is for a tape on any of channels 1-11. For all panel settings, a 1 indicates the up or set position; a 0 indicates the down position. Addresses not shown are irrelevant.

ADDRESS				
0001	1 1 1	1 0 1	0 0 c	c c c
0002	1 1 1	1 1 1	0 0 c	c c c
0003	e e e	r r r	0 0 u	u u u
0004	1 1 1	1 1 1	0 0 c	c c c
0005	0 0 0	0 0 0	0 0 1	0 0 0
0006	1 1 1	1 1 1	0 0 c	c c c
0007	0 0 1	1 0 0	0 0 0	s 0 0
0010	1 1 1	1 0 0	0 0 c	c c c
0011	1 1 1	0 0 1	0 0 c	c c c
0012	0 0 0	0 0 0	0 0 1	0 1 1

The remainder of the panel is irrelevant.

cccc	Tape channel number (0, 12, or 13)
eee	Tape controller number
uuuu	Tape unit number
s	PP0 save switch (1 if PP0 is not to be saved)
rrr	CMR number (000 for first CMR)

ADDRESS				
0001	1 1 1	0 1 1	0 0 c	c c c
0002	0 0 0	0 0 0	0 0 1	0 1 1
0003	1 1 1	1 0 1	0 0 c	c c c
0004	1 1 1	1 1 1	0 0 c	c c c
0005	e e e	r r r	0 0 u	u u u
0006	1 1 1	1 1 1	0 0 c	c c c
0007	0 0 1	1 0 0	0 0 0	s 0 0
0010	1 1 1	1 0 0	0 0 c	c c c
0011	1 1 1	0 0 1	0 0 c	c c c
0012	0 0 0	0 0 0	0 0 1	0 1 1
0013	0 0 0	0 0 0	0 0 0	0 0 0
0014	1 1 1	0 0 1	0 0 1	0 1 0

cccc	Tape channel number (1-11)
eee	Tape controller number
uuuu	Tape unit number
s	PP0 save switch (1 if PP0 is not to be saved)
rrr	CMR number (000 for first CMR)

Dead Start Panel to Read from Tape on Channels 1-11

MAIN DEADSTART DISPLAY

The Operator Option Matrix (Main Deadstart Display) for SCOPE 3.4 is described in the SCOPE 3.4 Operator's Guide. To provide additional flexibility in RMS device label and permanent file processing for SCOPE 3.4, it was necessary to separate functions thus increasing the number of options in the Option Matrix. The following descriptions equate selections between SCOPE 3.3 and SCOPE 3.4.

In SCOPE 3.3:

On non-recovery deadstarts, it always was necessary to pre-load the system from tape.

RMS device labels and permanent file tables (PFD and RBTC) were initialized simultaneously.

In SCOPE 3.4:

SYSTEM=ZZZZZ04 is a permanent file; it is not always necessary to preload from tape on non-recovery deadstarts. It is possible to:

Update RMS device labels (especially the RBR flaw information) without reloading permanent files because the permanent file tables (PFD and RBTC) are not reinitialized.

Reload permanent files after initializing the permanent file tables without re-entering RBR flaw information in RMS device labels.

Prohibit permanent files from residing on mass storage other than pack devices (854 and 841).

When SCOPE 3.4 is brought up initially and the mass storage devices had been used by a system other than SCOPE 3.4, it is necessary to:

Load a fresh system (1.L) (default)

Preload from tape (2.A)

Initialize RMS device labels (3.I)

Initialize permanent file tables (4.I)

This procedure corresponds to an Initial (1.I) deadstart of SCOPE 3.3.

To guarantee the system mass storage file will reflect the current deadstart tape option, the following procedure is necessary when SCOPE 3.4 is deadstarted:

Load a fresh system (1.L) (default)

Preload from tape (2.A)

Check existing RMS device labels (3.C) (default)

Process existing permanent file tables (4.C) (default)

This procedure corresponds to a Normal (1.N) deadstart of SCOPE 3.3.

When SCOPE 3.4 is deadstarted, and this deadstart tape was the last one used with the preload from tape (2.A option), it will be necessary to:

Load a fresh system (1.L) (default)

Load from the system mass storage file (2.B) (default)

Check existing RMS device labels (3.C) (default)

Process existing permanent file tables (4.C) (default)

Basically, this procedure corresponds to a Normal (1.N) deadstart of SCOPE 3.3; but the time-consuming process of copying the deadstart tape to mass storage (preloading) is bypassed.

DEADSTARTING A DUAL PROCESSOR 6000

In SCOPE 3.3:

With corrective code transmitted for SCOPE 3.3 PSR SC30915, it is possible to deadstart on CPU-B of a dual processor system if CPU-A is inoperative. If the system is deadstarted on CPU-B, CPU-A is turned off and locked out by deadstart and MTR. This option allows the user to deadstart on CPU-B, keeping CPU-A down until it is fixed, and still carry on limited production with CPU-B.

The only change to the regular deadstart procedure is to set bit 2**5 (bit 5 relative to zero) of word 7 on the deadstart panel (SET=SWITCH UP). This applies to all initial, normal, and recovery deadstarts.

In SCOPE 3.4:

Another deadstart option appears on the Operator Option Matrix to provide for a dual processor 6000 (assembled only if IP.MCPU=2). This option provides the capability to:

1. Deadstart on CPU-A and keep CPU-B active to the rest of the Operating System.

2. Deadstart on CPU-A and turn off and lockout CPU-B to the rest of the Operating System.

3. Deadstart on CPU-B and turn off and lockout CPU-A to the rest of the Operating System.

   For this option to work, CMR must be configured for dual CPU's. If the user turns off either CPU, that CPU is locked out and cannot be turned on again until another deadstart is performed.

Deadstart Configuration Suggestions

The released values of default parameters in the Operator Option Matrix are determined by a conditional micro in the deadstart parameters common deck DSLCOM. The default MICRO OPTDF determines the initial values for the options. The MICRO appears as follows:

```
OPTDF CMICRO 1,, $LBCCNNYNN$
```

the default of the options appear from left to right in the MICRO, with L=1, B=2,---etc. To change a default value, insert a micro (named OPTDF) with the desired changes at DSLCOM.11.

The options in the MICRO as released are as follows: (left to right in the micro string):

1.L ACTION
2.B SYSTEM
3.C RMS LABELS
4.C PERMANENT FILES
5.N SYSTEM PERMANENT PACKS
6.N EQUIPMENT CHANGES
7.Y RMS pre-allocation

The following two values may or may not be defined. Their assembly is governed by the IPARAMS symbols IP.ECSB and IP.MCPU.

8.N ECS or OFF CPU
9.N OFF CPU

All symbols described below are defined in the common deck DSLCOM. Default values are shown. A 49K memory is assumed. Central memory usage by deadstart may be modified by changing the symbol values at DSLCOM.11. Most symbols are keyed from a symbol defining an adjacent area, and all depend on the value of the symbol BASE. For example, if a 131K system is to have an unusually large CM resident library, it may be necessary to set the origin address of IRCP (IRADR) to a higher value redefining BASE to any arbitrary address in the middle of CM. In either a 65K or 131K machine, ample space is available to enlarge both the CM resident area and the RBT area.

BASE	CEQU	120000B	Location from which origins of other areas are keyed.
CMRSIZE	CEQU	17000B	Number of words in CMR to be saved for recovery purposes.
PPOSVADR	CEQU	BASE	Address in CM at which contents of PP0 will be saved for dump purposes.
CHPR	CEQU	12B	Dump printer channel number.
CNTPR	CEQU	3	Dump printer controller number.
DFLTDT	CEQU	2RLP	Default dump printer mnemonic.
DFLTCHN	CEQU	2R63	Default dump 512 printer chain type.
DSPLCHAN	CEQU	10B	Display channel number.
DSPLCTLR	CEQU	7	Display controller number.
ROCKCNT	CEQU	10B	Retry count for tape parity error.

The following dependencies and constraints must be observed:

1. The central memory resident library programs must not extend past IRADR, or loading cannot complete; if they do, BASE must be redefined.

2. IRCP must not be larger than BASE - IRADR (15000B); or it will overlay DRIVBUF, destroying the driver overlays. An assembly error will occur if an attempt is made to generate an IRCP larger than the current value of BASE - IRADR.

3. The total area occupied by the RMS driver overlays must not exceed OPCXCTLW - DRIVBUF (2000B) words. The current eight drivers occupy about 1100B words.

4. When the old CMR is saved for recovery, the number of words to be moved is determined by the DSLCOM symbol CMRSIZE (17000B). This value corresponds to the start of CP.MTR in the CPR block in CMR. If CMR is larger, CMRSIZE must be redefined.

5. It is strongly recommended that the origin of BASE be at machine FL minus (maximum length of RBT area + 3000B + MAX(10200B,CMRSIZE)).

System ECS Resident Library

SCOPE 3.4 system ECS resident routines are no longer in the direct user access area (as in SCOPE 3.3); rather, they occupy a part of the paged area. The paged area is defined in terms of a page stack and accessed through a CMR central processor program (CP.CIO).

The definition and usage of ECS in SCOPE 3.4 is unlike the BNL ECS of previous versions of SCOPE. The following approach provides system ECS residency. ECS residency cannot be specified via EDITLIB creation of the deadstart tape. An EDITLIB run must be performed after deadstart to move routines from their residence as loaded by deadstart to ECS.

When EDITLIB creates a deadstart tape, it terminates the tape with a double end-of-file, effectively creating a null file following the last system library file on the tape. During the pre-loading process (after system library files have been copied to mass storage) if this last file is not null, a job of the following structure is assumed:

```
Job card.
EDITLIB(SYSTEM)
7/8/9
EDITLIB Directives
6/7/8/9
```

This job will be copied to mass storage and cataloged as a permanent file with the following parameters:

LFN = ZZZZECS

PFN = ZZZZECS

ID = SYSTEM (granted automatically for control point 0 PF operations)

TK = SYSECSLIB

XR = ECSLIB

This job will be run automatically by the Terminate Deadstart Sequence PP program (TDS) whenever ECS is up and the SYSTEM level (option 2) is either A (preload from tape) or B (load from the system permanent file).

Because of restrictions imposed for system (control point 0) permanent file operations, a user cannot catalog a new file with an ID of SYSTEM. Thus, ZZZZECS can be created only in the manner just described. Thereafter, the job can be modified by new-cycle catalog and old-cycle purges with the appropriate permissions (ID=SYSTEM allowed and required).

Permanent File (PF) Devices

To indicate which devices can hold permanent files, installations must define the PFD device and one or more permanent file devices via the EST macro. The PFD device need not be defined as a permanent file device. However, it is recommended that the PFD device be distinct from the system device. A PFD device must exist since the running system is maintained on a permanent file.

Parameters

Each installation may customize the permanent file structure by modifying the options according to the following instructions.

In the released system, permanent file parameters are defined as listed below. The installation can change them with CEQU statements inserted in the installation deck SCOPE1 when SCOPE is assembled. These installation parameters appear in COMDECK PFMIP at the line sequence number shown in parentheses.

IP.ARCH	CEQU	1	(PFMIP.7)
IP.MREWR	CEQU	0	(PFMIP.9)
IP.PFABT	CEQU	0	(PFMIP.10)
IP.PP	CEQU	0	(PFMIP.11)
IP.US	CEQU	0	(PFMIP.13)
IP.PFAC	CEQU	0	(PFMIP.15)
IP.PFRP	CEQU	10D	(PFMIP.14)
IP.UP	CEQU	10B	(PFMIP.12)
IP.RPMAX	CEQU	999	(PFMIP.16)

In addition, the following permanent file parameters appear in CMR:

N.SD	CEQU	11D
N.RBTC	CEQU	20D
N.ESD	CEQU	32D
L.APF	CEQU	30D

IP.PFAC (PF Accounting Feature) (0)

0 Accounting parameter is taken from the permanent file directive with the AC keyword.

1 Accounting parameter is taken from the control point area word, W.CPFACT. Contents of this word are supplied by the installation.

IP.PFABT (Permission Abort Override on user calls with the RC or RT bit set in FDB) (0)

0 All permanent file return codes greater than 67B will flag fatal errors.

1 All permanent file return codes will flag non-fatal errors.

IP.ARCH (Archive Feature) (1)

Permanent files dumped under a Mode 2 permanent file DUMPF:

0 No longer will have an RBTC entry and will not be retrieved from tape at ATTACH time.

1 RBTC entries will be retained and will be retrieved from tape at ATTACH time.

2 RBTC entries retained but will not be retrieved from tape at ATTACH time.

If IP.ARCH=1:

If an installation has system modifications which require accounting information before a job can come to a control point, changes must be made to the routine 1PF to insert appropriate information into the control stream of the job that performs archive file retrieval.

Any accounting information needed by the installation must be inserted into the control card buffer. This card or cards, including the job card, must be formatted according to installation procedures, using DIS or DATA statements. Each DIS or DATA that completes a card must be followed by a call to the PAD macro, which pads the card with zeros.

JOBNAME must be set equal to a valid, five-character, local file name to be used in setting up the input FNT for the archive retrieval job. 1PF will add two random digits to this jobname before storing it into the FNT. JOBNAME should be the same as that used on the job card. No CM, tape, or priority requirements need be on the job card, as 1PF sets up the input FNT with all such requirements satisfied.

Example 1 (accounting information on job card):

```
*D     PTR9312.53
CARD1    DIS    ,*JBNME.                 ACCOUNTING INFORMATION.*
         PAD
*D     PTR9312.71
JOBNAME DIS     ,*JBNME*
```

Example 2 (accounting information not on job card):

```
*D     PTR9312.53
CARD1    DIS     ,*ACCOUNTING INFORMATION*
         PAD
CARD2    DIS     ,*FGHIJ.*                   (jobcard)
         PAD

*D     PTR9312.71
JOBNAME DIS      ,*FGHIJ*
```

IP.MREWR (Multiple Rewrite Access) (0)

If this parameter is set to one, multiple modify access read access on a permanent file are allowed simultaneously.

Installations are cautioned against indiscriminate use of this feature, as the operating system cannot prevent two users from attempting to rewrite the same record at the same time.

IP.PP (Installation Privacy Procedure) (0)

If the installation uses the standard privacy procedure provided, this parameter should be zero. An installation's own privacy procedure routine must be a PP program. This parameter should be the program's name.

IP.US (User Slot Size) (0)

When the installation reserves space (slot) in the RBTC for information to be saved with each permanent file, this parameter is the space length in central memory words.

IP.PFRP (Default Retention Period) (10D)

This parameter, 0-999(decimal) is the number of days to be used as a file's retention period in the default case. The value 999 is interpreted as permanent retention.

IP.RPMAX (Maximum Retention Period) (999D)

This parameter defines the maximum retention period for permanent files. Only 999D allows an infinite retention period. All other values define the maximum retention disallowing infinite retention.

IP.UP (Universal Permission) (10B) (17B)

The value assigned IP.UP will determine permissions granted when the Universal permission password is submitted. (see section on Universal Permission) Any combination of permissions can be granted, depending on the bit settings as shown below:

Control (8)

Modify (4)

Extend (2)

Read (1)

L.APF (APF table length in CM words) (30D)

This value, an integral multiple of LE.APF, defines the APF table length which is low-core resident.

LE.APF (Length of an APF entry) (2) (Defined in COMDECK PPSYS)

This parameter defines the length of an APF entry.

N.SD (Number of Sub-directories) (11D) (17D)

The value selected should be a prime number.

N.RBTC (Length of RBTC) (20D)

Length of the RBT catalog in PRU's divided by 16(decimal).

N.ESD (Sub-directory Length) (32D)

Decimal number of PFD entries per sub-directory (4 entries per PRU). If N.ESD is not a multiple of 32D, it will be rounded up to the nearest multiple of 32.

A cross reference listing of the PFMIP parameters follows:

```
*****
PFMIP
*****
IF.ARCH       1PF   PFA   DPF
IP.MREWR      PFC   1PF   PFA
IF.PFABT      PFC   1FC   PFA   LPF   PFF   PFE   PFR   1PD   PFS   TPT
IP.PFAC       1FC   PFR
IF.PFRF       1PC   PFC   1PF   1FC   PFA   LPF   PFP   PFE   PFR   1PD
              PFS   DPF   TPT
IP.PF         PFC   1FC   PFA   LPF   PFR
IF.RFMAX      1PC   PFC   1PF   1FC   PFA   LPF   PFP   PFE   PFR   1PD
              PFS   DPF   TPT
IF.UF         PFA
IF.US         1FC
*****
```

Procedures

Size of Directory

Before a permanent file manager can be customized for an installation, the size of the permanent file directory must be determined by estimating the maximum number of unique file names to be allowed in the system. (The actual number of files can be five times this number.) The maximum number of unique file names is determined by the number of entries per subdirectory (N.ESD) times the number of subdirectories (N.SD). After estimating the number of file names and allowing a sufficient margin for expansion (at least 25 percent), the number of subdirectories must be determined (N.SD). In the PFD, files are found by hashing the ID to a subdirectory. Overflow is to the next highest subdirectory. Because of the hashing scheme, the more subdirectories the better. However, an FNT (3 CM words in low core) is required for each subdirectory. The number of subdirectories should be dictated by the number of FNTs the installation wants to allocate. The ID hashing algorithm optionally will distribute the files to the subdirectories if the number of subroutines is a prime number.

Size of Catalog

Each permanent file cycle has an RBT catalog entry of variable length; the minimum is 15 CM words. The average length of an RBT catalog entry for a given system is dependent on the average size of a permanent file. If the size of the RBT chain for the average file was calculated, the size of the average RBT catalog entry would be 15 CM words (minimum length of RBT catalog entry) plus the size of the RBT chain for the average file (CM words) plus the value chosen for IP.US. (These averages are relative to a given installation). The length of the average RBT catalog entry then could be multiplied by the total number of permanent file cycles allowed by the installation, as determined by the size of the PFD. This product would give the suggested size in CM words of the RBT catalog. This number then could be converted to the number of PRUs needed. N.RBTC should be equated to the number of PRUs needed, divided by 16.

RBTC space is reused. The size of the RBTC, if calculated by this algorithm, allows for five cycles for every permanent file. Normally, this many cycles will not be needed. Therefore, the installation may want to decrease the initial estimate.

Size of APF Table

The APF table contains two-word entries and is central memory resident. Every attached permanent file active in the system as opposed to files merely cataloged, must have an APF entry. Number of APF entries (L.APF) limits the number of permanent files attached simultaneously by all jobs in the system.

Other Parameters

Other installation parameters that must be assigned values are described below.

The default retention period (IP.PFRP) for permanent files is decided by the installation, based partly on the amount of available mass storage.

If an installation writes its own privacy procedure routine, IP.PP should be equated to its name. In this case also, if space is required for installation information, it may be reserved within each RBTC entry by equating IP.US to the number of CM words. The only restriction on this option is that no word should start with two bytes of 7's. This option affects the minimum size of the RBTC entry and may affect the size of N.RBTC.

Universal Permission

By setting the parameter IP.UP non-zero, a universal permission code mask may be defined. This mask is the combination of permissions to be granted when the universal password is correctly submitted; it consists of two octal digits as follows:

01	Read permission	04	Modify permission
02	Extend permission	10	Control permission

Thus, if IP.UP were defined as 13(octal), Read, Extend and Control permissions (but not Modify permission) would be granted when the universal password was submitted.

The universal password (nine characters) is defined by the installation and assembled into the PF routines as in the following sample:

```
*DELETE    PREAMB.152

UNIV       DIS         ,*UNIVRPERM*
```

As released, IP.UP is set at 10(octal), granting only control permission. This is externally identical to the 3.2 and 3.3 release; and as only control permission is granted, privacy is not threatened.

Public ID Permission

If a permanent file has the ID of PUBLIC, the ID parameter need not be specified on all permanent file directives. In other words, ID of PUBLIC is the default ID. Since the ID is used to locate files in the PFD, it is not desirable that many files have the same ID. Therefore, an ID permission password must be supplied at catalog time to allow a file to be cataloged with the ID of PUBLIC.

The PUBLIC ID password (nine characters) is defined by the installation and assembled into the PF routines (identified by PREAMB.153).

```
IDPERM     DIS         ,*PBLICPERM*
```

Privacy Procedures

Specialized needs of any installation may be met by including their own privacy procedures. Each privacy procedure written for a specific installation must be a PP routine executable in two modes: CATALOG and ATTACH.

Privacy procedures written for SCOPE 3.2 or SCOPE 3.3 will not run under SCOPE 3.4.

Installation parameter IP.PP must be defined as non-zero for individual privacy procedures. The parameter, consisting of the three characters of the PP routine name, must be unique within the system. For a CATALOG or ATTACH function, the permanent file manager checks IP.PP; if non-zero, the permanent file manager calls that routine and does not execute its privacy routine.

Method of Call

When the CATALOG or ATTACH function calls an installation privacy procedure, the address of the FDB+4 is put into the function input register. The PF PP routine will call the privacy procedure into its PP. The privacy procedure will save and restore the last three words of the message buffer and then call the PF PP routine. The PP routines should call each other in the same way by: setting bit 41 to 1 in the input register, putting the name of the PP routine to be called in the input register, and jumping to R.IDLE.

A new cycle catalog will be flagged by non-zero bits 24-35 of the input register.

Time of Call

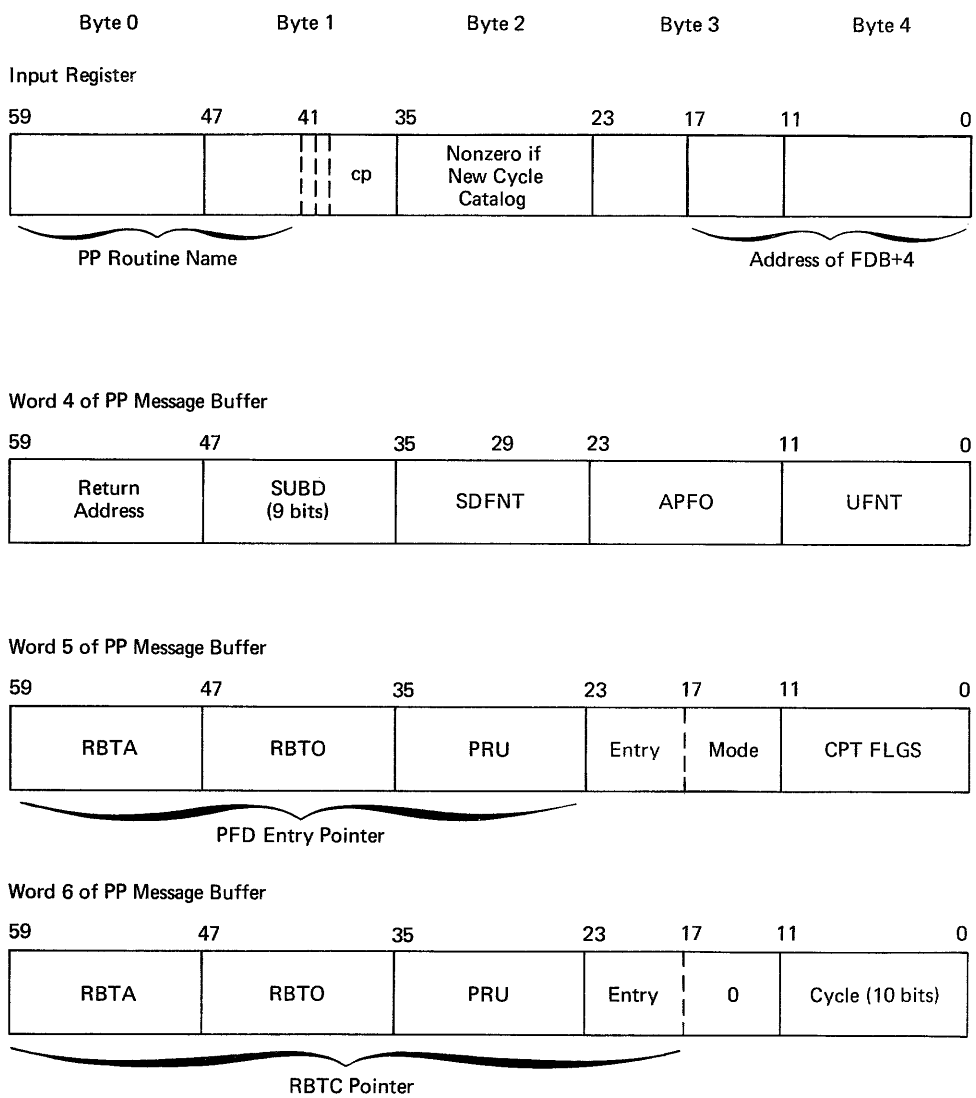
The CATALOG mode call is made at completion of the CATALOG function, prior to unlocking the APF entry. At this time, passwords may be saved, any additional parameters can be extracted from the FDB, and other manipulation carried out. Care must be taken to avoid information changes that will cause a PFM abort.

The ATTACH and newcycle CATALOG mode calls are made instead of calling the system password checking routines. The privacy procedure must include its own password checking and store resultant permissions in word 6 of the PP message buffer.

The PP parameter is available to the installation privacy routine; it can be used to pass up to nine display-coded characters from the user to the routine via the FDB.

Tables

Shown below are the contents of the input register and the last 3 words of the PP message buffer when another permanent file PP routine of privacy procedure is to be called:



Assembly Options

Delay Limit

At times during execution, permanent file manager routines may be delayed because an interlock is not available. If a routine delays longer than 1000 milliseconds, the event stack is used; a lesser delay is implemented within the PP.

Delay Times

Two specific delay times, associated with different interlocks, are specified in the routines; they can be varied for optimum performance. The system is released with the decimal values shown below.

APFDLY	EQU	1	Wait, APF table	(PREAMB.108)
RWDELAY	EQU	100	Stack processor interface delay	(PREAMB.109)

The CM table delays are set to one millisecond each.

Converting Files from SCOPE 3.2 or 3.3 to SCOPE 3.4 System

Because table formats differ for permanent files depending on the version of SCOPE, files cataloged under SCOPE 3.2 or 3.3 cannot be accessed immediately by SCOPE 3.4.

To facilitate file conversion, the LOADPF utility in 3.4 accepts dump tapes created by SCOPE 3.2, or 3.3 permanent file manager. When LOADPF detects input, automatic conversion takes place if the LP parameter is used on the LOADPF control card. For files created and dumped under 3.2 or 3.3 systems, backup tapes should be reloaded onto the system after Initial deadstart of SCOPE 3.4.

Automatic Accounting Feature

This feature is activated by setting IP.PFAC=1, when the system is assembled. It is included for accurate accounting of charges for system resource use.

The installation must provide a PP routine (part of its own accounting routines), to store the user's account number into each control point area, in word W.CPFACT. This account number is presumed to have been taken from the job card or elsewhere. The identification, 1 to 9 alphanumeric display-coded characters, has the following format.

59 5	0
Account Number (Right Justified Display Code)	16

The account number should be right justified to the 16 code (octal) and left filled with binary zeros.

SCHEDULING PARAMETERS

Definitions:

1. Minimum queue priority (MINQP) The priority with which a job will first enter the CM queue.

2. Maximum queue priority (MAXQP) The maximum priority level a job in the CM queue may achieve while waiting for scheduling.

3. Base quantum (BQ) The amount of time that a job, once brought to a control point, will maintain a high enough priority to avoid being swapped out by another job.

4. Quantum priority (QP) The priority given to a job when it has been swapped-in. The job maintains that priority for the duration of its base quantum.

5. Age rate (AR) A factor used to weight the priority of a job according to the time it has spent in the CM queue.

The above parameters apply to each of the available classes of jobs. Each class serves to define a series of jobs by their common characteristics, such as response time requirements or the minimum amount of time that a job has access to core.

The five classes are:

BATCH

DEVICE (BATCH with non-allocatable devices)

INTERCOM

MULTI-USER

EXPRESS

When a job requests scheduling for central memory, its Job Descriptor Table entry is placed into the central memory queue with a queue priority equal to the minimum queue priority of its class. Its priority is evaluated according to: its minimum queue priority, the age rate of the class, time in the queue, and the job card priority. When the priority of the job reaches the maximum for the class, aging ceases. This priority evaluation is performed for all jobs in the central memory queue, and the results are compared with the priorities of those jobs at control points. When a job is swapped into central memory, it is given a priority equal to the quantum priority of its class.

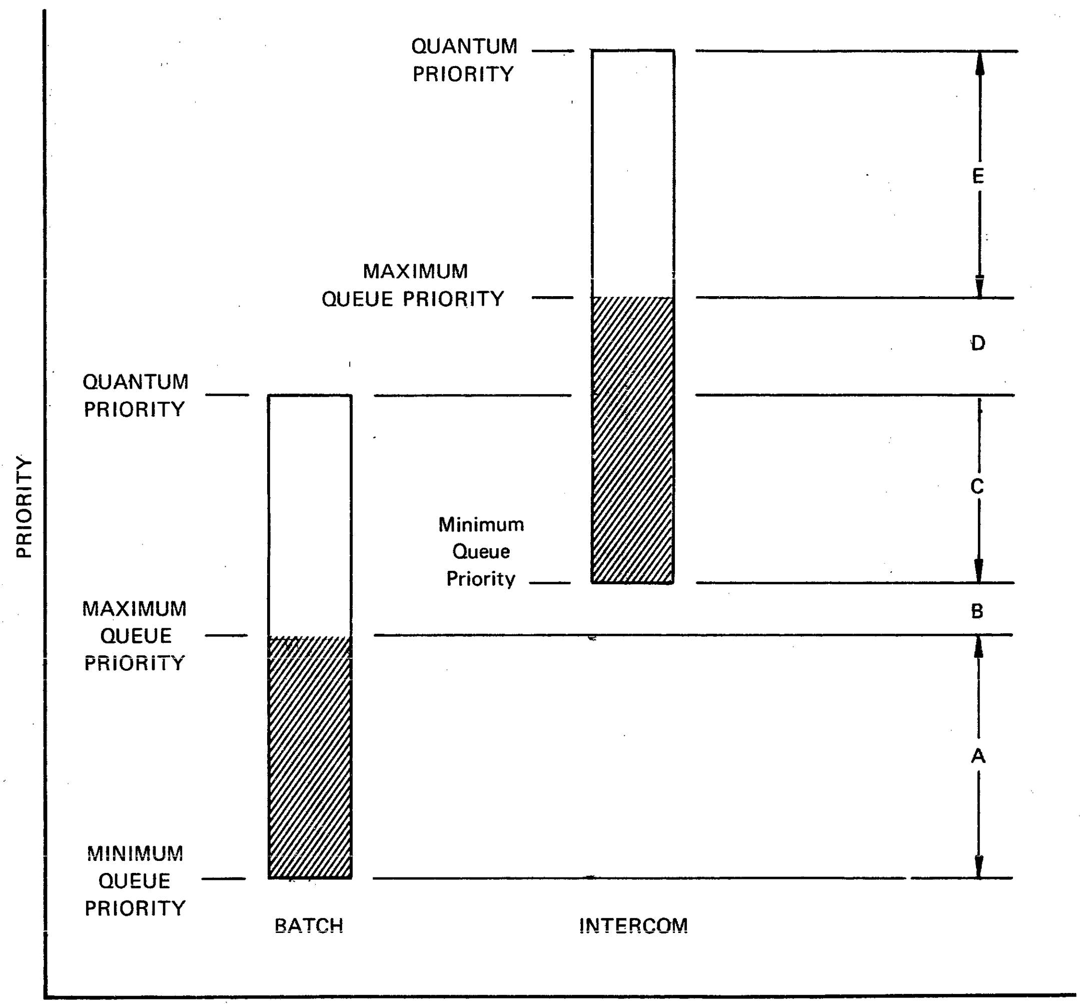
When the quantum of the job has elapsed, its priority is reduced to the minimum of its class.

Since it requires some overhead to swap a job, the quantum permits a job to remain at a control point for a reasonable length of time before it becomes eligible for swapping. The quantum of a job is considered elapsed when the job has used a specified amount of CPU and/or PPU time.

All priorities for a class, excepting MAXQP, are weighted by job card priority.

The following figure illustrates the interaction between two classes, batch and INTERCOM and displays in a graphical form, the relationship between the parameters of these two classes.

BATCH/INTERCOM CONSIDERATIONS



The assumptions used in formulating this set of parameters were that the response time for INTERCOM users should fall within certain bounds, irrespective of the batch loading; that, once a batch job is executing, it has a guaranteed period of execution before competing with other batch jobs; and that a batch job be allowed to execute a minimum period before a swap out can be forced by an INTERCOM job. Within the batch class, the aging between the minimum queue priority and maximum queue priority (interval A) is intended to ensure that, job card priority considerations aside, the first batch job to enter the central memory queue will be the first job to be swapped into central memory. The minimum queue priority for an INTERCOM job is greater than the maximum queue priority of the batch job (interval B) so that INTERCOM jobs will not have to compete with batch jobs waiting for central memory. Aging of jobs in the INTERCOM class serves two purposes: Firstly, as in the batch class, to ensure first into the central memory queue - first into central memory. Secondly, to allow INTERCOM jobs, after a certain period of time has elapsed, to force the swap-out of a batch job in order that the INTERCOM job may run.

The extra increment D, between the quantum priority of a batch job and the maximum queue priority of an INTERCOM job, allows INTERCOM jobs to be selective in the batch jobs that are swapped-out to provide core, by becoming eligible to swap, first of all, low job card priority batch jobs and, eventually, to be able to force out even the highest job card priority batch jobs. The interval D can be set smaller than the total range of the job card priority values. By doing so, those jobs with a very high job card priority will not be forced out by INTERCOM jobs before their quantum has expired. The interval E between the maximum queue priority and the quantum priority of INTERCOM jobs is used for similar purposes as interval C in the relationship between INTERCOM jobs and the next higher class of users. Similarly, this also will allow INTERCOM jobs to run to their quantum before they start to compete for central memory with other INTERCOM jobs.

The following list is the default set of parameters, as they appear in CMR. The parameters selected provide good through-put for an installation running a heavy load of batch jobs, as well as provide good response time for a 20-terminal INTERCOM system - where an average of ten terminals are active at any one time.

```
*                                                                      CMR     1016
*          SCHEDULER PARAMETER SETTINGS                                CMR     1017
*                                                                      CMR     1018
*                                 PARAMETER            S DISPLAY       CMR     1019
*                                 DESCRIPTION            TYPEIN        CMR     1020
MAXNBA     CEQU    30B       MAX NO OF JOBS W/O        MAXN1           CMR     1021
*                            NON-ALLOC EQUIP                           CMR     1022
MAXNDE     CEQU    10B       MAX NO OF JOBS WITH       MAXN2           CMR     1023
*                            NON-ALLOC EQUIP                           CMR     1024
QPINP      CEQU    2200B     INPUT QUEUE QUANTUM       QP0             CMR     1025
*                            PRIORITY                                  CMR     1026
BQINP      CEQU    2000B     INPUT QUEUE BASE          BQ0             CMR     1027
*                            QUANTUM                                   CMR     1028
*                                                                      CMR     1029
*          BATCH CLASS                                                 CMR     1030
*                                                                      CMR     1031
MINQPBA    CEQU    100B      MIN QUEUE PRIORITY        MINQP1          CMR     1032
MAXQPBA    CEQU    1000B     MAX QUEUE PRIORITY        MAXQP1          CMR     1033
ARBA       CEQU    4B        AGEING RATE               AR1             CMR     1034
QPBA       CEQU    1400B     QUANTUM PRIORITY          QP1             CMR     1035
BQBA       CEQU    2000B     BASE QUANTUM              BQ1             CMR     1036
*                                                                      CMR     1037
*          DEVICE CLASS                                                CMR     1038
*                                                                      CMR     1039
MINQPDE    CEQU    200B      MIN QUEUE PRIORITY        MINQP2          CMR     1040
MAXQPDE    CEQU    1000B     MAX QUEUE PRIORITY        MAXQP2          CMR     1041
ARDE       CEQU    10B       AGEING RATE               AR2             CMR     1042
QPDE       CEQU    1400B     QUANTUM PRIORITY          QP2             CMR     1043
BQDE       CEQU    2000B     BASE QUANTUM              BQ2             CMR     1044
*                                                                      CMR     1045
*          INTERCOM CLASS                                              CMR     1046
*                                                                      CMR     1047
MINQPIN    CEQU    1110B     MIN QUEUE PRIORITY        MINQP3          CMR     1048
MAXQPIN    CEQU    2400B     MAX QUEUE PRIORITY        MAXQP3          CMR     1049
ARIN       CEQU    1000B     AGEING RATE               AR3             CMR     1050
QPIN       CEQU    2500B     QUANTUM PRIORITY          QP3             CMR     1051
BQIN       CEQU    200B      BASE QUANTUM              BQ3             CMR     1052
*                                                                      CMR     1053
*          MULTI-USER CLASS                                            CMR     1054
*                                                                      CMR     1055
MINQPMUJ   CEQU    2410B     MIN QUEUE PRIORITY        MINQP4          CMR     1056
MAXQPMUJ   CEQU    2510B     MAX QUEUE PRIORITY        MAXQP4          CMR     1057
ARMUJ      CEQU    200B      AGEING RATE               AR4             CMR     1058
QPMUJ      CEQU    3000B     QUANTUM PRIORITY          QP4             CMR     1059
BQMUJ      CEQU    4000B     BASE QUANTUM              BQ4             CMR     1060
*                                                                      CMR     1061
*          EXPRESS CLASS                                               CMR     1062
*                                                                      CMR     1063
MINQPEXP   CEQU    1000B     MIN QUEUE PRIORITY        MINQP5          CMR     1064
MAXQPEXP   CEQU    3200B     MAX QUEUE PRIORITY        MAXQP5          CMR     1065
AREXP      CEQU    400B      AGEING RATE               AR5             CMR     1066
QPEXP      CEQU    3200B     QUANTUM PRIORITY          QP5             CMR     1067
BQEXP      CEQU    400B      BASE QUANTUM              BQ5             CMR     1068
****                                                                   CMR     1069
```

The two batch classes have low minimum and maximum queue priorities as well as low age rates. The device class has twice the age rate as the batch class, providing the device class with a scheduling advantage over the batch class. A device class job would experience, on the average, half the wait time of a batch class job. Since the device class represents additional resources being tied up, such as control points and tapes, it is preferable to get that job through the system with a minimal delay.

The quantum priorities of the batch and device classes are low enough so that INTERCOM jobs, having a high age rate, can force batch jobs to be swapped out after a one-half to two second delay, depending on job card priority and quantum considerations.

The INTERCOM class job is given a small base quantum which normally will be enough time to execute an INTERCOM job step. The batch quantum on the other hand, is larger, preventing batch jobs from swapping other batch jobs unnecessarily. The multi-user class job, such as EDITOR, is given the highest priority because it can service several INTERCOM users simultaneously.

The parameter MAXN determines the maximum number of batch or device class jobs which can run at any given time. The number of device class jobs is kept small; the determining factor being that device class jobs are rolled-out rather than swapped out; each job can make a control point unavailable for swapping. It is essential to keep a reasonable number of control points available for serving other jobs. The maximum number of batch jobs is much higher, a large number being preferable to provide the scheduler with a better pool of job candidates, allowing better core utilization. However, too large a job pool may adversely affect individual job turn-around while improving total system through-put.

The two parameters QPO and BQO in the lower half of the S display are the Quantum Priority and Base Quantum given to jobs coming out of the input queue and entering a control point for the first time. The quantum priority is higher than that for normal batch jobs, enabling short jobs to run to completion without swapping.

The express queue is given a high priority and aging rate, since it contains all jobs terminated by operator intervention. The quantum is small because the end-of-job procedure is minimal. This class was given express consideration under the assumption that these jobs would release valuable resources back to the system.

SELECTION OF ECS INSTALLATION PARAMETERS

The ECS extensions are designed primarily to improve the efficiency of an I/O bound system by:

1. Buffering the sequentially accessed RMS files through ECS
2. Swapping jobs to ECS
3. Moving a part of the system library to ECS
4. Allocating files in ECS

The default values of the ECSCOM configuration parameters are defined with the CEQU or CMICRO macros, so that an installation can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variable have not been previously defined, any modifications should precede them.

Installation parameters oriented to ECS are defined in the COMDECK ECSCOM. Changes may be made at ECSCOM.8. Default values, other tested values, and parameter descriptions are as follows:

IP.EBUF (16D)

Defines the default ECS buffer size in pages. To significantly improve system I/O, the ECS buffer allocated to a file should be at least four times larger than the buffer used in CM for the same file, resulting in a default value in the 10000 to 20000 (octal) words range. A larger ECS buffer (40K or more) does not provide any significant improvement compared with the default value.

If an ECS buffered file does not overflow its buffer, it stays in ECS and is processed as an ECS resident file, possibly locking a very large amount of ECS for only one file. Buffer space is not reserved when the buffer is requested; it is allocated only when needed and released as soon as possible, one page at a time. Allocation of an ECS buffer to a file having a CM buffer approximating one RB does not improve throughput because of the scheduling algorithum used by the stack processor.

IP.BDCT (1) (3)

Number (1-3) of CM system buffers. Throughput may be improved by having a ratio of one system buffer per RMS device controller (maximum of three can be handled).

IP.ELIB (0) (60)

If zero, the code for ECS resident library will not be assembled in the system. If non-zero, maximum number of words/1000 (octal) that may be used for storing ECS resident library programs. This value can be changed at deadstart time; however, it can be non-zero only if IP.ECSB is non-zero.

IP.ERES (0) (1)

If set to one the ECS Resident File capability is activated. The ECS Resident File Option feature can improve system throughput for a given job by keeping large files (particularly random access) in ECS. However it can have an adverse effect on the overall improvement of the system by drastically reducing the amount of ECS available for job swapping, ECS buffering, and the system library.

IP.EDAA (40B)

Number of words/1000(octal) reserved for ECS direct access. This value can be changed at deadstart time.

IP.EPAG (8D)

ECS page size in number of PRUs: 8*64=512 words. This value can be changed at deadstart time. Allocatable ECS is divided into pages. The size of a page is at least 64 words and, normally, a multiple of 64. A small page size (64 to 256 words) will increase the system overhead for space management and a large size (2048 words and up) will increase the amount of wasted space. An initial setting of 512 or 1024 words page is a good compromise.

IP.SBLG (8D)

CM system buffer length in number of PRUs (minimum 4): 2*(8*64+8)=1040 words. The CM system buffers are used by the stack processor to perform the RMS/ECS transfers.

CM System Buffer

A CM system buffer is used by each stack processor for the RMS/ECS transfers. Each independent RMS device should have one such buffer. Because of the CM conflicts, no more than three RMS devices can be driven simultaneously for ECS I/O buffering, therefore, the number of CM system buffers (IP.BDCT) is limited to 1, 2 or 3.

The minimum size (IP.SBLG) of a CM system buffer is 4 PRU's (256 words). A size between 6 and 8 PRU's should be used to minimize the possibility of lost revolutions on RMS devices because of CM access conflicts that can slow down or lock out the stack processor.

A cross reference of the ECS installation parameters is as follows:

```
*****
ECSCOM
*****
IP.BDCT           CMR

IP.EBUF           IRCP

IP.EDAA           IRCP
IP.ELIB           STL        IRCP       CMR        1AJ        TDS
IP.EPAG           IRCP
IP.ERES           REQ
IP.SBLG           CMR
```

Installation of SCOPE 3.4 cannot be accomplished using SCOPE 3.2 or SCOPE 3.3 as the running system. The unconfigured deadstart tape provided with the release must be used.

The SCOPE 3.4 unconfigured deadstart tape requires the entry of EST entries for RMS, tape, and unit record equipment at deadstart time. A new assembly of CMR is necessary, with a configured program library and installation tape constructed as soon as possible, so that normal loading becomes feasible. The user is cautioned to review the NOTES AND CAUTIONS, in the early part of this section prior to building a configured system. The SCOPE 3.4 program library requires a full 2400-foot tape reel when it is recorded at 800BPI.

The four jobs listed on the following pages, plus a small verification program, are included on the release PL1; they may be obtained by executing a job such as that listed below.

Job SCOPE1 updates PL1 and places the assembled binary of the revised PL1 on tape BIN. Job SCOPE1 is set up assuming that ECS is present. If the ECS code is not activated, cards must be removed from the SCOPE1 Update record. To use the SCOPE1 deck for program library maintenance, remove the cards sequenced 46 and 47 after initial installation. The updated system texts are added to the running system prior to assembling SCOPE. This effort, not required but strongly encouraged for initial installation, is recommended when subsequent system maintenance is to be performed.

To install SCOPE 3.4 for use as a station front end, a card of the form *DEFINE,STATION must be added to the update record of deck SCOPE1. This card should be removed from the deck once a configured program library is created.

Job SCOPE2 will create a configured deadstart tape containing SCOPE 3.4 from tape BIN and the balance of the running system.

Job SCOPE3 will create a deadstart tape of the running system.

Job SCOPE4 will install system text SYSTEXT in a form equivalent to CPCTEXT. As present on the unconfigured deadstart tape and installed by 6RM installation decks SIXRM1 and SIXRM2, text SYSTEXT will be equivalent to IOTEXT.

FORTRAN Extended 4.0 must be added to the running system prior to assembling SCOPE.

To obtain the installation decks, perform a job of the type:

```
Job card
REQUEST,PL1,E,HY.             Mount released PL1
REWIND(PL1)
SKIPF(PL1,1,17)
COPYBF(PL1,PUNCH,4)           Punch installation decks
COPYBF(PL1,PUNCH,1)           Punch verification program
UNLOAD(PL1)
6/7/8/9
```

If INTERCOM 4.1 is to be installed, the required IPARAMS and CMR modifications must be included in the SCOPE1 installation deck: the INTERCOM installation deck does not include provision for access to the SCOPE program library.

The unconfigured deadstart tape contains 3DO, 1S5, 1SX, 1SP and its 3Sx system device overlays as CM resident. If 1EP is DEFINEd, the MOVE directive COMMENT cards regarding 1EP and its system device overlays must be invoked within job SCOPE2. Routines established as CM resident in the running system will have CM residency on the new deadstart tape created by jobs SCOPE2 and/or SCOPE3.

CAUTION: ECS residency cannot be carried on a deadstart tape: routines moved to ECS resident will be established as disk resident on a new deadstart tape. The only way to set ECS resident is via MOVE directive changes to the running system.

If one additional CMR is to be added, these cards must be inserted behind the indicated sequence number of jobs SCOPE1 and SCOPE2.

UPDATE(Q,P=PL1,C=COMP) COMPASS(I=COMP,S=PPTEXT,L=0)	(After card 018 in job SCOPE1)
*/ PLACE CMR CONFIGURATION HERE. *COMPILE CMR 7/8/9	(After card 055 in job SCOPE1) Only installation parameters in the full update CMR will be assembled into the system.
COPYBR(BIN,CMR1)	(After card 008 is job SCOPE2)
REWIND(CMR1) TRANSFER(CMR,CMR1)	(After card 023 in job SCOPE2) CMR from Q update

If more than one additional CMR is added, the CMR parameter on the COPYBR, REWIND, and TRANSFER cards must be changed to reflect the additional CMR number, for example, CMR2, CMR3, etc. If multiple CMRS are to be added, multiple copies of the above card sequences are required.

NOTE: The T7000 parameter on the SCOPE1 jobcard must be changed to T20000 if the deck is to be run on a 6200; T14000 on a 6400.

```
SCOPE1,CM65000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF SCOPE
COMMENT. THE TEXTS WILL BE EDITLIBED INTO THE RUNNING SYSTEM AFTER
COMMENT. THE FULL UPDATE
COMMENT. THE NEW PL1 WILL BE THE OLDPL OF SCOPE
COMMENT.  THE NEW BIN TAPE WILL CONSIST OF TWO FILES
COMMENT. FILE CNE WILL CONTAIN BINARY OF SCOPE PP ROUTINES
COMMENT. FILE TWO WILL CONTAIN BINARY OF SCOPE CP ROUTINES
LAEEL(PL1IN,R,L=SCOPE3P4,D=HY)
REGUEST,PL1,N,HY.     SCRATCH FOR NEW PL1
REWINC(PL1IN,PL1)
LABEL(PL1,W,L=SCOPE3P4,D=HY)
REWIND(PL1)
UPDATE(F,P=FL1IN,N=PL1)
COMMENT.   *** THE FOLLOWING CARD COMPILES LDRTEXT THRU IPTEXT ***
COMPASS(I=CCMPILE,S=PPTEXT,B=SAVTXT,L=0)
REWIND(SAVTXT)
EDITLIB(SYSTEM).
COMMENT.   *** THE FOLLOWING CARD COMPILES CEA THRU DSD ***
COMPASS(I=CCMFILE,S=PPTEXT,L=0)
UNLOAD(PL1,FL1IN)
COMMENT.     *** THE FOLLOWING CARD COMPILES CIO THRU EPF
COMPASS(I=CCMFILE,S=SCPTEXT,L=0)
REWIND(LGO)
REGUEST,EIN,HI.    SCRATCH FOR BINARY
REWIND(BIN)
COFYEF LGO,EIN.
REWIND LGO.
COMMENT.     *** THE FOLLOWING CARD COMPILES PFCCP THRU SYSEQ
COMPASS(I=CCMFILE,S=SCPTEXT,L=0)
COMMENT.     *** THE FOLLOWING CARD COMPILES AUDIT, DMPECS AND PFDUMP
FTN(I=COMPILE,S=CPCTEXT,B=OVLS,L=0)
LOAD(CVLS)         GENERATION OF OVERLAYS
NOGO.
```

```
                                                                        035
REWIND(SCF)                                                             036
COPYBF(SCF,LGO)                                                         037
BKSP(LGO,1)                                                             038
REWIND(SAVTXT)                                                          039
COPYBF(SAVTXT,LGO)                                                      040
REWIND LGO.                                                             041
COPYEF(LGO,BIN)                                                         042
UNLOAD(BIN)
7/8/9  END OF RECORD                                                    044
*/ ADD CORRECTIONS HERE                                                 045
*/ REMOVE CARDS SEQUENCED 46 AND 47 AFTER INITIAL INSTALLATION          046
*ID 1EFON                                                               047
                                      NOTE
*DEFINE,1EP
7/8/9  END OF RECORD                                                    049
READY(SYSTEM,OLD).                                                      050
LIBRARY(NUCLEUS,OLD).                                                   051
REPLACE(LDRTEXT+IPTEXT,SAVTXT)                                          052
FINISH.                                                                 053
COMPLETE.                                                               054
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE

                                                                        001
SCOPE2,CM55000,T1000,MT02.                                              002
COMMENT. THIS JOB CREATES A NEW DEADSTART TAPE FROM TAPE  BIN  CREATED BY 003
COMMENT. JOB SCOPE1 AND THE BALANCE OF THE RUNNING SYSTEM (PRODUCT SET  004
COMMENT. MEMBERS).                                                      005
COMMENT. THIS JOB IS SET UP FOR ONE CMR.                                006
REQUEST,BIN,HI.    MOUNT BIN TAPE CREATED BY DECK SCOPE1.               007
REQUEST,NEWSYS,HY.          MOUNT SCRATCH TAPE FOR NEW DEADSTART        008
REWIND,BIN,NEWSYS.                                                      009
COPYBF(BIN,PPLIB)              SCOPE PP ROUTINES                        010
COPYBF(BIN,NUC)               SCOPE CP ROUTINES                         011
REWIND,BIN,PPLIB,NUC.                                                   012
EDITLIB(SYSTEM)                                                         013
UNLOAD(BIN,NEWSYS)                                                      014
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD                                                    016
READY(NEWSYS)                                                           017
REWIND(SYSTEM)                                                          018
REWIND(PPLIB)                                                           019
TRANSFER(CEA,PPLIB)                                                     020
SKIPF(1,SYSTEM)                                                         021
*/  ADD  TRANSFER(12,SYSTEM)   IF CEDIAGNOSTICS IN RUNNING SYSTEM       022
TRANSFER(CEC+DMT,PPLIB)                                                 023
SKIPF(14,PPLIB)                                                         024
TRANSFER(CMR,PPLIB)      FROM FULL UPDATE                               025
SKIPB(15,PPLIB)                                                         026
TRANSFER(CCM+IRCP,PPLIB)                                                027
SKIPF(1,PPLIB)                                                          028
TRANSFER(*,PPLIB)                                                       029
INCLUDEEP(SYSTEM)                                                       030
INCLUDE(NUCLEUS,SYSTEM,CM)                                              031
INCLUDE(SYSOVL,SYSTEM,DS)                                               032
INCLUDE(FORTRAN,SYSTEM,DS)                                              033
INCLUDE(COBOL,SYSTEM,DS)                                                034
INCLUDE(RUN2P3,SYSTEM,DS)
```

```
INCLUDE(SYSIO,SYSTEM,DS)                                          035
INCLUDE(SYSMISC,SYSTEM,DS)                                        036
INCLUDE(IGS274,SYSTEM,DS)                                         037
REPLACE(*,PFLIB)                                                  038
*/    IF ECS TURNED ON ADD CARDS ←— NOTE                          039
*/    MOVE(1EP,CM)                                                040
*/    MOVE(3EP+3ES,CM)                                            041
SETAL(DIS,1)                                                      042
SETAL(DMP,1)                                                      043
SETAL(VSN,1)                                                      044
SETAL(LOC,1)
LIBRARY(NUCLEUS,OLD)                                              045
REPLACE(PFCCF,NUC,AL=7777,FL=2000,FLO=0)                          046
REPLACE(LOADPF,NUC,AL=7777,FL=31000,FLO=0)                        047
REPLACE(TRANSPF,NUC,AL=7777,FL=10000,FLO=0)                       048
REPLACE(DUMPF,NUC,AL=7777,FL=20000,FLO=0)                         049
REPLACE(REQUEST,NUC,AL=3,FL=2000,FLO=0)                           050
REPLACE(RESTART,NUC,AL=7777,FL=25000,FLO=0)                       051
REPLACE(COPYXS,NUC,AL=7777,FL=20000,FLO=0)                        052
REPLACE(SEGBILD+SEGRES,NUC,AL=7777,FL=60000,FLO=1)                053
REPLACE(LOAD,NUC,AL=7777,FL=30000,FLO=1)                          054
REPLACE(LOADC+UCLOAD,NUC,AL=0)                                    055
REPLACE(LIBRARY,NUC,AL=7777,FL=300,FLO=0)                         056
REPLACE(LOAD0+LOAD03,NUC,AL=0)                                    057
REPLACE(CCMBINE,NUC,AL=7777,FL=20000,FLO=0)                       058
REPLACE(XXXRESQ+XXXDMPQ,NUC,AL=7777,FL=10000,FLO=0)               059
REPLACE(EDITLIB,NUC,AL=3,FL=45000,FLO=1)                          060
REPLACE(EDITSYS+EDITUSR,NUC,AL=0)                                 061
REPLACE(COPYBCD,NUC,AL=3,FL=20000,FLO=0)                          062
REPLACE(LABEL,NUC,AL=7777,FL=2000,FLO=0)                          063
REPLACE(COPYL,NUC,AL=3,FL=20000,FLO=0)                            064
REPLACE(LISTMF,NUC,AL=7777,FL=1000,FLO=0)                         065
REPLACE(CPC+IC,NUC,AL=0)                                          066
REPLACE(CCMPARE,NUC,AL=3,FL=7000,FLO=0)                           067
REPLACE(CHEKPT,NUC,AL=7777,FL=300,FLO=0)                          068
REPLACE(TRAP,NUC,AL=7777,FL=20000,FLO=1)                          069
REPLACE(TRAPPER,NUC,AL=0)                                         070
REPLACE(SETCORE,NUC,AL=7777,FL=30000,FLO=1)                       071
REPLACE(BKSP,NUC,AL=3,FL=1000,FLO=0)                              072
REPLACE(COPY+COPYBR,NUC,AL=3,FL=20000,FLO=0)                      073
REPLACE(COPYN,NUC,AL=3,FL=6000,FLO=0)                             074
REPLACE(COPYSBF,NUC,AL=3,FL=20000,FLO=0)                          075
REPLACE(REWIND,NUC,AL=3,FL=2000,FLO=0)                            076
REPLACE(UNLOAD+RETURN,NUC,AL=1,FL=2000,FLO=0)                     077
REPLACE(UPDATE,NUC,AL=3,FL=40000,FLO=0)                           078
REPLACE(SKIPB+SKIPF,NUC,AL=3,FL=1000,FLO=0)                       079
REPLACE(TRANSR+TRANSF,NUC,AL=7777,FL=200,FLO=0)                   080
REPLACE(DISPOSE,NUC,AL=3,FL=0,FLO=0)                              081
REPLACE(RECOVR,NUC,AL=0)                                          082
REPLACE($SYS.RM$,NUC,AL=0)                                        083
REPLACE(AUDIT,NUC,AL=3,FL=3500 0,FLO=0)                           084
REPLACE(DMPECS,NUC,AL=7777,FL=30000,FLO=0)                        085
REPLACE(PFDUMP,NUC,AL=3,FL=35000,FLO=0)                           086
REPLACE(LDRTEXT+IPTEXT,NUC,AL=0)                                  087
FINISH.                                                           088
LIBRARY(RUN2P3,OLD)                                               089
REPLACE(R23RCVR,NUC,AL=0)                                         090
```

```
FINISH.                                                           091
LIBRARY(SYSMISC,OLD)                                              092
REPLACE(F3ORCVR,NUC,AL=0)                                         093
FINISH.                                                           094
COMPLETE.                                                         095
ENDRUN.                                                           096
7/8/9  END OF RECORD
6/7/8/9  END OF FILE

SCOPE3,CM55000,T7000,MT1.                                         001
COMMENT. THIS JOB CREATES A DEADSTART TAPE FROM THE RUNNING SYSTEM. 002
REQUEST,NEWSYS,HY.          MOUNT SCRATCH TAPE FOR NEW DEADSTART  003
REWIND,NEWSYS.                                                    004
EDITLIB(SYSTEM)                                                   005
UNLOAD(NEWSYS)                                                    006
COMMENT.   *** END OF JOB ***                                     008
7/8/9  END OF RECORD
READY(NEWSYS)                                                     009
REWIND(SYSTEM)                                                    010
TRANSFER(*,SYSTEM)                                                011
SKIPF(2,SYSTEM)                                                   012
INCLUDEF(SYSTEM)                                                  013
INCLUDE(NUCLEUS,SYSTEM,CM)                                        014
INCLUDE(SYSOVL,SYSTEM,DS)                                         015
INCLUDE(FORTRAN,SYSTEM,DS)                                        016
INCLUDE(COBOL,SYSTEM,DS)                                          017
INCLUDE(RUN2P3,SYSTEM,DS)                                         018
INCLUDE(SYSIO,SYSTEM,DS)                                          019
INCLUDE(SYSMISC,SYSTEM,DS)                                        020
INCLUDE(IGS274,SYSTEM,DS)                                         021
COMPLETE.                                                         022
ENDRUN.                                                           023
7/8/9  END OF RECORD
6/7/8/9  END OF FILE

SCOPE4,CM55000,T7000,MT01.                                        001
LABEL(PL1,R,L=SCOPE3P4,D=HY)        LATEST SCOPE PL               002
REWIND(PL1)                                                       003
UPDATE(Q,P=PL1)                                                   004
COMPASS(I=COMFILE,S=0,L=0)                                        005
REWIND(LGO)                                                       006
EDITLIB(SYSTEM)                                                   007
UNLOAD(PL1)                                                       008
COMMENT.  *** END OF JOB ***                                      009
7/8/9  END OF RECORD
*ID SYSTEXT                                                       011
*DELETE CPCTEXT.2                                                 012
         IDENT     SYSTEXT                                        013
*DELETE CPCTEXT.8                                                 014
***      SYSTEXT - SYSTEM TEXT FOR 6000 SCOPE 3 WITH *CPC*.       015
*DELETE CPCTEXT.29                                                016
         END       SYSTEXT                                        017
*C CPCTEXT                                                        018
7/8/9  END OF RECORD
READY(SYSTEM,OLD)                                                 020
LIBRARY(NUCLEUS,OLD)                                              021
```

```
REPLACE(*,LGO)                                                   022
FINISH.                                                          023
COMPLETE.                                                        024
ENDRUN.                                                          025
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

The dayfile output for the SCOPE, COMPASS, and 6RM programs follows:

```
            SCOPE 3.4
05.43.30.VSCOP3W
05.43.30.VSCOPE,CM60000,T1000.
05.43.30.    THIS JOB CHECKS THE INSTALLATION OF
05.43.30.     SCOPE 3.4, 6RM, AND COMPASS
05.43.30.COMPASS(L=0,S=SCPTEXT)
05.43.31. ASSEMBLY COMPLETE.    46400B SCM USED.
05.43.31.     0.121  CPU SECONDS  ASSEMBLY TIME.
05.43.31.LGO.
05.43.33.      1.232 RT SECONDS LOAD TIME
05.43.33. CPCTEXT AND COMPASS ARE IN THE SYSTEM
05.43.33.    JOB PASSED
05.43.33.REWIND(LGO)
05.43.33.COMPASS(L=0,S=IOTEXT)
05.43.34. ASSEMBLY COMPLETE.    46600B SCM USED.
05.43.34.     0.132  CPU SECONDS  ASSEMBLY TIME.
05.43.34.LGO.
05.43.36.      1.315 RT SECONDS LOAD TIME
05.43.36. 6RM IS IN THE SYSTEM
05.43.36.      JOB PASSED
05.43.36.CPA       .505 SEC.
05.43.36.PP       5.354 SEC.
```

RELEASE DESCRIPTION

COMPASS 3.0 is an improved comprehensive assembler program for the 6000 Series systems which provides many new capabilities. COMPASS 3.0 runs under SCOPE 3.4 and requires the same minimum hardware configuration as SCOPE.

RELEASE MATERIALS

The program library for COMPASS 3.0 is known as PL2.

MODIFICATIONS

1. New Built-in Operation Code Synonyms

New Format	Equivalent To
LXi Bj	LXi Bj,Xi
AXi Bj	AXi Bj,Xi
NXi	NXi B0,Xi
NXi Bj	NXi Bj,Xi
ZXi	ZXi B0,Xi
ZXi Bj	ZXi Bj,Xi
UXi	UXi B0,Xi
UXi Bj	UXi Bj,Xi
PXi	PXi B0,Xi
PXi Bj	PXi Bj,Xi

2. New Built-in Micros

BASE	PCOMMENT
CODE	QUAL
JDATE	SEQUENCE
MODLEVEL	

3. New Pseudo Instructions

IFPL	MACHINE
IFMI	IFCP6
ENTRYC	IFCP7
ORGC	IFPP6
REPC	IFPP7

4. Extensions to Old Pseudo Instructions

CODE	PPOP
COMMENT	QUAL
CPOP	USELCM
LIST	

5. New Machine Instructions

MAN	Exchange jump to (MA)
IM	Indirect Move
MD	Indirect Move Descriptor Word (pseudo)
DM	Direct Move
CC	Compare Collated
CU	Compare Uncollated

6. Multiple System Text Capability

Up to seven system text overlays may be used for a COMPASS 3.0 assembler run. They are specified by G and S parameters on the COMPASS control card.

7. New Parameters on COMPASS Control Card

ML	Initial Value of MODLEVEL Micro
PC	Initial Value of PCOMMENT Micro

8. Object Program Binary Format Extensions

The binary card format generated by COMPASS has been extended in Version 3.0. Details are discussed in the Loader Reference Manual, Publication No. 60344200.

CORRECTIONS

All eligible COMPASS 2.0 Programming System Report corrective code through Summary 312 is incorporated in the release tape.

GENERAL DESCRIPTION

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

INSTALLATION PARAMETERS

The installation parameter definitions are in alphabetic order on pages 5-8 of the COMPASS 3.0 assembly listing. Parameters that an installation may change are described below.

INTMUL - integer multiply instruction usage. In the released system, an integer multiply OPDEF is used. If the integer multiply hardware feature is installed:

```
*D CMP30.114
 INTMUL     EQU 1
```

LIBRARY - library name for overlay. In the released system, the (1,0) overlay COMP2$ must be in a library in the job's global library set, or in the NUCLEUS library. To make COMPASS load its overlay from a specific library:

```
*D CMP30.120
 LIBRARY     MICRO 1,,*libname*
```

Changing this parameter from its default state will necessitate change in the installation deck EDITLIB records.

MODEL - CDC CYBER 70 Series model on which COMPASS runs. Model 74 is assumed in the released system. To change:

```
*D CMP30.152
 MODEL MICRO 1,,*xx*
```

where xx may be 72 or 73.

TIMEMSG - assembly time dayfile message option. In the released system, COMPASS issues a dayfile message giving the total CPU time at the end of a batch of assemblies. To suppress this message:

```
*D CMP30.240
 TIMEMSG EQU 0
```

INSTALLATION PROCEDURES

The COMPASS 3.0 release tape contains four files:

File 1 program library
File 2 assembled binary
File 3 installation deck CMP1A
File 4 installation deck CMP2A

CMP1A is a maintenance deck which can be used to create a revised program library and binary file. CMP2A can be used to enter COMPASS 3.0 into the running system from either the released PL2 or a tape created by CMP1A. After deck CMP2A has completed, job SCOPE3 (discussed in Section 1) should be run to create a deadstart tape of the running system.

To obtain the decks included as files 3-4, perform the following job:

```
Job card.
REQUEST(PL2,E)
REWIND(PL2)
SKIPF(PL2,2,17)
COPYBF(PL2,PUNCH,2)
RETURN(PL2)
6/7/8/9
```

The installation decks are shown below.

```
CMP1A,CM54000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF COMPASS
COMMENT. THE FIRST FILE OF THE NEW PL2 WILL BE THE NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY OF COMPASS
LABEL(PL2IN,R,L=COMPASS3P0*3P4,D=HI)    COMPASS OLDPL
REQUEST,PL2,N,HI.     SCRATCH FOR NEW PL2
LABEL(PL2,W,L=COMPASS3P0*3P4,D=HI)
REWIND(PL2IN,PL2)
UPDATE(F,P=PL2IN,N=PL2,E,X)
UNLOAD(PL2IN)
COMPASS(I=COMFILE,S=0,B=COMTEXT,L=0)
COMPASS(I=COMFILE,G=COMTEXT,L=0)
REWIND(LGO)
SKIPF(PL2,1,17)
COPYBF(LGO,PL2)
UNLOAD(PL2)
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE


CMP2A,CM54000,T7000,MT01.
COMMENT. THIS JOB EDITLIBS THE COMPASS 3.0 BINARIES FROM THE PL2 TAPE
COMMENT. INTO THE RUNNING SYSTEM. EITHER THE RELEASED VERSION OR THE
COMMENT. VERSION CREATED BY DECK CMP1A MAY BE USED.
LABEL(PL2,R,L=COMPASS3P0*3P4,D=HI)  MOUNT COMPASS V3.0 OLDPL
REWIND,PL2.
SKIPF(PL2,1,17)              SKIP OLDPL
COPYBF(PL2,COMP)             COMPASS BINARIES
REWIND,PL2,COMP.
UNLOAD(PL2)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(COMPASS,COMP,AL=3,FL=50000,FLO=1)
REPLACE($COMP2$$$,COMP,AL=0)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

RELEASE DESCRIPTION

To differentiate between 6000 and 7000 Series Record Manager, the term 6RM is used in the following description.

6RM 1.0 operates under SCOPE 3.4, on the same minimum configuration as SCOPE.

A program library for 6RM is contained on the release program library tape PL3. 6RM occurs first on a tape shared with SCOPE Indexed Sequential (IS), SCOPE Direct Access (DA), and the 8-Bit Subroutines.

The structure of the release format PL3 tape is as follows:

Files 1-5 6RM 1.0

1	6RM program library in UPDATE format
2	SYSTEXT, IOTEXT, TXT6RM binary
3	I/O modules binary
4	FILE control card processor absolute binary
5	FILE control card processor relocatable binary

Files 6-7 IS 2.0

6	IS 2.0 program library
7	IS 2.0 binary

Files 8-9 DA 1.0

8	DA 1.0 program library
9	DA 1.0 binary

Files 10-13 8-Bit Routines

10	9-Bit Routines program library
11	8-Bit Routines binary
12	COPY8P relocatable binary
13	COPY8P absolute binary

Files 14-26 Installation decks for 6RM, IS,DA, 8-Bit Routines

14-15	6RM installation decks
16-19	IS 2.0 installation and verification decks
20-22	DA 1.0 installation and verification decks
23-26	8-Bit Routine installation and verification decks

The procedures defined herein are intended for the installation of 6RM I/O routines in relocatable binary format; methods for assembling 6RM Sequential and Word Addressable I/O routines into absolute programs are described in an appendix of the Record Manager Reference Manual.

DEFICIENCIES

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

Checksumming of I type blocks is not available.

To extend most 6RM files, which are also permanent files, MODIFY permission is required.

Calls to the 7000 Record Manager are not available.

If C blocked, non-W record, SCOPE tapes are copied to S tapes, section boundaries may be lost.

ADDITIONAL INFORMATION

6RM I/O modules are divided into two parts:

Basic Access Modules

These routines control selective loading based on file organization. They contain jump vectors directing a user call to the I/O code appropriate to the file organization selected. Their program names have an RM suffix.

Sequential and Word Addressable I/O Modules

Texts: The I/O macro text included with the 6RM program library is IOTEXT, which is identical to the default SYSTEXT. It consists of, but is not limited to the macros included in the following table. (Some auxiliary macros exist which are not supported at the user level.)

MACRO NAME	SYSTEM	REFERENCE	COMDECK
FILE	6RM	RM Ref. Man.	6RMCOM
FETCH			
STORE			
OPENM			
CLOSEM			
GET			
GETP			
GETN	IS/DA		
PUT	6RM		
PUTP			
REPLACE			
DELETE			
ENDFILE			
SKIPdu			
d≡F/B,u≡L/P/F			
SEEK	IS/DA		
REWINDM	6RM		
WOER			
WTMK			
GETL			
PUTL			
CLOSEL			
CHECK			
ABORT	SCOPE	SCOPE 3.4 Ref. Man.*	ACTCOM
CHECKPT			
CLOCK			
CONTRLC			
DATE			
DISPOSE			
ENDRUN			
JDATE			
FILESTAT			
LOADREQ			
MEMORY			
MESSAGE			
RECALL			
RECOVR			
REQUEST			
RTIME			
SYSTEM			
TIME			
TRANSR			

* These macros are source compatible with the corresponding macros on CPCTEXT (SCOPE 3.3 SYSTEXT), but they do not generate the same code.

INSTALLATION PARAMETERS

The installation parameters, described below, permit a certain amount of tailoring. To facilitate writing the UPDATE cards, each installation parameter has a unique, mnemonic UPDATE identifier. To change the parameter from its default value to a user value, the following process is required:

```
*DELETE   <mnemonic>.1

≡<parameter>≡  EQU  <user-value>
```

Updates which set installation parameters always should be done under UPDATE IDENTS of the following format:

```
*IDENT    URM<mmddyy>

 mm       month

 dd       day

 yy       year
```

Parameter	Mnemonic Update ID	Description	Default Value
DBG	DBG	Causes extensive debug code to be assembled if defined.	Off
MCTL	MCTL	Memory catalog table length. MCTL/2 is the maximum number of files allowed open at one time.	100 decimal
LBLIM	LBLIM	Length of label buffer. Size limit of a user label string. Each user label requires 9 words. LBLIM should be n*9+1, where n is the maximum number of labels permitted (HDR1-9,...).	10 decimal

INSTALLATION PROCEDURES

File 1 of PL3 contains the 6RM program library.

Files 2-5 are pre-assembled 6RM binaries assembled with default installation parameters. Files 14 and 15 contain decks necessary to install 6RM. They may be obtained as follows:

```
Job card
REQUEST(PL3,E)          MOUNT TAPE PL3
SKIPF(PL3,13,17)        SKIP 6RM, IS, DA, and 8-Bit Subroutine
                        program libraries
COPYBF(PL3,PUNCH,2)     TWO INSTALL DECKS
UNLOAD(PL3)
6/7/8/9
```

The installation decks are listed below.

Deck SIXRM1 is a maintenance deck which allows regeneration of the 6RM portion of PL3. This deck updates the program library, assembles 6RM, and places the binary on the new PL as supplemental files. User selected installation parameters should be modified at the indicated place in SIXRM1. Deck SIXRM1 requires access to the SCOPE program library to acquire the common deck ACTCOM used by the 6RM system texts.

Deck SIXRM2 adds 6RM to the running system, either from the released PL3 or a PL3 created by deck SIXRM1. Then deck SCOPE3, described in the SCOPE section of this document, can be run to create a deadstart tape of the running system.

```
SIXRM1,CM60000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF 6RM
COMMENT. AND COPIES THE OLDPL AND BINARY OF IS DA AND BIT8
COMMENT. TO THE NEWPL3. THE NEW PL3 WILL CONSIST OF THIRTEEN FILES
COMMENT. THE FIRST FILE WILL BE THE 6RM NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARIES CF THE TEXTS
COMMENT. THE THIRD FILE WILL BE THE BINARY OF THE OBJECT TIME ROUTINES
COMMENT. THE FOURTH FILE WILL BE THE BINARY OF THE OVERLAY FILE
COMMENT. THE FIFTH FILE WILL BE THE RELOCATABLE BINARY OF FILE
COMMENT. THE LAST 8 FILES WILL BE THE PLS AND BINARIES OF IS DA AND BIT8
LABEL(PL1,R,L=SCOPE3P4,D=HY)       LATEST SCOPE PL
REWIND(PL1)
UPDATE(Q,F=PL1,N=RANPL,C=0)
UNLOAD(PL1)
LABEL(PL3IN,R,L=RM*3P4,D=HI)  6RM CLDPL
REQUEST,PL3,N,HI.     SCRATCH FOR NEW PL3
LABEL(PL3,W,L=RM*3P4,D=HI)
REWIND(PL3IN,PL3)
UPDATE(F,P=PL3IN,N=PL3,X)
COMPASS(I=CCMFILE,S=0,B=TEXTS,L=0,X=RANPL)  SYSTEXT, IOTEXT, TXT6RM
COMPASS(I=CCMFILE,G=TEXTS/TXT6RM,S=PFMTEXT,B=IOMODS,L=0) I/O MODS
COMPASS(I=CCMFILE,G=TEXTS/TXT6RM,B=FILEC,L=0)  RELOCATABLE BINARY OF FILE
LIBRARY(SYSIO)
LOAD(FILEC)        GENERATION OF OVERLAY FILE
NOGO.
REWIND,PL3,IOMODS,FILE,TEXTS.
REWIND(FILEC)
SKIPF(PL3,1,17)
COPYBF(TEXTS,PL3)
COPYBF(IOMODS,PL3)
COPYBF(FILE,PL3)
COPYBF(FILEC,PL3)
SKIPF(PL3IN,5,17)
COPYBF(PL3IN,PL3,8)          COPY IS DA AND BIT8 TO TAPE
UNLOAD(PL3IN,PL3)
7/8/9  END CF RECORD
*C CPCTEXT
7/8/9  END CF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
SIXRM2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS ALL OF THE 6RM BINARIES INTO THE RUNNING
COMMENT. SYSTEM FROM THE RELEASED VERSION OF PL3 OR THE VERSION CREATED
COMMENT. BY THE JOB SIXRM1.
LABEL(PL3,R,L=RM*3P4,D=HI)    MOUNT PL3.
SKIPF(PL3,1,17)               SKIP OLDPL
COPYBF(PL3,TEXTS)             SYSTEXT, IOTEXT,TXT6RM
COPYBF(PL3,IOMODS)            I/O MODULES
COPYBF(PL3,FILE)              FILE GENERATION
UNLOAD(PL3)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(*,TEXTS,AL=0)
REPLACE(*,FILE,AL=3,FL=2000,FLO=0)
FINISH.
LIBRARY(SYSIO,OLD)
REPLACE(*,IOMODS)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

RELEASE DESCRIPTION

Operating as a functional module of 6RM, SCOPE Indexed Sequential 2.0(IS) provides for the creation and maintenance of a random storage file with records that can be retrieved by key as well as sequentially. An indexed sequential file created by IS(6RM) can be processed only by IS(6RM).

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

Interface routines provided allow the execution of SIS V1.0 programs that exist in either source language or binary format.

RELEASE MATERIALS

The SCOPE Indexed Sequential module is contained on program library PL3. This tape also contains 6RM, DA, and the 8-Bit Subroutines.

A complete catalog of PL3 contents may be found in section I-3.

MODIFICATIONS AND DEFICIENCIES

Known Deficiencies and Limitations:

1. If IS is used in a program with overlays, it should be included in the main overlay. The IS routines must not be overlaid.
2. IS will not be reliable if permanent file installation parameter IP.MREWR is in effect, as this option permits access to a file from a control point while another control point is modifying that file.

INSTALLATION PARAMETERS

This system contains parameter values which are effective when the user does not supply settings. These values may be altered during installation as explained in the next section.

The default parameters are defined on the program library tape PL3 in the common deck SISCOMM. Their definitions, default values, ranges of acceptable decimal values, and UPDATE sequence locations are as follows:

DAHDRSZW	EQU	1	Data block header size in words, range 1-31, (SISCOMM.12)
DFBKGFAC	EQU	2	Default data record blocking factor, range 1-4095 (SISCOMM.16)
DFDAPADP	EQU	0	Default data block padding factor, range 0-99 (SISCOMM.17)
DFERRLIM	EQU	26	Maximum number of trivial errors + 1. For SIS 1.0 Programs only, range 1-32767 (SISCOMM.18).
DFIBKSZW	EQU	511	Default index block size in words, range 1-23767 (SISCOMM.19)
DFINPADP	EQU	5	Default index padding factor, range 0-99 (SISCOMM.20)
DFNRLVLS	EQU	1	Default number of index levels, range 1-63 (SISCOMM.21)
KEYLIMIT	EQU	255	Maximum key size in characters, range 1-511 (SISCOMM.31)
TOTFILES	EQU	10	Maximum number of active IS files per run. Defines an internal table size in words - no practical limit. (SISCOMM.338)

INSTALLATION PROCEDURES

PL3 contains 18 files: files 6, 7, 16, 17, 18, and 19 pertain to IS 2.0. File 6 contains the IS program library which will generate four COMPILE file records:

COMPASS code for all IS functions

COMPASS code for IS utilities

FORTRAN Extended code for the IS utilities

COMPASS code for the interface routines to SIS 1.0

File 7 contains the binary produced by assembling the contents of file 6. Files 16 and 17 are installation decks IS1 and IS2 as listed below. Files 18 and 19 contain the verification programs. These decks can be obtained by executing the following job:

```
Job card.
REQUEST(PL3,E)               MOUNT PL3
REWIND(PL3)
SKIPF(PL3,15,17)             SKIP PL AND BINARY DECKS
COPYBF(PL3,PUNCH,2)          PUNCH INSTALLATION DECKS
COPYBF(PL3,PUNCH,2)          PUNCH VERIFICATION PROGRAM DECK
UNLOAD(PL3)
6/7/8/9
```

IS1 is a maintenance deck which can be used to create a revised program library and binary file containing modifications. IS2 can be used to enter IS into the running system from either the released tape or a tape created by IS1. Job SCOPE3 described in Section 1 should be run to capture a deadstart tape containing IS. If the SIS 1.0 interface code is not desired, remove the last COMPASS card in deck IS1.

To obtain a running version of IS, DA must be in the system. The DA peripheral processor routine MSD is used by IS to process error messages. The DA routine TRC is needed to process trace messages.

```
IS1,CM63500,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF IS
COMMENT. AND COPIES 6RM DA AND BIT8 TO THE NEW PL3A.
COMMENT. THE NEW PL3A WILL CONSIST OF THIRTEEN FILES.
COMMENT. THE FIRST FIVE FILES WILL BE 6RM
COMMENT. FILE SIX WILL BE THE IS NEWPL
COMMENT. FILE SEVEN WILL BE THE IS BINARY
COMMENT. THE LAST SIX FILES WILL BE THE PLS AND BINARIES OF DA AND BIT8
LABEL(PL3AIN,R,L=RM*3P4,D=HI) IS OLDPL
REQUEST,PL3A,N,HI.    SCRATCH FOR NEW PL3A
LABEL(PL3A,W,L=RM*3P4,D=HI)
REWIND(PL3AIN,PL3A)
COPYBF(PL3AIN,PL3A,5)        COPY 6RM TO TAPE
UPDATE(F,P=PL3AIN,N=PL3A,R)
REWIND,COMPILE.
COMPASS(I=COMPILE,B=NEWBIN,S=TXT6RM,L=0,S=IPTEXT)
COMPASS(I=COMPILE,B=NEWBIN,S=TXT6RM,L=0)
FTN(I=COMPILE,B=NEWBIN,L=0,SYSEDIT)
COMPASS(I=COMPILE,B=NEWBIN,S=TXT6RM,S=IOTEXT,L=0)
REWIND(NEWBIN)
SKIPF(PL3A,1,17)
COPYBF(NEWBIN,PL3A,1)
SKIPF(PL3AIN,2,17)
COPYBF(PL3AIN,PL3A,6)       COPY DA AND BIT8 TO TAPE
UNLOAD(PL3A,PL3AIN)
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE

IS2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS THE SIS BINARIES FROM THE PL3 TAPE, EITHER
COMMENT. THE RELEASED VERSION OR THE VERSION CREATED BY DECK IS1, INTO
COMMENT. THE RUNNING SYSTEM.
LABEL(PL3A,R,L=RM*3P4,D=HI)  MOUNT PL3 CONTAINING SIS
REWIND,PL3A.
SKIPF(PL3A,6,17)
COPYBR(PL3A,SIS,35)      ACOMMON-INSERT
COPYBR(PL3A,RUN,1)       RUNCALL
COPYBF(PL3A,SIS)         FTNCALL-TERMNAT
REWIND(SIS,RUN)
UNLOAD(PL3A)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(ESTMATE,SIS,AL=3,FL=37000,FLO=1)
REPLACE(SISTAT,SIS,AL=3,FL=21000,FLO=1)
REWIND(SIS)
FINISH.
LIBRARY(SYSIO,OLD)
REPLACE(*,SIS)
DELETE(SISTAT)
DELETE(ESTMATE)
FINISH.
LIBRARY(RUN2P3,OLD)
REPLACE(*,RUN)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAMS

FORTRAN Extended and COBOL 4.0 must be installed before the corresponding installation verification program on files 14 and 15 can be run. Comment cards describe the purpose of each deck.

Dayfile output obtained by running the decks included on files 14 and 15 of the release tape is as follows:

```
               SCOPE 3.4                         07/14/72
21.14.29.VSIS100
21.14.29.VSIS1,CM60000,T7000,P37.
21.14.29.   ******** S I S   V 1      ************
21.14.29.COMPASS(S=CPCTEXT)
21.14.32. ASSEMBLY COMPLETE.   43500B SCM USED.
21.14.32.    2.376  CPU SECONDS  ASSEMBLY TIME.
21.14.32.LIBRARY(SYSIO)
21.14.32.LGO.
21.14.35.     2.898 RT SECONDS LOAD TIME
21.14.38.   ***** T E S T   P A S S E D  *******
21.14.38.REWIND(SISMESS)
21.14.39.       .488 RT SECONDS LOAD TIME
21.14.39.REWIND(OUTPUT)
21.14.39.       .493 RT SECONDS LOAD TIME
21.14.39.RETURN(SISFILE)
21.14.40.       .498 RT SECONDS LOAD TIME
21.14.40.REWIND(LGO)
21.14.41.       .505 RT SECONDS LOAD TIME
21.14.41.FTN.
21.14.42.       .612 RT SECONDS LOAD TIME
21.14.43.       .264 CP SECONDS COMPILATION TIME
21.14.43.LGO.
21.14.51.     7.101 RT SECONDS LOAD TIME
21.14.54.     END  TEST
21.14.54.    **** T E S T   P A S S E D  *****
21.14.54.REWIND(SISMESS)
21.14.55.       .769 RT SECONDS LOAD TIME
21.14.55.REWIND(OUTPUT)
21.14.56.       .827 RT SECONDS LOAD TIME
21.14.56.REWIND(LGO)
21.14.57.     1.293 RT SECONDS LOAD TIME
21.14.58.RETURN(SISFILE)
21.14.59.     1.310 RT SECONDS LOAD TIME
21.15.00.COBOL(LM,Z)
21.15.02.     2.087 RT SECONDS LOAD TIME
21.15.04.COMPILING SISTEST
21.15.15. 000 E AND     T/U  DIAGNOSTICS ISSUED
21.15.16. FIELD LENGTH NEEDED FOR COBOL 052600
21.15.16.       .648 CP SECONDS COMPILATION TIME
21.15.16.END COBOL
21.15.16.LGO.
21.15.26.     9.894 RT SECONDS LOAD TIME
21.15.33.    **** TEST PASSED ****
21.15.33.REWIND(SISMESS)
21.15.36.     1.965 RT SECONDS LOAD TIME
21.15.36.REWIND(OUTPUT)
21.15.45.     1.432 RT SECONDS LOAD TIME
21.15.49.COPYBF(EOF,OUTPUT)
21.16.41.     1.438 RT SECONDS LOAD TIME
21.16.42.EOF/EOI ENCOUNTERED
21.16.42.CPA      9.527 SEC.
21.16.42.PP      30.417 SEC.
21.16.42.IO       3.865 SEC.
```

```
21.14.45.VSIS200
21.14.45.VSIS2,CM60000,T7000,P37.
21.14.45.            ***  S I S    V 2    ***
21.14.45.    THIS IS THE COMPASS VERIFICATION TES
21.14.45.T DECK.....SIS V2
21.14.45.LIBRARY(SYSIO)
21.14.46.REWIND(LGO)
21.14.47.        .803 RT SECONDS LOAD TIME
21.14.47.COMPASS(S=IOTEXT)
21.14.48.        .864 RT SECONDS LOAD TIME
21.14.54.           1 WARNING MESSAGE IN  I2P01
21.14.54. ASSEMBLY COMPLETE.   47600B SCM USED.
21.14.54.    3.377  CPU SECONDS  ASSEMBLY TIME.
21.14.54.LGO.
21.14.59.     4.373 RT SECONDS LOAD TIME
21.15.03.   ***** T E S T  P A S S E D  *******
21.15.03.REWIND,ZZZZZEF.
21.15.03.       .688 RT SECONDS LOAD TIME
21.15.03.REWIND(OUTPUT)
21.15.04.       .644 RT SECONDS LOAD TIME
21.15.04.REWIND(LGO)
21.15.05.       .802 RT SECONDS LOAD TIME
21.15.05.RETURN(GENFILE)
21.15.08.     2.034 RT SECONDS LOAD TIME
21.15.08.FTN.
21.15.12.     3.736 RT SECONDS LOAD TIME
21.15.18.       .593 CP SECONDS COMPILATION TIME
21.15.18.LIBRARY(NUCLEUS,SYSOVL)
21.15.19.       .863 RT SECONDS LOAD TIME
21.15.19.LGO.
21.15.43.    13.441 RT SECONDS LOAD TIME
21.17.01.     END  INSTEST
21.17.01.  ***** TEST PASSED *******
21.17.01.REWIND,ZZZZZEF.
21.17.01.REWIND(OUTPUT)
21.17.01.REWIND(LGO)
21.17.01.RETURN(SISFILE)
21.17.02.COBOL.
21.17.04.     1.982 RT SECONDS LOAD TIME
21.17.07.COMPILING BUILD
21.17.15. 000 E AND     T/U  DIAGNOSTICS ISSUED
21.17.15. FIELD LENGTH NEEDED FOR COBOL 052600
21.17.15.       .478 CP SECONDS COMPILATION TIME
21.17.15.END COBOL
21.17.15.LGO.
21.17.25.     NON-FATAL LOADER ERRORS - SEE MAP
21.17.27.    11.578 RT SECONDS LOAD TIME
21.17.43.   ******** T E S T  P A S S E D  ******
21.17.43.***
21.17.43.REWIND,OUTPUT.
21.17.43.REWIND,ZZZZZEF.
21.17.44.COPYBF(EOF,OUTPUT)
21.17.45.EOF/EOI ENCOUNTERED
21.17.46.CPA    10.021 SEC.
21.17.46.PP     46.539 SEC.
21.17.46.IO      4.221 SEC.
```

RELEASE DESCRIPTION

Operating as a functional module of 6RM, SCOPE Direct Access (DA) provides all the routines requisite to creating, updating, and accessing random files on mass storage. It consists of a related set of central processor routines which are loaded, as required, in the user's field length. Additionally, DA contains one PP routine which is used for diagnostic processing.

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

A create utility routine is available for use in creating DA files efficiently.

A direct access file created by DA(6RM) can be processed only by DA(6RM) operating under the SCOPE 3.4 operating system.

RELEASE MATERIALS

DA 1.0 is contained on program library tape PL3. PL3 also contains 6RM 1.0, IS 2.0, and the 8-Bit Subroutines.

A complete catalog of PL3 contents may be found in section I-3.

MODIFICATIONS AND DEFICIENCIES

Known Deficiencies and Limitations

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

2. DA will not be reliable if permanent file installation parameter IP.MREWR is in effect, as this option permits access to a file from one control point while another control point is modifying that file.

3. The create utility requires that SORT MERGE be installed. If SORT/MERGE is not available, comparable DA files can be created through explicit 6RM calls, at the expense of appreciably greater creation time.

INSTALLATION PARAMETERS

DA contains a single parameter value which is effective when the user does not supply his own setting. This value may be altered during installation of DA. The default parameter is defined on the program library tape in the common deck, SDACOM. Its definition, default value, range of acceptable decimal values, and UPDATE sequence location is:

```
BLKHDL    EQU  1          Block header length (CM words), range 1-31 (SDACOM.38)
```

If the above parameter is out of range, the following message will be issued:

```
INSTALLATION PARAM BLKHDL OUTSIDE 1-31 RANGE
```

INSTALLATION PROCEDURES

PL3 contains 26 files: files 8, 9, 20, 21, and 22 pertain to DA. File 8 contains a program library that will generate four records on the COMPILE file:

COMPASS code for all DA CP routines
COMPASS code for the DA PP diagnostic routine
COMPASS code for the FORTRAN interface
COMPASS code for the key analysis and create utilities

File 9 contains the binary produced by assembling the contents of file 8. Files 20 and 21 contain installation decks DA1 and DA2. File 22 contains the installation verification programs. These decks can be obtained by executing the following job:

```
Job card.
REQUEST(PL3,E)              MOUNT DA PROGRAM LIBRARY
REWIND(PL3)
SKIPF(PL3,19,17)            SKIP UPDATE AND BINARY DECKS
COPYBF(PL3,PUNCH,2)         PUNCH INSTALLATION DECKS
COPYBF(PL3,PUNCH,1)         PUNCH VERIFICATION PROGRAM DECKS
UNLOAD(PL3)
6/7/8/9
```

Successful assembly of the routines processed by deck DA1 card 020 requires that SORT MERGE be previously installed (see limitation 3). If SMTEXT is not present in the running system when deck DA1 is run, three assembly errors will appear in routine DCREATE, and deck DA2 will yield an EDITLIB diagnostic that cannot be satisfied in the interval.

DA1 is a maintenance deck which can be used to create a revised program library and binary file containing modifications. DA2 can be used to enter DA into the running system from either the released tape or a tape created by SDA1. Job SCOPE3, described in Section 1, should be run to capture a deadstart tape containing DA.

NOTE: If the key analysis utility is to be used as an owncode exit from FORM (see Section 6), FORM must be present in the system.

```
DA1,CM60000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF DA
COMMENT. AND COPIES 6RM AND IS TO THE NEW PL3B.
COMMENT. THE NEW PL3B WILL CONSIST OF THIRTEEN FILES.
COMMENT. THE FIRST SEVEN FILES WILL BE THE OLDPLS AND BINARIES
COMMENT. OF 6RM AND IS
COMMENT. FILE EIGHT WILL BE THE DA NEWPL
COMMENT. FILE NINE WILL BE THE BINARY OF DA
COMMENT. THE LAST FOUR FILES WILL BE THE PL AND BINARY OF BIT8.
LABEL(FL3BIN,R,L=RM*3P4,D=HI) DA OLDPL
REQUEST,FL3B,N,HI.            SCRATCH FOR NEW PL3B
LABEL(FL3B,W,L=RM*3P4,D=HI)
REWIND(FL3BIN,FL3B)
COPYBF(FL3BIN,FL3B,7)         COPY 6RM AND IS TO TAPE
UPDATE(F,P=FL3BIN,N=PL3B,R,X)
REWIND,COMPILE.
COMPASS(I=COMPILE,L=0,S=TXT6RM)
COMPASS(I=COMPILE,S=PPTEXT,L=0)
COMPASS(I=COMPILE,S=IOTEXT,L=0)
COMPASS(I=COMPILE,S=IOTEXT,S=CPCTEXT,S=LDRTEXT,S=SMTEXT,L=0)
REWIND(LGO)
SKIPF(FL3B,1,17)
COPYBF(LGO,FL3B)
REWIND(FL3BIN)
SKIPF(FL3BIN,9,17)
COPYBF(FL3BIN,FL3B,4)             COPY BIT8 TO TAPE
UNLOAD(FL3BIN,FL3B)
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
DA2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS THE SDA BINARIES FROM THE PL3 TAPE, EITHER
COMMENT. THE RELEASED VERSION OR THE VERSION CREATED BY THE DECK DA1,
COMMENT. INTO THE RUNNING SYSTEM.
LABEL(PL3B,R,L=RM*3P4,D=HI)  MOUNT PL3 CONTAINING SDA
REWIND,PL3B.
SKIPF(PL3B,8,17)
COPYBF(PL3B,SDA)              SDA BINARIES
REWIND(SDA)
UNLOAD(PL3B)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
REPLACE(MSD,SDA)
REWIND(SDA)
LIBRARY(NUCLEUS,OLD)
REPLACE(KYAN+DCREATE,SDA,AL=3,FL=20000,FLO=1)
FINISH.
LIBRARY(SYSIO,OLD)
REWIND(SDA)
REPLACE(*,SDA)
DELETE(MSD)
DELETE(KYAN)
DELETE(DCREATE)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAMS

FORTRAN Extended 4.0 and COBOL 4.0 must be installed before the corresponding installation verification program on file 18 can be run. Comment cards describe the purpose of the deck.

Dayfile output of the verification programs is as follows:

```
               SCOPE 3.4
21.14.58.VSDA000
21.14.58.VSDA,CM65000,T7000.
21.14.58.LIBRARY(SYSIO)
21.14.58.COMPASS(S=IOTEXT)
21.14.59.       .804 RT SECONDS LOAD TIME
21.15.07.           1 WARNING MESSAGE IN  INSTALL
21.15.07. ASSEMBLY COMPLETE.   47600B SCM USED.
21.15.07.     4.276  CPU SECONDS  ASSEMBLY TIME.
21.15.07. *** INSTALLATION TEST ***
21.15.07. LANGUAGE- COMPASS
21.15.07. PRODUCT- SDA
21.15.07.SWITCH(1)
21.15.07.LGO.
```

```
21.15.15.      6.737 RT SECONDS LOAD TIME
21.15.45. ***********TEST PASSED ***************
21.15.45.REWIND,ZZZZZEF.
21.15.47.      2.009 RT SECONDS LOAD TIME
21.15.48.REWIND(OUTPUT)
21.16.05.      9.565 RT SECONDS LOAD TIME
21.16.05.REWIND(LGO)
21.17.36.       .477 RT SECONDS LOAD TIME
21.17.36.FTN.
21.17.55.      1.994 RT SECONDS LOAD TIME
21.17.59.       .419 CP SECONDS COMPILATION TIME
21.17.59. *** INSTALLATION TEST ***
21.17.59. LANGUAGE- FORTRAN
21.17.59. PRODUCT- SDA
21.17.59.LGO.
21.18.12.     12.583 RT SECONDS LOAD TIME
21.18.47.      END  INSTALL
21.18.47. ***********TEST PASSED ***************
21.18.47.REWIND,ZZZZZEF.
21.18.49.      1.783 RT SECONDS LOAD TIME
21.18.49.REWIND(OUTPUT)
21.18.51.      1.750 RT SECONDS LOAD TIME
21.18.51. *** INSTALLATION TEST ***
21.18.51. LANGUAGE- COBOL
21.18.51. PRODUCT- SDA
21.18.51.REWIND(LGO)
21.18.53.      1.362 RT SECONDS LOAD TIME
21.18.53.COBOL.
21.18.57.      3.412 RT SECONDS LOAD TIME
21.19.00.COMPILING PTEST2
21.19.05. 000 E AND     T/U  DIAGNOSTICS ISSUED
21.19.05. FIELD LENGTH NEEDED FOR COBOL 052500
21.19.05.       .512 CP SECONDS COMPILATION TIME
21.19.05.END COBOL
21.19.05.LGO.
21.19.15.      NON-FATAL LOADER ERRORS - SEE MAP
21.19.16.     10.671 RT SECONDS LOAD TIME
21.19.50. ***********TEST PASSED ***************
21.19.50.REWIND,ZZZZZEF.
21.19.51.       .985 RT SECONDS LOAD TIME
21.19.51.REWIND(OUTPUT)
21.19.52.       .997 RT SECONDS LOAD TIME
21.19.53.COPYBF(EOF,OUTPUT)
21.19.54.      1.008 RT SECONDS LOAD TIME
21.19.55.EOF/EOI ENCOUNTERED
21.19.55.CPA     11.383 SEC.
21.19.55.PP      62.271 SEC.
```

FORM VERSION 1.0

GENERAL DESCRIPTION

FORM is a general purpose utility routine operating on records supplied by a user; it is capable of performing the following functions:

Conversion to and from System/360 format
Record selection according to data content
Record redefinition as to data type, content, and organization
File reorganization
Sequence numbering
Print reformatting

RELEASE MATERIALS

FORM 1.0 is contained on program library PL4.

HARDWARE CONFIGURATION

FORM requires the same minimum hardware configuration as SCOPE.

INSTALLATION PARAMETER

FORM will reference IP.CSET at installation time and provide the necessary options as defined for the character set.

INSTALLATION PROCEDURES

The FORM release tape contains five files: file 1, program library; file 2, compiled binary; files 3 and 4, installation decks; and file 5, the verification program.

FORM1 (file 3) is a maintenance deck which can be used to create a revised program library and binary. FORM2 (file 4) may be used to add FORM to the running system from either the released tape or a tape prepared by FORM1. To obtain these three decks, execute a job of the type:

```
Job card.
REQUEST(PL4,E)
REWIND(PL4)
SKIPF(PL4,2,17)         SKIP PL and BINARY
COPYBF(PL4,PUNCH,2)     PUNCH INSTALLATION DECKS
COPYBF(PL4,PUNCH,1)     PUNCH VERIFICATION DECK
UNLOAD(PL4)
6/7/8/9
```

Following execution of FORM2, Job SCOPE3, described in Section 1, may be run to create a deadstart tape containing FORM.

```
FORM1,CM70000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF FORM.
COMMENT. THE FIRST FILE OF THE NEW PL4 WILL BE THE NEWPL
COMMENT. AND THE SECOND FILE WILL BE THE BINARY
LABEL(PL4IN,R,L=FORM1P0*3P4,D=HI) FORM OLDPL
REQUEST,PL4,N,HI.     SCRATCH FOR NEW PL4
LABEL(PL4,W,L=FORM1P0*3P4,D=HI)
REWIND(PL4IN,PL4)
UPDATE(F,P=PL4IN,N=PL4)
UNLOAD(PL4IN)
SYMPL(I=COMPILE,L=0)
COMPASS(I=COMPILE,S=IOTEXT,S=IPTEXT,L=0)
REWIND(LGO)
SKIPF(PL4,1,17)
COPYBF(LGO,PL4)
UNLOAD(PL4)
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE


FORM2,CM55000,T7000,MT1.
COMMENT. THIS JOB TAKES THE RELEASED VERSION OF FORM OR THE VERSION
COMMENT. CREATED BY THE JOB FORM1 AND EDITLIBS THE BINARIES FROM PL14
COMMENT. INTO THE RUNNING SYSTEM.
LABEL(PL4,R,L=FORM1P0*3P4,D=HI)    MOUNT PL4 TAPE
REWIND,PL4.
SKIPF(PL4,1,17)              SKIP OLDPL
COPYBR(PL4,FMCC,1)           FORM CONTROL CARD CALLABLE ROUTINE
COPYBF(PL4,FML)              FORM LIBRARY
REWIND,FMCC,FML,PL4.
UNLOAD(PL4)
EDITLIB(SYSTEM)
COMMENT. *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(FORM,FMCC,AL=3,FL=53000,FLO=1)
FINISH.
LIBRARY(SYSIO,OLD)
REPLACE(*,FML,AL=0)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

After installation deck FORM2 has completed, the FORM verification program should be run. Dayfile output from this job should match the following.

```
              SCOPE 3.4
05.44.21.VFORM3Y
05.44.21.VFORM,CM65000,T50.
05.44.21.FORM.
05.44.28.      7.118 RT SECONDS LOAD TIME
05.44.50.     END OF JOB
05.44.50.CPA      1.728 SEC.
05.44.50.PP       8.315 SEC.
```

RELEASE DESCRIPTION

CE Diagnostics require the same hardware configuration as SCOPE 3.4.

RELEASE MATERIALS

CE Diagnostics are released on program library tape PL5.

NEW FEATURES

A new magnetic tape test (MTT) has been added to comply with the new tape subsystem testing standard.

A new random instruction test (ALX) has been added also which is basically the same as ALS with the exception that it does not utilize a store after store. Essentially ALX is a faster version of ALS.

Two new routines (CEDIAG and DLE) enable all non-RMS diagnostics to be called into execution in the same manner. Central processor routine CEDIAG uses the central programmable L display to allow the operator to assign equipments to this control point for future testing by a diagnostic program. CP routine CEDIAG uses PPU program DLE to pass equipment status table information and assign operator requested I/O devices. As a result of new programs CEDIAG and DLE, no I/O diagnostics are callable by control card.

The automatic sequencer program (APR) has been removed from the SCOPE program library and added to the CE Diagnostics PL. Externally, APR is the same as it appeared on PL1. Internally, the code is optimized and tag names redefined to add lucidity to the listing. APR also is capable of processing a 17-bit sequencer table address.

Because of new system philosophy, a new ordering of job control cards is required to run a sequence job. See 6000 SCOPE 3.4 HARDWARE MAINTENANCE FEATURES REFERENCE MANUAL (Publication No. 60364800) for example.

CORRECTIONS

All eligible PSR Code published through PSR Summary No. 312 has been added to the CE Diagnostic program library.

INSTALLING CE DIAGNOSTIC PROGRAMS

Installation Parameters

Release values of installation parameters in the deadstart diagnostic sequencer routine, CES, are shown below. To change these parameters, cards with the proper code and the CEQU macro should be placed after an *INSERT CES.54 card and inserted into installation deck CEDIAG1.

IP.DFL CEQU 200B

Central memory field length, divided by 100B, used by the deadstart diagnostics. This parameter is used by the Y-option testing only.

IP.DRA CEQU 400B

Central memory RA/100B, used by the deadstart diagnostics. This parameter is used by the Y-option testing only.

IP.MCPU CEQU 1

Number of central processors present in the system; should be set to 1 for 6400 and 6600, to 2 for 6500 and 6700.

IP.NOISE CEQU 3

Same as IP.NOISE in CIOCOM of the SCOPE program library. IP.NOISE must have identical settings on the SCOPE and CE Diagnostic program libraries.

INSTALLATION PROCEDURES

The structure of the CE Diagnostic program library is: file 1, program library; file 2, binary form of running system diagnostics; file 3, binary form of diagnostic deadstart records; file 4, relocatable binary of CEFAP and EC2; and files 5 and 6 installation decks. To obtain the decks, perform the job:

```
Job card.
REQUEST(PL5,E)
REWIND(PL5)
SKIPF(PL5,4,17)              SKIP PL AND BINARY
COPYBF(PL5,PUNCH,2)          INSTALLATION DECKS
UNLOAD(PL5)
6/7/8/9
```

Installation job CEDIAG1 updates and creates a new CE diagnostics program library including assembled/compiled binaries of both the deadstart and running system diagnostics as files 2, 3 and 4. Job CEDIAG2 creates a deadstart tape of the running system including deadstart and running system CE diagnostic routines.

When a deadstart tape is configured for one system, location SITENUM in subroutine CEFAPC should be changed to the CE site number of that particular system.

Example:

```
*ID MYSITE
*D PTR8081.285            (Current sequence number of SITENUM)
 SITENUM    DIS     ,*C18J2.3*
*COMPILE CEFAP
```

When a deadstart tape is configured for multiple systems and the CE site number is to change for each system, location SITENUM should be changed to DATA 0; and CEFAP will request the site number through the L display.

Example:

```
*ID MYSITE
*D PTR8081.285            (Current sequence number of SITFNUM)
 SITENUM    DATA    0
*COMPILE CEFAP
```

Because of new system philosophy, a new ordering of job control cards is required to run a sequence job. See 6000 SCOPE 3.4 HARDWARE MAINTENANCE FEATURES REFERENCE MANUAL (Publication No. 60364800) for example.

```
CEDIAG1,CM62000,T7000,MT02.
COMMENT. THIS JCB UPDATES AND CREATES THE BINARY OF C.E.DIAGNOSTICS.
COMMENT. THE NEW PL5 WILL CONSIST OF FOUR FILES
COMMENT. THE FIRST FILE WILL BE THE NEWPL
COMMENT. THE SECOND FILE WILL BE THE ABSOLUTE AND RELOCATABLE BINARIES
COMMENT. THE THIRD FILE WILL BE THE BINARY OF THE DEADSTART ROUTINES
COMMENT. THE FOURTH FILE WILL BE THE RELOCATABLE BINARY OF CEFAP AND EC2
LAEEL(PL5IN,R,L=CEDIAG*3P4,D=HI)   CE DIAG OLDPL
REQUEST,PL5,N,HI.             SCRATCH FOR NEW PL5
LAEEL(PL5,W,L=CEDIAG*3P4,D=HI)
REWIND(PL5IN,PL5)
UPDATE(F,P=PL5IN,N=PL5)
UNLOAD(PL5IN)
COMPASS(I=COMPILE,S=SCPTEXT,L=0,B=BIN1)
COMPASS(I=CCMFILE,S=SCPTEXT,L=0,B=EIN1)
COMPASS(I=CCMPILE,S=SCPTEXT,S=SCHTEXT,L=0)
FTN(I=COMPILE,S=CPCTEXT,S=PFMTEXT,B=BIN,L=0)
REWIND(BIN)
LOAD(EIN)      GENERATION OF OVERLAYS CEFAP AND EC2
NOGO.
REWIND(CEFAF,EC2)
REWIND(BIN)
COFYBF(CEFAF,LGC)
BKSP(LGO,1)
COFYEF(EC2,LGO)
REWIND(BIN1)
COFYER(BIN1,NIL,5)
BKSP(LGO,1)
COFYBF(BIN1,LGO)
REWIND(EIN1,LGO)
SKIPF(FL5,1,17)
COFYEF(LGO,PL5)
COFYBF(BIN1,PL5)
COFYBF(BIN,PL5)
UNLOAC(PL5)
7/8/9  END CF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  ENC OF FILE
```

```
CEDIAG2,CM55000,T7000,MT2.
COMMENT. THIS JOB TAKES THE BINARIES FROM THE C. E. DIAGNOSTICS PL AND
COMMENT. EDITLIBS THEM INTO THE RUNNING SYSTEM. THE SECOND EDITLIB WILL
COMMENT. CREATE A NEW DEADSTART TAPE WHICH WILL CONTAIN THE DEADSTART
COMMENT. DIAGNOSTICS ROUTINES.
COMMENT.
COMMENT. IF C. E. DIAGNOSTICS HAS NOT BEEN INSTALLED IN THE SYSTEM,
COMMENT. REMOVE THE CARD...SKIPF(12,SYSTEM)...FROM THE SECOND EDITLIB.
LABEL(PL5,R,L=CEDIAG*3P4,D=HI)   MOUNT C. E. DIAGNOSTICS PL
REQUEST,NEWSYS,HY.          MOUNT SCRATCH TAPE FOR NEW DEADSTART
REWIND,PL5,NEWSYS.
SKIPF(PL5,1,17)             SKIP OLDPL
COPYBF(PL5,CEDIAG)          PP AND CP ROUTINES
COPYBF(PL5,DSR)             DEADSTART ROUTINES
REWIND,PL5,CEDIAG,DSR.
EDITLIB(SYSTEM)
EDITLIB(SYSTEM)
UNLOAD(PL5,NEWSYS)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
REPLACE(MTT+CR1,CEDIAG)
REPLACE(APR,CEDIAG,AL=1)
REPLACE(IEP+DLE,CEDIAG)
REWIND(CEDIAG)
LIBRARY(NUCLEUS,OLD)
REPLACE(CEDIAG+CERMS,CEDIAG,AL=7777,FL=2000,FLO=1)
REPLACE(CEFAP,CEDIAG,AL=7777,FL=40000,FLO=0)
REPLACE(EC2,CEDIAG,AL=7777,FL=45000,FLO=0)
REPLACE(CM6+MY1,CEDIAG,AL=7777,FL=20000,FLO=0)
REPLACE(ALS,CEDIAG,AL=7777,FL=1500,FLO=0)
REPLACE(FST,CEDIAG,AL=7777,FL=1200,FLO=0)
REPLACE(CT3,CEDIAG,AL=7777,FL=5000,FLO=0)
REPLACE(CU1,CEDIAG,AL=7777,FL=7000,FLO=0)
REPLACE(ALX,CEDIAG,AL=7777,FL=1500,FLO=0)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
READY(NEWSYS)
REWIND(SYSTEM)
TRANSFER(CEA,SYSTEM)
SKIPF(12,SYSTEM)          PULL OUT IF C. E. DIAG NOT IN SYSTEM
TRANSFER(CES+ALX,DSR)
TRANSFER(*,SYSTEM)
SKIPF(2,SYSTEM)
INCLUDEF(SYSTEM)
INCLUDE(NUCLEUS,SYSTEM,CM)
INCLUDE(SYSOVL,SYSTEM,DS)
INCLUDE(FORTRAN,SYSTEM,DS)
INCLUDE(COBOL,SYSTEM,DS)
INCLUDE(RUN2P3,SYSTEM,DS)
INCLUDE(SYSIO,SYSTEM,DS)
INCLUDE(SYSMISC,SYSTEM,DS)
INCLUDE(IGS274,SYSTEM,DS)
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

RELEASE DESCRIPTION

Maintenance tools for SCOPE 3.4 are provided on program library tape PL6. These maintenance tools are divided into three categories: SYMPL compiler, Version 3.0 object libraries, and conversion aids.

The structure of the release format PL6 is as follows:

Files 1-4	SYMPL, as follows
1	SYMPL source in UPDATE program library format
2	SYMPL compiler in relocatable binary
3	SYMPL compiler in absolute overlay binary
4	SYMPL object library in relocatable binary
Files 5-10	Version 3.0 object library source and binary
5	V3 FTN Object Library PL
6	V3 FTN Object Library Binary
7	V3 COBOL Object Library PL
8	V3 COBOL Object Library Binary
9	V3 SORT Object Library PL
10	V3 SORT Object Library Binary
Files 11-12	Conversion Aids
11	Conversion Aids source in UPDATE program library format
12	Conversion Aids binary
Files 13-22	Installation Decks
13,14	SYMPL installation decks
15,16	V3.0 FTN Object Library installation decks
17,18	V3.0 COBOL Object Library installation decks
19,20	V3.0 SORT Object Library installation decks
21,22	Conversion Aids installation decks

A. SYMPL

SYMPL (Systems Programming Language) is designed to facilitate systems programming; it does not contain some features normally found in higher level languages, such as complex arithmetic and input/output capability. Instead, it contains features particularly suited to systems programming -- bit manipulations, based arrays, and an elementary macro capability. It produces code optimized for efficient register and functional unit usage, particularly oriented toward the 6600 computer.

GENERAL DESCRIPTION

The SYMPL compiler is written mainly in SYMPL; only the system interface routines are in COMPASS. Thus an absolute binary of SYMPL is necessary for installation if changes are to be made to the source.

INSTALLATION PARAMETERS

SYMPL has no installation parameters.

INSTALLATION PROCEDURES

Before SYMPL can be installed, SCOPE, COMPASS, Record Manager and the FORTRAN Extended object library must have been installed previously. SYMPL can be updated and installed with the following jobs. Job SYMPL1 updates the SYMPL library tape. Job SYMPL2 edits SYMPL into the system from a SYMPL program library tape. To obtain the installation decks for SYMPL, perform a job of the type:

```
Job card.
REQUEST PL6,E.  Mount Maintenance Tools Tape
REWIND PL6.
SKIPF(PL6,12,17)
COPYBF(PL6,PUNCH,2)
UNLOAD PL6.
6/7/8/9
```

NOTE: The T7000 parameter on the SYMPL1 job card must be changed to T20000 if the deck is to be run on a 6200.

```
SYMPL1,CM100000,T7000,MT2.
COMMENT. THIS JOB UPDATE AND ASSEMBLES SYMPL.
COMMENT. THE NEW PL6 WILL CONTAIN 12 FILES.
COMMENT. FILE 1---SYMPL OLDPL
COMMENT. FILE 2---RELOCATABLE BINARY FOR COMPILER ROUTINES
COMMENT. FILE 3---ABSOLUTE BINARY OF COMPILER
COMMENT. FILE 4---RELOCATABLE BINARIES OF OBJECT ROUTINES
COMMENT. FILES 5 THRU 12 ARE THE VERSION 3.0 OLDPL≠S AND BINARIES OF
COMMENT. THE FTN, COBOL, SORT LIBRARIES AND THE CONVERSION AIDS.
COMMENT. THIS JOB UPDATE AND ASSEMBLES SYMPL.
LABEL,PL6IN,R,L=MAINTTOOLS*3P4,D=HI)
REQUEST,PL6,N,HI.
LABEL(PL6,W,L=MAINTTOOLS*3P4,D=HI)
REWIND,PL6IN,PL6.
COPYBF(PL6IN,OLDPL)
COPYBF(PL6IN,RELOC)
SKIPF(PL6IN,1,17)
COPYBF(PL6IN,OBJ)
REWIND(OLDPL,RELOC,OBJ)
UPDATE(W,N)
SYMPL(I=COMPILE,B=SRR,L=0)
COMPASS(I=COMPILE,B=SRR,S=IOTEXT,S=IPTEXT,L=0)
COMPASS(I=COMPILE,B=SRR,L=0)
REWIND,SSR.
COPYL(RELOC,SRR,SLGO)
LOAD(SLGO)
NOGO.
REWIND,SYMPL,SLGO.
REWIND(SRR)
```

```
COPYL(OBJ,SRR,NOBJ)
COPYBF(NEWPL,PL6)
COPYBF(SLOC,PL6)
COPYBF(SYMPL,PL6)
REWIND(NOBJ,PL6IN)
COPYBF(NOBJ,PL6)
SKIPF(PL6IN,4,17)
COPYBF(PL6IN,PL6,8)
UNLOAD(PL6IN,PL6)
7/8/9  END OF RECORD
*C CRFLST,ENDSYM
*C INIT52,ENDCOMP
*C SYMIO,ENDOBJ
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
SYMPL2,CM55000,T700,MT1.
COMMENT. THIS JOB EDITLIBS THE SYMPL BINARIES INTO THE RUNNING SYSTEM
COMMENT. FROM THE PL6 TAPE. EITHER THE RELEASED VERSION OF PL6 OR THE
COMMENT. VERSION CREATED BY THE DECK SYMPL1 MAY BE USED.
LABEL(PL6,R,L=MAINTTOOLS*3P4,D=HI)  MOUNT PL6
REWIND,PL6.
SKIPF(PL6,2,17)
COPYBF(PL6,SYMP)              SYMPL COMPILER BINARIES
BKSP(SYMP,1)
COPYBF(PL6,SYMP)              SYMPL OBJECT TIME BINARIES
REWIND,PL6,SYMP.
UNLOAD(PL6)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(SYMPL,SYMP,AL=3,FL=52000,FLO=1)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(SYMP10+SYMP52,SYMP)
FINISH.
LIBRARY(SYSIO,OLD)
REPLACE(*,SYMP)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

The best verification of successful installation of SYMPL is satisfactory compilation of FORM, QU, and/or QUIDDL.

B. VERSION 3.0 OBJECT LIBRARIES

The Maintenance Tools tape contains the object libraries for FORTRAN Extended 3.0, COBOL 3.0, and SORT/MERGE 3.0 for use in conversion of SCOPE 3.3 jobs to SCOPE 3.4. These three object libraries are to be used to satisfy external symbols from binary decks created by Version 3 of FORTRAN Extended, COBOL, or COMPASS programs containing calls to SORT MERGE 3.0. The Version 3 libraries do not use Record Manager for input/output; they retain calls to CIO. These routines are provided solely to allow usage of existing binary decks during the 3.3 to 3.4 transition phase. All three of these object libraries contain eligible PSR code as published through PSR Summary 312.

INSTALLATION PROCEDURES

Separate decks are provided to update and assemble each of the program libraries. Separate EDITLIB decks are provided to add the binary provided, or that created by the assembly deck, to the running system. To obtain these decks perform a job of the type

```
Job card.
REQUEST PL6,E.          Mount Maintenance Tools Tape
REWIND PL6.
SKIPF(PL6,14,17)
COPYBF(PL6,PUNCH,2)     PUNCH  V3FTN1,3VFTN2 DECKS
COPYBF(PL6,PUNCH,2)     PUNCH  V3COB1,V3COB2 DECKS
COPYBF(PL6,PUNCH,2)     PUNCH  V3SRT1,V3SRT2 DECKS
UNLOAD PL6.
6/7/8/9
```

Note that the deck V3FTN1 will run correctly only under SCOPE 3.3.

Once the V3FTN2, V3COB2 and/or V3SRT2 EDITLIB operations have completed, job SCOPE3 (see section 1) should be run to create a deadstart tape of the running system.

Use of the 3.3 object time routines under SCOPE 3.4 requires an LDSET(LIB=SYSMISC/SYSIO) control card prior to the EXECUTE control card.

```
V3FTN1,CM60000,T7000,MT2.                                              001
COMMENT. T H I S   D E C K   C A N   O N L Y   R U N   O N   3 . 3   A N D   002
COMMENT. W I L L   P R O D U C E   I N C O R R E C T   R E S U L T S   003
COMMENT. I F   R U N   O N   3 . 4 .                                   004
COMMENT. THIS JOB UPDATES AND ASSEMBLES THE FTN EXTENDED V3.0 OBJECT   005
COMMENT. LIBRARY ROUTINES AND CREATES A NEW PL6A.                      006
COMMENT. THE NEW PL6A WILL CONTAIN 12 FILES                            007
COMMENT. FILES 1 THRU 4---THE SYMPL OLDPL AND BINARIES.                008
COMMENT. FILES 5 AND 6---V3.0 FTN OLDPL AND BINARIES.                  009
COMMENT. FILES 7 THRU 12---V3.0 COBOL, SORT AND CONVERSION AIDS OLDPL≠S 010
COMMENT. AND BINARIES.                                                 011
LABEL(PL6AIN,R,L=MAINTTOOLS*3P4,D=HI)                                  012
REQUEST,PL6A,N,HI.                                                     013
LABEL(PL6A,W,L=MAINTTOOLS*3P4,D=HI)                                    014
REWIND,PL6AIN,PL6A.                                                    015
COPYBF(PL6AIN,PL6A,4)                                                  016
UPDATE(F,P=PL6AIN,N=PL6A,R=C)                                          017
COPYBF(ECF,PL6A)                                                       018
SKIPF(PL6AIN,2,17)                                                     019
FTN(SYSEDIT=IDENT,I=COMPILE,S=IPTEXT,L=0)                              020
REWIND,LGO.                                                            021
COPYBF(LGO,PL6A)                                                       022
COPYBF(PL6AIN,PL6A,6)                                                  023
UNLOAD(PL6AIN,PL6A)                                                    024
COMMENT.   *** END OF JOB ***                                          025
7/8/9  END OF RECORD
*/ ADD CORRECTIONS FOR V3.0 FTN OBJECT LIBRARY ROUTINES ONLY           026
7/8/9  END OF RECORD
6/7/8/9  END OF FILE


V3FTN2,CM55000,T700,MT1.                                               001
COMMENT. THIS JOB EDITLIBS THE FORTRAN EXTENDED V3.0 OBJECT TIME       002
COMMENT. ROUTINES INTO THE RUNNING SYSTEM FROM THE PL6 TAPE. EITHER THE 003
COMMENT. RELEASED VERSION OF PL6 OR THE VERSION CREATED BY THE DECK    004
COMMENT. V3FTN1 MAY BE USED.                                           005
LABEL(PL6A,R,L=MAINTTOOLS*3P4,D=HI)  MOUNT PL6                         006
REWIND,PL6A.                                                           007
SKIPF(PL6A,5,17)                                                       008
COPYBF(PL6A,FORT)          FTN V3.0 BINARIES                           009
REWIND,PL6A,FORT.                                                      010
UNLOAD(PL6A)                                                           011
EDITLIB(SYSTEM)                                                        012
COMMENT.  *** END OF JOB ***                                           013
7/8/9  END OF RECORD
READY(SYSTEM)                                                          015
LIBRARY(SYSMISC,OLD)                                                   016
REPLACE(*,FORT)                                                        017
FINISH.                                                                018
COMPLETE.                                                              019
ENDRUN.                                                                020
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
V3COB1,CM60000,T7000,MT2.
COMMENT. THIS JOB UPDATES AND ASSEMBLES THE COBOL V3.0 OBJECT LIBRARY
COMMENT. ROUTINES AND CREATES A NEW PL6B.
COMMENT. THE NEW PL6B WILL CONTAIN 12 FILES
COMMENT. FILES 1 THRU 6---SYMPL AND V3.0 FTN OLDPL≠S AND BINARIES.
COMMENT. FILES 7 AND 8---V3.0 COBOL OLDPL AND BINARY.
COMMENT. FILES 9 THRU 12---V3.0 SORT AND CONVERSION AIDS OLDPL≠S AND
COMMENT. BINARIES.
LABEL(PL6BIN,R,L=MAINTTOOLS*3P4,D=HI)
REQUEST,PL6B,N,HI.
LABEL(PL6B,W,L=MAINTTOOLS*3P4,D=HI)
REWIND,PL6BIN,PL6B.
COPYBF(PL6BIN,PL6B,6)
UPDATE(F,P=PL6BIN,N=PL6B,R=C,X)
COPYBF(ECF,PL6B)
SKIPF(PL6BIN,2,17)
COMPASS(I,S=CPCTEXT,S=IPTEXT,L=0)
REWIND,LGO.
COPYBF(LGO,PL6B)
COPYBF(PL6BIN,PL6B,4)
UNLOAD(PL6BIN,PL6B)
COMMENT.   *** END OF JOB ***
7/8/9  END OF RECORD
*/ ADD CORRECTIONS FOR V3.0 COBOL OBJECT LIBRARY ROUTINES ONLY
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
V3COB2,CM55000,T700,MT1.
COMMENT. THIS JOB EDITLIBS THE COBOL V3.0 OBJECT TIME ROUTINES INTO THE
COMMENT. RUNNING SYSTEM FROM THE PL6 TAPE. EITHER THE RELEASED VERSION
COMMENT. OF PL6 OR THE VERSION CREATED BY DECK V3COB1 MAY BE USED.
LABEL(PL6B,R,L=MAINTTOOLS*3P4,D=HI)  MOUNT PL6
REWIND,PL6B.
SKIPF(PL6B,7,17)
COPYBF(PL6B,COB)            COBOL V3.0 BINARIES
REWIND,PL6B,COB.
UNLOAD(PL6B)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(SYSMISC,OLD)
REPLACE(*,COB)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
V3SRT,CM60000,T7000,MT2.
COMMENT. THIS JOB UPDATES AND ASSEMBLES THE SORT V3.0 OBJECT
COMMENT. LIBRARY ROUTINES AND CREATES A NEW PL6C.
COMMENT. THE NEW PL6C WILL CONTAIN 12 FILES
COMMENT. FILES 1 THRU 8---SYMPL, V3 FTN AND COBOL OLDPL≠S AND BINARIES.
COMMENT. FILES 9 AND 10---V3.0 SORT OLDPL AND BINARIES.
COMMENT. FILES 11 THRU 12---CONVERSION AIDS OLDPL≠S AND BINARIES.
LABEL(PL6CIN,R,L=MAINTTOOLS*3P4,D=HI)
REQUEST,PL6C,N,HI.
LABEL(PL6C,W,L=MAINTTOOLS*3P4,D=HI)
REWIND,PL6CIN,PL6C.
COPYBF(PL6CIN,PL6C,8)
UPDATE(F,P=PL6CIN,N=PL6C,R=C,X)
COPYBF(EOF,PL6C)
SKIPF(PL6CIN,2,17)
COMPASS(I,S=CPCTEXT,S=IPTEXT,L=0)
REWIND,LGO.
COPYBF(LGO,PL6C)
COPYBF(PL6CIN,PL6C,2)
UNLOAD(PL6CIN,PL6C)
COMMENT.   *** END OF JOB ***
7/8/9  END OF RECORD
*/ ADD CORRECTIONS FOR V3.0 SORT OBJECT LIBRARY ROUTINES ONLY
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
V3SRT2,CM55000,T700,MT1.
COMMENT. THIS JOB EDITLIBS THE SORT/MERGE V3.0 OBJECT TIME ROUTINES INTO
COMMENT. THE RUNNING SYSTEM FROM THE PL6 TAPE. EITHER THE RELEASED
COMMENT. VERSION OF PL6 OR THE VERSION CREATED BY DECK V3SRT1 MAY BE
COMMENT. USED.
LABEL(PL6C,R,L=MAINTTOOLS*3P4,D=HI)   MOUNT PL6
REWIND,PL6C.
SKIPF(PL6C,9,17)
COPYBF(PL6C,SORT)           SORT V3.0 BINARIES
REWIND,PL6C,SORT.
UNLOAD(PL6C)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(SYSMISC,OLD)
REPLACE(*,SORT)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

C. CONVERSION AIDS

The following conversion aid programs are provided on the Maintenance Tools tape:

SIFT	A program to convert RUN 2.3 programs to FTN 4.0 format.
BBTO6RM	A program to convert FORTRAN binary blocked files to files that can be processed by SCOPE 3.4 and 6RM.
RANCONV	A program to convert single level SCOPE name/number indexed files created by COBOL 3.0 or FORTRAN Extended 3.0 to indexed word-addressable files that can be processed by COBOL 4.0 or FORTRAN Extended 4.0.
SPY	Utility package used to monitor the P register of a CPU program and provide an histogram of elapsed time used in specific areas of code.
WHEN	Control card callable utility that skips over selected cards in the control card record.
DOCK	Utility for extracting IMS information from the SCOPE program library.
STIMULATOR	Utility package used to simulate live INTERCOM 4.1 low speed terminals.
CONVPF	Utility allowing conversion of permanent files between SCOPE 3.3 and 3.4 formats. CONVPF is also described in the SCOPE 3.3 to SCOPE 3.4 Conversion Aids PSB.

INSTALLATION PROCEDURES

Before the Conversion Aids can be installed, FORTRAN Extended must be in the running system. Job AIDS1 updates and compiles File 11 creating a complete revised PL6. Job AIDS2 adds binary from File 12 to the running system. Jobs AIDS1 and AIDS2 may be obtained by performing a job of the type:

```
Job card.
REQUEST PL6,E.  Mount Maintenance Tools Tape
REWIND PL6.
SKIPF(PL6,20,17)
COPYBF(PL6,PUNCH,2)
UNLOAD PL6.
6/7/8/9
```

Once AIDS2 has completed, job SCOPE3 (see section 1) may be run to create a deadstart tape of the running system.

```
AIDS1,CM73000,T7000,MT02.
COMMENT.  THIS JOB UPDATES AND ASSEMBLES THE 3.4 CONVERSION AIDS.
COMMENT.  THE NEW PL6D WILL CONSIST OF 12 FILES.
COMMENT.  FILES 1 THRU 4 CONTAIN SYMPL OLDPL AND BINARIES.
COMMENT.  FILES 5 THRU 6 CONTAIN V3FTN OBJECT OLDPL AND BINARY.
COMMENT.  FILES 7 THRU 8 CONTAIN V3COBOL OBJECT OLDPL AND BINARY
COMMENT.  FILES 9 THRU 10 CONTAIN V3SORT OBJECT OLDPL AND BINARY
COMMENT.  FILE 11 CONTAINS THE CONVERSION AIDS OLDPL
COMMENT.  FILE 12 CONTAINS THE BINARIES OF CONVERSION AIDS.
REQUEST(OLDPL,E,HY)     LATEST SCOPE PL
UPDATE(Q,N=RANPL)
UNLOAD(OLDPL)
LABEL(PL6DIN,R,L=MAINTTOOLS*3P4,D=HI)
REQUEST(PL6D,N,HI)        SCRATCH TAPE FOR NEW PL6D
LABEL(PL6D,W,L=MAINTTOOLS*3P4,D=HI)
REWIND(PL6DIN,PL6D)
COPYBF(PL6DIN,PL6D,10)
UPDATE(F,P=PL6DIN,N=PL6D,R=C)
COPYBF(ECF,PL6D)
UNLOAD(PL6DIN)
COMPASS(I=CCMFILE,S=SCPTEXT,L=0)
COMPASS(I=CCMFILE,S=SCPTEXT,L=0,B=AIDS)
FTN(I=COMPILE,S=IOTEXT,S=CPCTEXT,OPT=1,L=0,B=AIDS)
COPYBR(CCMPILE,NIL)       SKIPF SIFT DEBUGGING AID
RUN(S,,,COMPILE,NUL,AID)
COMPASS(I=CCMFILE,S=0,L=0,B=DOCTEXT)
COMPASS(I=CCMFILE,S=0,L=0,G=DOCTEXT)
REWIND(AIDS)
LDSET(PRESET=ZERO)
LOAD(AIDS)
NOGO.
REWIND(AID)
LOAD(AID)
NOGO.
REWIND(AIDES,LGO)
COPYBF(LGO,PL6D)
BKSP(PL6D,1)
COPYBF(AIDES,PL6D)
UNLOAD(PL6D)
7/8/9  END OF RECORD
```

```
*COMPILE,PFMAC
7/8/9  END OF RECORD
*/   ADD CORRECTIVE CODE HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE


AIDS2,CM55000,T7000,MT01.
COMMENT.  THIS JOB EDITLIBS THE BINARIES OF THE CONVERSION AIDS INTO
COMMENT.  THE RUNNING SYSTEM.  EITHER THE RELEASE VERSION OF PL6 OR THE
COMMENT.  VERSION CREATED BY DECK AIDS1  MAY BE USED.
LABEL(PL6D,R,L=MAINTTOOLS*3P4)      MOUNT PL6
REWIND(PL6D)
SKIPF(PL6D,11,17)
COPYBF(PL6D,AIDS)
UNLOAD(PL6D)
EDITLIB(SYSTEM)
COMMENT.   *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
*/   REMOVE THE FOLLOWING CARD IF STIMULATOR IS NOT TO BE INSTALLED
REPLACE(VSM+1VG,AIDS)
*/   REMOVE THE FOLLOWING CARD IF SPY IS NOT TO BE INSTALLED
REPLACE(SPY+SP2,AIDS)
LIBRARY(NUCLEUS,OLD)
*/   REMOVE THE FOLLOWING CARD IF STIMULATOR IS NOT TO BE INSTALLED
REPLACE(SIF,AIDS,AL=3,FL=20000,FLO=1)
*/   REMOVE THE FOLLOWING CARD IF SPY IS NOT TO BE INSTALLED
REPLACE(=CPSPY$=$,AIDS,AL=1)
*/   REMOVE THE FOLLOWING CARD IF DOCK IS NOT TO BE INSTALLED
REPLACE(DOCK,AIDS,AL=3,FL=16000,FLO=1)
*/   REMOVE THE FOLLOWING CARD IF WHEN IS NOT TO BE INSTALLED
REPLACE(WHEN,AIDS,AL=3,FL=1200,FLO=1)
*/   REMOVE THE FOLLOW CARD IF EBT06RM IS NOT TO BE INSTALLED
REPLACE(EBT06RM,AIDS,AL=3,FL=10000,FLO=1)
*/   REMOVE THE FOLLOWING CARD IF RANCONV IS NOT TO BE INSTALLED
REPLACE(RANCONV,AIDS,AL=3,FL=6000,FLO=1)
*/   REMOVE THE FOLLOWING CARD IF SPY IS NOT TO BE INSTALLED
REPLACE(PRNTSPY,AIDS,AL=1,FL=35000)
*/   REMOVE THE FOLLOWING CARD IF CONVPF IS NOT TO BE INSTALLED
REPLACE(CONVPF,AIDS,AL=7777,FL=30000,FLO=0)
*/   REMOVE THE FOLLOWING CARD IF SIFT IS NOT TO BE INSTALLED
REPLACE(SIFT,AIDS,AL=3,FL=30000,FLO=1)
*/   REMOVE THE FOLLOWING 3 CARDS IF STIMULATOR IS NOT TO BE INSTALLED
REPLACE(DATAR,AIDS,AL=3,FL=60000,FLO=1)
REPLACE(SORT,AIDS,AL=3,FL=57000,FLO=1)
REPLACE(LEE,AIDS,AL=3,FL=62000,FLO=1)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

D. USAGE OF PF CONVERSION AIDS

Usage instructions for SIFT may be found in SIFT (FORTRAN Translator Program PSB, publication 60358400).

Usage instructions for BBTO6RM, RANCONV, WHEN, and SPY, may be found in SCOPE V3.3 to SCOPE V3.4 Conversion Aids PSB, publication 60358200.

Usage instruction for the other conversion aids are as follows.

D.1 WHEN Use in a SCOPE 3.3 Environment

During conversion from SCOPE 3.3 to SCOPE 3.4, the compass source language utility WHEN allows control cards unique to one environment to be skipped in the other. As WHEN is not a part of the standard SCOPE 3.3 environment, the following job structure may be used for entering WHEN into a 3.3 system through EDITLIB.

```
JOB,CM55000,T100,MT01.
REQUEST OLDPL,E,HI.        MOUNT 3.4 CONVERSION AIDS PL
SKIPF(OLDPL,10,17)
COPYBF(OLDPL,PL)
UPDATE(Q,P=PL)
COMPASS(I=COMPILE,S=SCPTEXT,L=0)
REWIND LGO.
LOAD(LGO)
NOGO.
REWIND AIDES.
EDITLIB.
7/8/9
*C   WHEN
7/8/9
READY(SYSTEM)
RPADD(WHEN,AIDES)
COMPLETE.
6/7/8/9
```

D.2 DOCK USAGE

DOCK is a FORTRAN source language utility for extracting listable Internal Maintenance Specification information from a COMPILE file generated from the SCOPE 3.4 program library.

The control card directive is of the form;

```
DOCK(p1,p2,p3,...,pn)
```

Definition:

Default, if parameter is not specified.

Assumed, if parameter is specified, but not equivalenced.

I= Name of program source file. (Assumed to be an update COMPILF or source file not exceeding 90-column BCD characters.) Default = SOURCE, Assumed = COMPILE.

L= Name of file containing documentation list; (cannot be the same name as I). Default = Assumed = OUTPUT.

F= Up to 25 characters to be printed in the bottom left corner of each page of documentation.
Default = INT, Folio = $INTERNAL DOCUMENTATION.$
EXT, Folio = $EXTERNAL DOCUMENTATION.$
Assumed = $I M S.$

INT Internal; all internal and external documentation will be listed on file L.

EXT External; only external documentation will be listed on file L. Default = INT.

INDEX At the end of each routine processed an index is printed, all symbols found in location field of EJECT, SPACE, TITLE, and TTL cards. Default = INDEX off.

NR No rewind of input file. (I) Default = rewind of INPUT.

NT No table generation. Default = table generation.

NP No propogation of page numbers across routine. Default = on.

TE Documentation file, L, formatted for input into program TEXTJAB. Default = off.

Default Parameter Settings:

DOCK(I=SOURCE,L=OUTPUT,F=$INTERNAL DOCUMENTATION.$,INT,P)

Assumed Parameter Settings:

DOCK(I=COMPILE, L=OUTPUT,F=$INTERNAL DOCUMENTATION.$,INT,P)

The following dayfile messages are issued by DOCK:

FL TOO SHORT FOR DOCK. (REQUIRES 12K).

Not enough field length was allowed; current minimum field length is 10K (octal).

FILE NAME CONFLICT.

Input, I, and List, L, file names are the same.

MEMORY OVERFLOW IN BUILDING INDEX TABLE.

Not enough field length for index table; increase by 4K (octal).

EMPTY INPUT FILE. NO DOCUMENTATION PRODUCED.

Input file was empty.

INPUT FILE NAME IS ILLEGAL.
OUTPUT FILE NAME IS ILLEGAL.

Illegal character specified in file name.

FILE EQUIVALENCE MAY NOT BE 0.

A file parameter cannot be set to zero.

D.3 STIMULATOR OPERATING INSTRUCTIONS

D.3.1 Automatic Table Setting:

The STIMULATOR is revised to automatically set the EST and mux-subtables if the user so specifies. The new operating procedures are as follows:

Bring up the STIMULATOR at any NEXT control point, read in the SIP job.

After requesting appropriate tape assignments, the STIMULATOR asks:

DO YOU WANT AUTOMATIC TABLE SETTING -- N.YES OR N.NO

When the answer is YES, the STIMULATOR makes the following checks:

Is INTERCOM already up? If so, the following diagnostic is issued and the SIP job drops:

INTERCOM IS ALREADY UP--DROP INTERCOM TO CONTINUE

If INTERCOM is not already up, the STIMULATOR checks for a mux-subtable pointer in low core. If no pointer is found, the following diagnostic is issued and the SIP job drops.

ERROR--DONT HAVE MUX SUBTABLE DEFINED

If both checks are passed:

The STIMULATOR searches the EST until it finds either a YC or DC entry (turned OFF). When not enough YC or DC entries are found in the EST, the following is displayed:

NOT ENOUGH EXISTING EST YC OR DC ENTRIES--SET MANUALLY

EST YC entries will not be created because a valid mux-subtable pointer cannot be chosen arbitrarily.

When a valid YC or DC entry is found, the entry is changed to YC if necessary; and the channel number from the SIP card C parameter is used.

The mux-subtable pointer from this EST entry is used to modify the appropriate mux-subtable. The SIP card T parameter is placed in the mux-subtable to indicate the number of Teletypes.

The mux table header word also is set to indicate a STIMULATOR run.

When all tables are set properly, the following message is displayed:

--OFF EXTRA MUXS THEN BRING UP INTERCOM--

When INTERCOM is brought up, the following is displayed, and the STIMULATOR run is continued:

CONTINUE SIMULATION

The STIMULATOR changes only the first one or two EST entries encountered (depending on M parameter of SIP card).

If NO is the answer to DO YOU WANT AUTOMATIC TABLE SETTINGS the following is displayed:

YOU ARE RESPONSIBLE FOR TABLES SETUP--TYPE N.GO TO CONTINUE

Thus, the user can set tables manually if he wishes.

D.3.2 Manual Table Setting:

The EST must contain an entry for a 6676 multiplexer specifying the channel to be used for the stimulation. The entry should point to a valid multiplexer subtable. The entry can be typed manually after unlocking the keyboard. For example: if the entry is to be at 2532 in memory, type:

2532,0000	0004	0000	3103	0002.
	Channel #	Equip #	YC Mnemonic	Points to 1st entry in mux subtable

The mux header word must have byte 4 set to the value 1. The INTERCOM pointer word (word 16, byte 1) points to this word. For example, if word 16 points to 5640 type:

5640,4,1.

The mux table must be set to reflect the number of terminals (this value must agree with the SIP card T parameter). Type:

5642, 00xx. xx=number of terminals

D.3.3 Running Two Multiplexers

Case one: two 6676's simulated on the same channel

The M parameter on the SIP card should be set to 2. The T parameter should indicate the number of terminals on each mux. The STIMULATOR input tape must contain as many sessions as the total number of terminals: 2*T. Two EST YC entries and two mux subtables are required. (Assemble the INTERCOM driver with IP.N6676 EQU 2.)

Case two: One simulated 6676 and one actual 667X on another channel

It is possible to simulate one 6676 multiplexer and at the same time run another actual hardware 6671 or 6676 multiplexer on a different channel. In this case, the user must ensure that an EST entry and mux subtable also exist and properly define the live multiplexer and its terminals.

The SIP parameters are set in the same way as for one simulated multiplexer.

D.3.4 Bringing up INTERCOM and the STIMULATOR:

Please refer also to Automatic Table Setting instructions.

Bring up INTERCOM by typing: INTERCOML.

The STIMULATOR can be run at any available control point but it should be locked in to avoid attempts to roll it out. Type in:

n.LOCKIN.

Background batch can be run as desired. Remember, in SCOPE 3.4, any NEXT control point can be used to run both INTERCOM and batch jobs.

The STIMULATOR will drop automatically when INTERCOM is dropped. To drop INTERCOM type:

INTERCOML,DROP.

D.3.5 Description of STIMULATOR Parameters:

A maximum of two tapes is required for a simulation run: TAPEI and TAPEO. TAPEI is the input tape containing the Teletype programs to be simulated during the run; this tape must be assigned for all simulation runs. TAPEO is used for recording all system output resulting from the simulation; it is required if the user selects the option to recover system output. The STIMULATOR requests assignment of the appropriate tapes during its initialization phase.

The STIMULATOR is called with a Program Call control card of the form:

SIP(Mx,Txxx,Dxxx,Sxxxx,Lxx,Cxx,Ox,Fxxxx)

The parameters M,T,D,S,L,C,O,F are order independent, all values are octal:

Mx	Number of simulated multiplexers (maximum of 2)
Txxx	Number of Teletypes per multiplexer (1-100)
Dxxx	Number of Teletypes per multiplexer to activate dynamically every S cycles
Sxxxx	Time interval in octal cycles for activating D Teletypes (one cycle equals approximately 200 ms)
Lxx	0 implies each TTY is to simulate all input programs for current run. L=0 option requires INTERCOM modifications, as same user would LOGIN at more than one terminal. 1-77 indicates the number of times each TTY is to loop on its assigned program
Cxx	Channel number of simulated multiplexers
Ox	Zero implies recover system output, non-zero implies to bypass output
Fxxx	File number of TAPEI to be used for current SIMULATION

The following default values are assumed:

M = 1
T = 1
D = 0
S = 0
L = 1
O = 0
F = 1

If the channel parameter C is not specified, SIP will be terminated with an error messages. The only restriction on the S parameter:

$$\frac{T.S}{D} < 4096$$

The following sample deck will perform the data acquisition for a 20 terminal simulation on one multiplexer:

```
JOB1,T1000,MT02,CM20000.
SIP(M1,T24,D24,S1,L2,CO,O0,F1)
6/7/8/9
```

D.3.6 STIMULATOR Input Card Format

Test programs are stored on cards for input to INTERCOM via the STIMULATOR. Each card image represents one Teletype line of information. The first character must be punched in column 1 and the last character must be a v (11-0 punch).

If the input line will generate a line feed as the only response from INTERCOM, as with text editing under EDITOR, the character (0-8-7 punch) must precede the character v (11-0 punch) on the card. The STIMULATOR interprets the character for internal purposes only; it is sent to INTERCOM as a blank. The v character is transmitted to INTERCOM as a carriage return.

Each test program must begin with the LOGIN procedure and end with the LOGOUT system commands. The main body of the test program can contain any combination of system commands, source input, or data. Essentially, each test program represents a complete user session at a Teletype from LOGIN to LOGOUT. In converting programs from Teletype input to cards for STIMULATOR input, the differences in character sets must be considered. For example, the quotations character " for Teletypes is the equivalence character (=, 0-8-6 punch) on cards. For a more detailed description of display and TTY characters refer to INTERCOM section (14) of this document or the INTERCOM 4 Reference Manual (publication no. 60307100).

The card images can be copied to tape with the COPYBF utility. EOR's separate each test program.

Since the following characters have special meaning to the STIMULATOR, they should not be used as data in the input tape:

V indicates carriage return
Λ indicates EDITOR text editing line
[indicates control X
] indicates control Z

example:] A gives user abort

Example of input tape preparation:

```
JOB,MT01.
REQUEST,TAPE,HI.
COPYBF(INPUT,TAPE)
7/8/9
LOGIN.v
NAMEv      (user name)
PASSWDv    (user password)
EDITOR.v
FOR,Fv
10 PROGRAM Z(INPUT,OUTPUT) v
20 PRINT 10∧v
30 READ 20,A∧v                  all text editing commands require
     .                          ∧ character before v character
     .
     .
160 END∧v
RUN,FTNv
   2.0v
   3.0v
BYEv
LOGOUT.v
7/8/9
     (second test program)
7/8/9
    .
    .
    .
7/8/9
     (last test program, maximum of 64)
7/8/9
6/7/8/9
```

D.3.7. Data Reduction Phase:

The following sample deck will perform data reduction of the STIMULATOR output tape:

```
JOB2,CM55000,T1000,MT01.
REQUEST,TAPE1,HI.  STIMULATOR OUTPUT TAPE
REWIND(TAPE1)
DATAR.
6/7/8/9
```

DATAR will give the following output:

A raw output showing: number of Active Terminals, all response times, and a two-word debugging output for each response.

A histogram which slots all response times and calculates a mean and standard deviation. The cumulative probability column gives the probability that response time will be less than a given number of seconds. If cumulative probability is .4891 and the interval is 10.5-11.0, 48.9% of all response times were less than 11.0 seconds.

The following deck will give a more detailed report of the simulation:

```
JOB3,CM60000,T7777,MT2.
REQUEST,TAPEO,HI.   STIMULATOR OUTPUT TAPE
REWIND(TAPEO)
COPYBF(TAPEO,TAPE2)
SORT.
REQUEST,TAPE3,HI.   STIMULATOR INPUT TAPE
REWIND,TAPE1,TAPE3.
LEE(LC=77777)
7/8/9
6/7/8/9
```

LEE will give a detailed report of the activity for each terminal. For each command, the response received will be shown as well as the response time in seconds and milli-seconds.

D.3.8 Miscellaneous Information:

Normally, the STIMULATOR will shrink its field length to the minimum required. If it is necessary to simulate a machine with less CM than is actually available (e.g., 49K), the CM on the STIMULATOR Job card can be set accordingly and FL reduction can be prevented by the following modification:

```
*IDENT,FIXCORE
*DELETE,VSM.335
*DELETE,VSM.358
*COMPILE,VSM
```

The operator can examine the data captured by the STIMULATOR by displaying the output tape buffer and the TTY output pots. Suppose the STIMULATOR is running at JDT 20:

Type: C = 20

Cycle display until the FET for TAPEO is found (about 350) FIRST will be about 2040, type: C4,2040. for beginning display of TAPEO buffer.

The display can be cycled until OUTPUT pots are found. The OUTPUT pots are two 8-word buffers per TTY that give an up-to-date account of the data being received by each terminal. They are located near 3000 relative to RA.

D.3.9 Type time: the release version of the STIMULATOR has a built in type time of 0 seconds. To change the type time, the following modification should be made to 1VG:

```
*IDENT,TYPET
*DELETE,THHH.1
     ADN    x  (x is the type time in octal seconds; must be ≤77)
*COMPILE,1VG
```

D.3.10 Changes to INTERCOM:

At times, changes should be made to INTERCOM depending on the type of simulation.

When more than 30 terminals are running, the EDITOR buffers should be increased appropriately. Refer to the INTERCOM V4.1 section of this document for description of EDITOR parameters.

The appropriate version of the INTERCOM driver must be available. The TYPEJ variant with IP.N6671 EQU 0 and IP.N6676 EQU 1 is required for a one multiplexer simulation. For other variations, the appropriate version of the driver msut be assembled. Refer to the INTERCOM V4.1 section for more details.

D.3.11 Hardware Resources needed by the STIMULATOR:

If the output is being saved on tape the STIMULATOR requires two dedicated PP's; but only one is required if output is not saved.

A free data channel with no equipment on it is required for communication between the STIMULATOR and the INTERCOM drivers. Equipment 0 is used for the first multiplexer and 1 for the second.

The STIMULATOR uses one control point; the field length depends on the length of the input tape and the number of simulated terminals. (Approximately 15K (octal) for a typical run consisting of one mux and an input tape consisting of 64 sessions, 30 lines per session.)

E. CONVPF UTILITY

The CONVPF utility allows the conversion of SCOPE permanent files between SCOPE 3.3 and SCOPE 3.4:

When the conversion is from SCOPE 3.3 to SCOPE 3.4 CONVPF execution is the last operation performed under the running SCOPE 3.3 system before a system load of SCOPE 3.4.

When the conversion is from SCOPE 3.4 to SCOPE 3.3, CONVPF execution is the last operation performed under the running SCOPE 3.4 system before a system load of SCOPE 3.3.

Procedures for using the CONVPF utility: Ensure that RBR descriptions of the PFD and PF devices are identical in both SCOPE 3.3 and SCOPE 3.4 system; correspondence of RBR ordinals for these devices must be the same in number and name. The best procedure is to construct the RBR table in the respective CMRs identically; however, the objective can be achieved by careful reconfiguration at deadstart time. If the RBR correspondence is not assured, several error diagnostics from deadstart could be displayed, such as, RB CONFLICT.

CONVPF requires a comparable interface relationship in both the SCOPE 3.3 and SCOPE 3.4 environments. The following PSRs must be installed to ensure that CONVPF will function as intended in both environments. Ensure the following PSRs are installed:

SCOPE 3.3:

SC31474 (PSR summary 322) 6PC only removes FNT for FNTs with APF ordinal of 7777B. Routines affected: EPF, 6PC.

SC31485 (PSR summary 324) Stack processor does not read ahead if the exact bit is set on a PP read. Routines affectdd: 1SP, PFC, PFA, PFR, LPF.

SC31486 (PSR summary 326) IRCP does not recognize RBT word pairs not being split across PRUs of RBTC. Routine affected: IRCP.

SCOPE 3.4:

SC40036 (PSR summary 324) IRCP doubles the values of the EOI PRU in the RBT chain of the PFD when permanent files are recovered after a TRANSPF of the permanent file tables. Routine affected: IRCP.

SC40039 (PSR summary 324) IOP modified to not issue INDEX BUFFER NOT SPECIFIED message to the B-display. Routine affected: IOP.

SC40040 (PSR summary 324) PFD is modified to support CONVPF. Routine affected: PFD.

Execution of the CONVPF utility -SCOPE 3.3

Step 1. EDITLIB routines CONVPF, PFD

CONVPF, a FORTRAN Program and its partner PP program, PFD, will not be a part of the standard SCOPE 3.3 system. Therefore, EDITLIB should be used. The following job setup, to be run under 3.3, is suggested:

```
Job card.
REQUEST(OLDPL,HI,E)            MOUNT SCOPE 3.4 CONVERSION AIDS PL
SKIPF(OLDPL,10,17)
COPYBF(OLDPL,PL)
UNLOAD(OLDPL)
UPDATE(Q,P=PL)
FTN(I=COMPILE,S=SCPTEXT,OPT=1,L=0)      COMPILE CONVPF
REWIND(LGO)
LOAD(LGO)
NOGO.
REWIND(AIDES)
REQUEST(OLDPL,HY,E)            REQUEST SCOPE 3.4 PL
UPDATE(Q,X)
COMPASS(I=COMPILE,S=SCPTEXT,L=0,B=BIN)
REWIND(BIN)
EDITLIB.
7/8/9
*COMPILE CONVPF
7/8/9
*COMPILE PFD
7/8/9
READY(SYSTEM)
RPADD(CONVPF,AIDES,DS)
RPADD(PFD,BIN,DS)
COMPLETE.
6/7/8/9
```

Step 2. Idle running system.

Step 3. Type in n.X CONVPF.

Step 4. Communicate with CONVPF utility.
Dialogue:

CONVPF: NUMBER (OCTAL) OF SUBDIRECTORIES IN 3.4 - GO OR CFO
OPERATOR: n.GO.
A GO. response will result in the same number of subdirectories as SCOPE 3.3 is using. (Note: In SCOPE 3.3, the number of subdirectories in CMR includes subdirectory zero; however, in SCOPE 3.4, subdirectory zero does not exist. Therefore, on a GO. response, the SCOPE 3.3 CMR value for the number of subdirectories will be decremented by 1 and used as the number of subdirectories for the target SCOPE 3.4 system.) Or,
OPERATOR: n.CFO m.
The value m should be the value of N.SD in the SCOPE 3.4 CMR that will be used.

CONVPF: mB SUBDIRECTORIES - GO OR DROP
CONVPF will echo the value that will be used.
OPERATOR: n.GO.
The operator believes CONVPF to have the correct value. Or,
OPERATOR: n.DROP.
The operator believes his response was incorrect.

CONVPF: NUMBER (OCTAL) OF ENTRIES PER SUBDIRECTORY IN SCOPE 3.4 - GO or CFO
OPERATOR: n.GO.
The SCOPE 3.3 CMR value for N.ESD will be used in the construction of the SCOPE 3.4 PFD. Or,
OPERATOR: n.CFO m.
The value m should be the value of N.ESD in the SCOPE 3.4 CMR that will be used.

CONVPF: mB ENTRIES PER SUBDIRECTORY - GO OR DROP
OPERATOR: n.GO.
Correct value. Or,
OPERATOR: n.DROP.

CONVPF: NUMBER (OCTAL) OF PRUS/20B IN 3.4 RBTC-GO OR CFO
OPERATOR: n.GO.
The SCOPE 3.3 CMR value for N.RBTC will be used for the SCOPE 3.4 RBTC. Or,
OPERATOR: n.CFO m.
The value m will be used for the value of N.RBTC in SCOPE 3.4. This is the only value which need not be identical to the respective value specified in the SCOPE 3.4 CMR.

CONVPF: mB PRUS/20B IN RBTC-GO OR DROP
OPERATOR: n.GO. Or,
OPERATOR: n.DROP.

```
CONVPF:   CONVPF FIELD LENGTH (OCTAL THOUSANDS), DEFAULT IS 50B
          -GO OR CFO
OPERATOR:      n.GO.
     50000B will be used as field length.  Or,
OPERATOR:      n.CFO  m.
     m thousand will be the field length used; the larger the field
     length that is used the faster the routine will execute.

CONVPF:   m 000B FIELD LENGTH -GO OR DROP
OPERATOR:      n.GO.          Or,
OPERATOR:      n.DROP.

CONVPF:   PFDUMP OR CONVPF REQUEST - GO OR DROP
OPERATOR:      n.GO.          Or,
OPERATOR:      n.DROP.
     At this point CONVPF is requesting permission to read PFD.

CONVPF:   PFDUMP OR CONVPF REQUEST - GO OR DROP
OPERATOR:      n.GO.          Or,
OPERATOR:      n.DROP.
     At this point CONVPF is requesting permission to read RBTC.

CONVPF:   PFDUMP OR CONVPF REQUEST - GO OR DROP
OPERATOR:      n.GO.          Or,
OPERATOR:      n.DROP.
     At this point CONVPF is requesting permission to read PFD device
     label.

CONVPF:   REWRITING PFD DEVICE LABEL - GO OR DROP
OPERATOR:      n.GO.          Or,
OPERATOR:      n.DROP.
     At this point CONVPF is requesting permission to rewrite PFD
     device label.  A GO. response will prohibit any further
     permanent file activity under SCOPE 3.3.

CONVPF:   DEADSTART (NORMAL) 3.4 1.L, 2.A, 3.I, 4.C
     This message informs operator of how he is to do a normal
     deadstart of SCOPE 3.4.
```

WARNING:

CONVPF will not stop flashing a question until a valid response is entered.

CONVPF can be called only through DIS or by the operator.

CONVPF can be dropped and restarted anytime before the the label is rewritten.

Execution of the CONVPF utility - SCOPE 3.4

Step 1. Idle running system.

Step 2. Type in n.X CONVPF.

Step 3. Communicate with CONVPF utility as in SCOPE 3.3.

EXCEPTIONS: A .GO. response should be made to the question of how many subdirectories in SCOPE 3.3 only if the CMR value of N.SD in 3.3 is one greater than the CMR value of N.SD in SCOPE 3.4.

The last message by CONVPF is the following:

DEADSTART (NORMAL) 1.N, 2.A, 3.Y

SCOPE 3.4 CONVPF messages in order of occurence:

1. NUMBER (OCTAL) OF SUBDIRECTORIES IN 3.4 - GO OR CFO
2. mB SUBDIRECTORIES - GO OR DROP
3. NUMBER (OCTAL) OF ENTRIES PER SUBDIRECTORY IN SCOPE 3.4 - GO OR CFO
4. mB ENTRIES PER SUBDIRECTORY - GO OR DROP
5. NUMBER (OCTAL) OF PRUS/20B IN 3.4 RBTC - GO OR CFO
6. mB PRUS/20B IN RBTC - GO OR DROP
7. CONVPF FIELD LENGTH (OCTAL THOUSANDS), DEFAULT IS 50B - GO OR CFO
8. m000 FIELD LENGTH - GO OR DROP
9. PFDUMP OR CONVPF REQUEST - GO OR DROP
10. PFDUMP OR CONVPF REQUEST - GO OR DROP
11. PFDUMP OR CONVPF REQUEST - GO OR DROP
12. REWRITING PFD DEVICE LABEL - GO OR DROP
13. DEADSTART (NORMAL) 3.3 1.N, 2.A, 3.Y

WARNING:

Permanent file names are qualified by the ID in SCOPE 3.4 but not in SCOPE 3.3; therefore, duplicate permanent file names could exist after a SCOPE 3.4 to SCOPE 3.3 Permanent file conversion. Some files could become inaccessible under SCOPE 3.3.

Cycle numbers can be no larger than 999D in SCOPE 3.4; however, SCOPE 3.3 only supports cycle numbers less than 64. Therefore, Permanent files with cycle numbers greater than 63 sould not be used in SCOPE 3.3.

Archived permanent files are not supported under SCOPE 3.3; therefore, under SCOPE 3.3 converted SCOPE 3.4 archived files will appear as vacuous files.

A conversion of SCOPE 3.4 permanent files to SCOPE 3.3 permanent file will result in the format of the permanent file tables introduced by PSR SC31480 published in PSR summary 324. This format is compatible with a SCOPE 3.3 system without SC31480 installed.

The first conversion of SCOPE 3.4 to SCOPE 3.3 always will increase the size of RBTC entries as compared to the size of SCOPE 3.3 RBTC entries that were not created as a result of a PF conversion. The amount of space used in the RBTC table may or may not increase over the original SCOPE 3.3 system.

If the Permanent file ID hashing scheme has been modified in SCOPE 3.4, the routine HSH must also be modified appropriately.

During SCOPE 3.3 normal deadstart process after a successful CONVPF execution on SCOPE 3.4, the following message may be issued to the display for the non-PFD devices:

CHECKSUM ERROR ON LABEL - RB nnnn

The correct operator response is NEW.

CONVPF ERROR MESSAGES AND RECOMMENDED OPERATOR ACTION

INSUFFICIENT FL AT 1031	Rerun CONVPF with larger field length specified.
WRITE ERROR AT nnnn where nnnn is statement number	Dump CONVPF field length and call system analyst. Dump high and low core of CM also.
READ ERROR AT nnnn where nnnn is statement number	Same as above.
PFD DEVICE OVERFLOW AT 143	Release disk space on PFD device and rerun CONVPF i.e. do DUMPF(MO=2) on PFD device.
END-OF-RECORD IN SUBDIRECTORY FILE	Dump high and low core in CM. Dump CONVPF field length and call system analyst.
PARTIAL ENTRY IN SUBDIRECTORY	Same as above.
PFD OVERFLOW AT 11 TP33 PFD OVERFLOW AT 11 TP34	PFD specification for target CMR is too small. Consult system analyst.
BAD 1ST RBT ORD AT 51 TP33 BAD 1ST RBT ORD AT 41 TP34	Same as above.
BAD RBT ORD AT 55 TP33 BAD RBT ORD AT 42 TP34	Same as above.
RBTC OVERFLOW AT 59 TP33 RBTC OVERFLOW AT 50 TP34 RBTC OVERFLOW AT 108 TR34 RBTC OVERFLOW AT 51 WRA	Rerun CONVPF with larger RBTC size specified.

RELEASE DESCRIPTION

FORTRAN Extended Version 4.0 provides many new capabilities. It operates under the SCOPE 3.4 operating system and requires the same minimum hardware configuration as SCOPE.

RELEASE MATERIALS

FORTRAN Extended 4.0 is released on two reels of tape: PL7 contains the FORTRAN Extended 4.0 compiler; PL8 contains the FTN 4.0 object library. This object library is used also by 7000 SCOPE 2.0, FORTRAN Extended 2.0, and RUN FORTRAN 2.0 programs.

MODIFICATIONS

Major improvements include the following:

1. IMPLICIT Statement

   IMPLICIT type1(range1),type2(range2),...,typen(rangen)

type	LOGICAL, INTEGER, REAL, COMPLEX, DOUBLE PRECISION
range	r1,r2,...rm
ri	letter or letter-letter

2. LEVEL Statement

   LEVEL n, list

n	6000	7000
1	Central Memory	SCM
2	Central Memory	LCM (Direct Access)
3	ECS (Block Copy)	LCM (Block Copy)
List	List of variable/array names separated by commas	

3. Quote Delimited Strings

   Version 4.0 extends the usage of the quote delimited string to any context where a Hollerith expression is allowed.

4. Expressions in Output Lists

Items which may occur in an output list have been expanded to permit Hollerith strings as well as any expression.

```
PRINT 23,≠X=≠,Y*Y+Z*Z
```

5. STORES Statement Extension

```
C$ STORES(variable1.operator.variable2)
```

6. Syntax Scan only Compiler Mode

FTN(Q), the program is scanned for syntax errors, and a reference map is produced.

7. Line Limit on Output File

OUTPUT file line control is provided by PL option on the control card.

```
FTN(R=3,OPT=2,PL=1000)
```

Also may be specified at object time.

```
LGO(PL=7000)
```

8. Multiple Systems Texts

Multiple systems texts may be specified on the FTN control card as:

```
FTN(S=IOTEXT,S=NUCLEUS/FTNMAC)
```

9. System Text Specification from a File

System texts may be specified on the FTN control card to be loaded from a local binary file as:

```
FTN(GT=MYTEXT,GT=TXTFILE/NEWTXT)
```

10. Exclusive OR Function

```
A=XOR(STRING,MASK)
```

11. Reference Map with Suppressed Program Listing

Any of the reference map options may be selected with the program listing suppressed, for example:

```
FTN(L=0,R=3)
```

12. Mass Storage Rewrite in Place

User may rewrite in place rather than at end-of-information.

13. Messages on STOP and PAUSE Statements

STOP	PAUSE
STOP n	PAUSE n
STOP≠string≠	PAUSE≠string≠

n octal number (up to five characters), ≠string≠ up to 70-character message

14. External text (XTEXT) Specification

An external text may be specified on the FTN control card as:

```
FTN (XT=TXTNAME)
```

15. MOVLEV Library Subroutine

```
CALL MOVLEV (from,to,n)
```

n=number of words

16. Integer Multiply Installation Option

CORRECTIONS

All eligible Programming System Report corrective code through Summary 312 is incorporated in the release program libraries.

LIMITATIONS

When the debugging compilation mode (control card parameter D) and the full cross reference features (R=2 or R=3) are used, more core may be required for compilation than otherwise. Detailed information is contained in the FORTRAN Extended Version 4.0 Reference Manual.

The intrinsic function SHIFT will not accept double word arguments (double precision or complex words).

FORTRAN Extended is designed to produce efficient object code. The rate of compilation tends to be higher on program units which avoid lengthy sequences of complicated arithmetic replacement statements (such as contiguous statements with no branching entries or exits). However, lengthy sequences tend to produce faster object code.

Code produced under the 6400 compiler option may not work properly on a 6600 because of optimization considerations. The compiler option to produce code for a 6600 produces binaries that can be run on either a 6400 or a 6600.

When the FTN control statement specifies either the C or E option, the compiler generates the object program in COMPASS source language form, rather than binary machine language. Since a local library set cannot be specified in a COMPASS source program, the user must place either the SCOPE control statement LIBRARY(FORTRAN,SYSIO) or the loader directive LDSET(LIB=FORTRAN/SYSIO) at appropriate points in the control card section of the job deck.

Under FTN 3.0 unformatted read and write statements processed S type records. Under FTN 4.0, W type records are the default style. Files of this type produced under 3.0 can be processed in the SCOPE 3.4 environment by using FILE and LDSET control cards to define the record type at object time.

Binary blocked and/or random indexed files created under FTN 3.0 cannot be handled directly by FTN 4.0: they must be converted to a file structure accepted by FTN 4.0 through use of the Conversion Aid utilities BBTO6RM and/or RANCONV.

INSTALLATION PARAMETERS

The amount of core needed to compile jobs can be altered by modifying the size of compiler tables and scratch file buffers. Parameters which modify table sizes are included in the options listed at the end of this section. File buffer sizes can be changed by modifying the controlling routine FTN. FTN is the main overlay of the compiler (level 0,0).

System compatibility parameters provide the following features. At the time of release, these parameters are set as listed below: UPDATE sequence numbers for all installation parameters may be determined by listing the OPTIONS portion of routine FTNTEXT and listing the routine FTN from release tape PL7.

Parameter Names		Release Setting
CCABT	Abort job on FTN control card error	Abort
CTIMO	Compiler message issued to control point dayfile (CPU seconds necessary for compilation)	Time message sent
LMAX	Lines per page listing limit (not applicable to intermixed COMPASS programs)	57

File names used by compiler:

INPUT	Source input
OUTPUT	Compiler listing
LGO	Relocatable object code
COMPS	COMPASS card images
ZZZZZRL	Intermediate language
ZZZZZRM	Reference map
ZZZZZOP	OPT=2 and D mode scratch

If the following control card options are not specified, the default settings at the time of release are as shown:

Control Card Option	Usage	Release Setting
A	Abort to EXIT(S) card if fatal compilation error occurs	No abort
B	Produce object code file	Produces object code on standard file (LGO)
C	Use COMPASS assembler for compiler generated code. If C is not selected, the FORTRAN assembler is used. (FORTRAN assembler saves about 60% of CPU time compared with COMPASS assembler.)	C is not set
D	Debug mode of compilation	No debug mode
E	Format file for editing (COMPASS card image file is produced with *DECK cards for each program unit, suitable as input for UPDATE)	No file for editing
G	Compile and go option	No compile and go
I = lfn	Select compiler input file	
	I Not Specified	lfn = INPUT
	I Specified without lfn	lfn = COMPILE
list = lfn	Select compiler listing file and listing options as follows:	lfn=OUTPUT list=L
	L List source code	L
	O List COMPASS card images	No list
	X List ANSI violation diagnostics	No list
	N Suppress informative diagnostics	No list
	R (equivalent to long reference map option R=2)	No list
	The R option may be used as a stand alone option of the form R = n if a reference level other than the default is required. The values for n select the following reference map options:	
	0 no reference map	
	1 short reference map	1
	2 full cross reference map	
	3 full cross reference map plus common and equivalence information	

Control Card Option	Usage	Release Setting
OPT=level	Select level of optimization: 0 Lowest optimization 1 Slightly above FORTRAN Extended 2.0 optimization 2 Program unit flow analysis used in optimization	OPT=1
Q	Program verification option	Q is not set
Round=s	s = */+- Select 1-4 of these operators to round arithmetic	No rounding
SYSEDIT=ss	This feature is intended for system programmer usage:	
	ss=FILES Form execution time input/output unit references through indirect search of low core table rather than by using entry points and external references.	None
T	Maximum error checking in mathematical library routines (basic external functions)	No error checking
V	Selects minimal input/output buffer allocation (513 words per buffer) for compiler buffers during compilation. This may increase compile time but will allow jobs with a large number of declarative statements to compile in smaller field length than would be possible otherwise.	
S	Specifies system text files (global library set) to be used for intermixed COMPASS programs.	SYSTEXT
GT	Specifies system text files (sequential binary file) to be used for intermixed COMPASS programs.	SYSTEXT
Z	Forces all subroutine calls with no parameters to pass a parameter list consisting of a zero word.	None
PL	Selects maximum number of allowable records on the OUTPUT file.	5000
XT	Specifies external text (XTEXT) to be used for intermixed COMPASS programs.	OLDPL
	XT specified without lfn	OPL

A listing of the Options portion of FTN follows:

```
****                                                                      F600250     2
***         O P T I O N S - FORTRAN EXTENDED INSTALLATION PARAMETERS.     F600250     3
*                                                                         F600250     4
*                                                                         F600250     5
*           THE PARAMETERS DEFINED IN THIS DECK DETERMINE THE SPECIFIC    F600250     6
*           CONFIGURATION OF THE FORTRAN EXTENDED COMPILER.  WHEN INSTAL- F600250     7
*           LING THE COMPILER, PARAMETERS SHOULD BE REVISED AS NECESSARY  F600250     8
*           TO DESCRIBE THE INSTALLATICN*S EXACT HARDWARE CONFIGURATICN   F600250     9
*           AND SPECIFIC FUNCTIONAL REQUIREMENTS.                         F600250    10
*                                                                         F600250    11
*           ADDITIONAL INSTALLATICN OPTION PARAMETERS WILL BE FOUND IN    F600250    12
*           THE PROGRAM DECK  < FTN > .                                   F600250    13


*           ************************************************************* F600250    15
*           *                                                           * F600250    16
*           *                                                           * F600250    17
*           *   I N S T A L L A T I O N   W A R N I N G   N O T I C E   * F600250    18
*           *                                                           * F600250    19
*           *                                                           * F600250    20
*           *      AN ASTERISKED BOX SIMILAR TO THIS ENCLOSES EACH      * F600250    21
*           *      PARAMETER THAT IS INTENDED TO BE CHANGED WHEN        * F600250    22
*           *      INSTALLING THE FORTRAN EXTENDED COMPILER.            * F600250    23
*           *                                                           * F600250    24
*           *      NO OTHER PARAMETER IN THE *OPTIONS* DECK SHOULD      * F600250    25
*           *      BE CHANGED.                                          * F600250    26
*           *                                                           * F600250    27
*           *      CONTROL DATA CORPORATION ACCEPTS NO RESPONSIBILITY   * F600250    28
*           *      FOR SATISFACTORY PERFORMANCE OF THE FORTRAN          * F600250    29
*           *      EXTENDED COMPILER IF THIS NOTICE IS DISREGARDED.     * F600250    30
*           *                                                           * F600250    31
*           *                                                           * F600250    32
*           ************************************************************* F600250    33

***         PROCESSOR NAME AND VERSION NUMBER                             F600307     2

 LPNAME     MICRO  1,7,/FTN    /     LANGUAGE PROCESSOR NAME              F600307     4
 VER        MICRO  1,3,/4.0/                                              F600307     5
```

```
                ***      CENTRAL PROCESSOR MODEL / SERIES DECLARATICNS.                         F600250      35
                                                                                                F600250      36
                                                                                                F600250      37
                *          ************************************************************        F600250      38
                *          *                                                          *        F600250      39
                *          *      TO INSTALL FORTRAN EXTENDED ON A CYBER 70/ MODEL XX *        F600250      40
                *          *      PROCESSOR, SET THE FOLLOWING SYMBOL TO THE APPRO-    *        F600250      41
                *          *      PRIATE MODEL NUMBER, AND DELETE THE SUBSEQUENT       *        F600250      42
                *          *      SYMBOL [ MACHINE. ].                                 *        F600250      43
                *          *                                                          *        F600250      44
       74        CYBERMOD  EQU    74B         CYBER 70/ MODEL NUMBER                            F600250      45
                *          *                                                          *        F600250      46
                *          *      REFERENCES -- FTN, FTNTEXT.                          *        F600250      47
                *          ************************************************************        F600250      48


                *          ************************************************************        F600250      50
                *          *                                                          *        F600250      51
                *          *      TO INSTALL FORTRAN EXTENDED ON A 6X00 / 7X00         *        F600250      52
                *          *      PROCESSOR, SET THE FOLLOWING SYMBOL TO THE APPRO-    *        F600250      53
                *          *      PRIATE SERIES NUMBER, AND DELETE THE PRECEDING       *        F600250      54
                *          *      SYMBOL [ CYBERMOD ].                                 *        F600250      55
                *          *                                                          *        F600250      56
     6600        MACHINE.  EQU    6600B       6X00 / 7X00 SERIES NUMBER                         F600250      57
                *          *                                                          *        F600250      58
                *          *      REFERENCES -- FAX, FTN, FTNMAC, OPTB, REFMAP.        *        F600250      59
                *          ************************************************************        F600250      60


                 C         IF     DEF,CYBERMOD                                                  F600250      62
                           IFLT   CYBERMOD,74B,2                                                F600250      63
                           IFEQ   CYBERMOD,74B,2                                                F600250      66
     6600        MACHINE.  EQU    6600B                                                         F600250      67
                 S         SKIP                                                                 F600250      68
                 S         ENDIF                                                                F600250      79
                                                                                                F600250      80
                *          [MACHINE.] REFERENCES -- FAX, FTN, FTNMAC, OPTB, REFMAP.             F600250      81
                                                                                                F600250      82
     6600        MACHINE   EQU    MACHINE.    FORMER NAME OF SYMBOL                             F600250      83

                 TARGET    OCTMIC MACHINE.                                                      F600307       7
                 V/I(JV    I]BR(  2/3/R=^CIH E1=R                                               (71141J       9


                **         DEFINE ≠MODEL≠ MICRO.                                                F600250      85
                                                                                                F600250      86
                 MODEL     OCTMIC CYBERMOD                                                      F600250      87
                                                                                                F600250      88
                *          [≠MODEL≠] REFERENCES -- FTN, FTNTEXT.                                F600250      89
```

```
             *      ****************************************************************** F600250     91
             *      *                                                                * F600250     92
             *      *      EARLY MODEL 6600 PARAMETER (STORE OUT OF ORDER).          * F600250     93
             *      *                                                                * F600250     94
             *      *      WHEN THE TARGET, I.E., OBJECT TIME, CPU IS A 6600         * F600250     95
             *      *      CLASS A, B OR C MACHINE, SERIAL NUMBER 1-47, THAT         * F600250     96
             *      *      LACKS FCO 20436, SET THE FOLLOWING PARAMETER TO ZERO.     * F600250     97
             *      *      (FCO 20436 RESCLVES A STORE OUT OF ORDER PROBLEM.)        * F600250     98
             *      *                                                                * F600250     99
     1       STOR6600 EQU    1                                                         F600250    100
             *      *                                                                * F600250    101
             *      *      REFERENCES -- OPTB.                                       * F600250    102
             *      ****************************************************************** F600250    103


             *      ****************************************************************** F600250    105
             *      *                                                                * F600250    106
             *      *      HARDWARE INTEGER MULTIPLY FEATURE.  VALUES ARE --         * F600250    107
             *      *                                                                * F600250    108
             *      *      1 = INTEGER MULTIPLY INSTALLED.                           * F600250    109
             *      *      0 = INTEGER MULTIPLY NOT INSTALLED.                       * F600250    110
             *      *                                                                * F600250    111
             *      ****************************************************************** F600250    112
             *      *                                                                * F600250    113
             *      *      [CPTIMULT] - DEFINE AVAILABILITY AT COMPILE TIME.         * F600250    114
             *      *                                                                * F600250    115
     0       CPTIMULT EQU    0            INTEGER MULTIPLY NOT AVAIL AT COMPILE TIME   F600250    116
             *      *                                                                * F600250    117
             *      *      REFERENCES -- FTNTEXT.                                    * F600250    118
             *      ****************************************************************** F600250    119
             *      *                                                                * F600250    120
             *      *      [OBJIMULT] - DEFINE AVAILABILITY AT OBJECT TIME.          * F600250    121
             *      *                                                                * F600250    122
     0       OBJIMULT EQU    0            INTEGER MULTIPLY NOT AVAIL AT OBJECT TIME    F600250    123
             *      *                                                                * F600250    124
             *      *      REFERENCES -- ARITH, DOPRE, MACROX, MACRS.                * F600266      1
             *      ****************************************************************** F600250    126


             **     7DM (DATA MANAGER) APPLICABILITY.  DENOTES WHETHER OR NOT 7DM      F600250    141
             *      WILL BE USED, BASED ON THE PROCESSOR MODEL.  SINCE THE SYMBOL      F600250    142
             *      IS MORE CONVENIENT IN A NEGATIVE SENSE, IT IS DEFINED AS --        F600250    143
             *                                                                         F600250    144
             *      1      7DM DOES NOT APPLY (MODEL 74 OR LOWER / 6X00 SERIES).       F600250    145
             *      0      7DM APPLIES (MODEL 75 OR HIGHER / 7X00 SERIES).             F600250    146
                                                                                       F600250    147
                    IFC    GE, ≠MODEL≠ 75 ,3                                           F600250    148
     1       NEDM   EQU    1                                                           F600250    152
```

```
                   MODEL     IFC     LT, ≠MODEL≠ 75                                         F600250     154
            0      CPERM     EQU     0              DO NOT ASSEMBLE 6RM                     F600250     165
                   MODEL     ENDIF                                                          F600250     168


                  *          *************************************************************** F600250     170
                  *          *       FILE CONTAINING CCMPASS RANDOM FL WITH /COMPCOM/.     * F600250     171
                  *          *                                                             * F600250     172
                   PLCMPS    MICRO   1,, COMPCOM                                              F600270       1
                  *          *                                                             * F600250     174
                  *          *************************************************************** F600250     175


                   PLRM      IFNE    CPERM*NEDM,0                                             F600250     177
                   PLRM      ENDIF                                                            F600250     185


                  *          *************************************************************** F600250     187
                  *          *                                                             * F600250     188
                  *          *       CHARACTER APPENDED TO FILE NAMES FOR UNIQUENESS.      * F600250     189
                  *          *                                                             * F600250     190
                   C         MICRO   1,,/E/                                                   F600250     191
                  *          *                                                             * F600250     192
                  *          *       REFERENCES -- BUGCTL,DBGFET,FTNMAC,LISTIO,PH1CTL.     * F600250     193
                  *          *************************************************************** F600250     194
```

```
                 *          ************************************************************** F600250
                 *          *                                                            * F600250
                 *          *      F I X E D   L E N G T H   T A B L E S.                * F600250
                 *          *                                                            * F600250
                 *          ************************************************************** F600250
                 *          *                                                            * F600250
                 *          *      LENGTH OF INTERMEDIATE STORAGE AREA FOR ALL CON-      * F600250
                 *          *      STANTS IN ANY ONE SOURCE STATEMENT.  EACH CONSTANT    * F600250
                 *          *      BEGINS ON A NEW WORD BOUNDARY, AND IS PACKED AS TEN   * F600250
                 *          *      DISPLAY-CODED CHARACTERS PER WORD.                    * F600250
                 *          *                                                            * F600250
      454       CONSTORS EQU     300D                                                      F600250
                 *          *                                                            * F600250
                 *          *      REFERENCES -- SCANNER.                                * F600250
                 *          ************************************************************** F600250
                 *          *                                                            * F600250
                 *          *      LENGTH OF SAVED OPERATOR STACK.                       * F600250
                 *          *                                                            * F600250
      240       MXOSE    EQU     160D                                                      F600250
                 *          *                                                            * F600250
                 *          *      REFERENCES -- ARITH.                                  * F600250
                 *          ************************************************************** F600250
                 *          *                                                            * F600250
                 *          *      LENGTH OF SAVED FUNCTION RESULT TABLE.                * F600250
                 *          *                                                            * F600250
      132       MXFRSTE  EQU     90D                                                       F600250
                 *          *                                                            * F600250
                 *          *      REFERENCES -- ARITH.                                  * F600250
                 *          ************************************************************** F600250
                 *          *                                                            * F600250
                 *          *      LENGTH OF <ARLIST> BUFFER.                            * F600250
                 *          *                                                            * F600250
     3200       ARLSZ    EQU     3200B                                                     F600250
                 *          *                                                            * F600250
                 *          *      REFERENCES -- PS1CTL.                                 * F600250
                 *          ************************************************************** F600250


                 *          ************************************************************** F600250
                 *          *                                                            * F600250
                 *          *      SELECT DIVISION BY RECIPROCAL MULTIPLICATION.         * F600250
                 *          *                                                            * F600250
                 *          *      WHEN THE FOLLOWING PARAMETER IS ZERO, <ARITH> WILL    * F600250
                 *          *      ATTEMPT TO REPLACE X/C BY X*(1/C), WHERE [X] IS AN    * F600250
                 *          *      EXPRESSION AND [C] IS A CONSTANT (REDUCTION IN        * F600250
                 *          *      STRENGTH).  SET NON-ZERO TO DEFEAT THE FEATURE.       * F600250
                 *          *                                                            * F600250
        0       NOINVERT EQU     0                                                         F600250
                 *          *                                                            * F600250
                 *          *      REFERENCES -- ARITH.                                  * F600250
                 *          ************************************************************** F600250
```

```
                 *          ************************************************************ F600250    247
                 *          *                                                          * F600250    248
                 *          *       ERROR LISTING LINE LIMIT.                          * F600250    249
                 *          *                                                          * F600250    250
          144     ERRMAX    EQU     1000                                                 F600250    251
                 *          *                                                          * F600250    252
                 *          *       REFERENCES -- ERPRO, LSTPRO.                       * F600250    253
                 *          ************************************************************ F600250    254
                                                                                         F600250    255
                                                                                         F600250    256
                 ****                                                                    F600250    257


                 ***        MICROS TO DEFINE THE DEFAULT CONTROL CARD OPTIONS.           F600250   1162
                 *                                                                       F600131    154
                 *                            OPTIONS  SELECTED     NOT SELECTED   FILE  F600131    155
                  CC.A      MICRO  1,, 0        A           1            0          NA   FE20005      1
                 * CHANGE +LGO+ MICRO BELOW     E=LFN  ALWAYS           NA         LFN   F600131    157
                  CC.G      MICRO  1,, 0        G          -1            0          NA   F600131    158
                  CC.C      MICRO  1,, 0        C          -1            0          NA   F600131    159
                  CC.D      MICRO  0            D          NA            0         LFN   F600131    160
                  CC.E      MICRO  1,, 0        E          -1            0               F600131    161
                  CC.GT     MICRO  1,,$SYSTEXT$ GT=LFN ALWAYS           NA         LFN   F600250   1163
                 * CHANGE +COMPS+ MICRO BELOW   E=LFN  ALWAYS           NA         LFN   F600131    162
                 * CHANGE +INPUT+ MICRO BELOW   I=LFN  ALWAYS           NA         LFN   F600131    163
                  CC.I      MICRO  1,,$COMPILE$ I                                        F600261      1
                  CC.L      MICRO  1,, 1        L           1            0          NA   F600250   1164
                  CC.LCM    MICRO  1,, 0       LCM         -1            0          NA   F600162      1
                  CC.O      MICRO  1,, 0        O          -1            0          NA   F600131    165
                  CC.X      MICRO  1,, 0        X          -1            0          NA   F600131    166
                  CC.N      MICRO  1,, 0        N          -1            0          NA   F600131    167
                 * CHANGE +OUTPUT+ MICRO BELOW  L=LFN  ALWAYS           NA         LFN   F600131    168
                  CC.PL     MICRO  1,, 5000     PL=N        N         5000          NA   F600215      1
                  CC.Q      MICRO  1,, 0        Q          -1            0          NA   F600158      1
                  CC.R      MICRO  1,, 1        R=     0, 1, 2, OR 3    NA          NA   F600131    169
                  CC.S      MICRO  1,,$SYSTEXT$ S=LFN  ALWAYS           NA         LFN   F600131    170
                  CC.T      MICRO  1,, 0        T          -1            0          NA   F600131    171
                  CC.XT     MICRO  1,,$OPL$     XT                                       F600296      1
                  CC.Z      MICRO  1,, 0        Z          -1            0          NA   F600159      4
                  CC.OPT    MICRO  1,, 1        OPT=   0, 1, OR 2       NA          NA   F600131    172
                  CC.ROUND  MICRO  1,, 0BS37    ROUND= 1B FOR +                     NA   F600131    173
                 *                                     2B FOR -                     NA   F600131    174
                 *                                     4B FOR *                     NA   F600131    175
                 *                                    10B FOR /                     NA   F600131    176
                  CC.IDENT  MICRO  1,, 0        SYSEDIT=IDENT                            F600131    177
                 *                                         -1            0          NA   F600131    178
                  CC.FILES  MICRO  1,, 1        SYSEDIT=FILES                            F600138      7
                 *                                          0            1          NA   F600138      8
                            IFC    EQ,/≠CC.OPT≠/0/,1                                     FE20041      1
```

```
                ***                                                                     FTN          59
                *         ASSEMBLY OPTIONS                                              FTN          60
                *                                                                       FTN          61
                          IFEQ    TEST,0                                                FTN          62
                          RMT                                                           FTN          63
                 W=SBW    EQU     CCERR              W=SPY BIN WIDTH                    FTN          64
                 M=LFN    EQU     CCERR              MACROX PATCH FACILITY              FTN          65
                          RMT                                                           FTN          67
                          ENDIF                                                         FTN          71
                                                                                        FTN          72
                                                                                        FTN          73
                ***                                                                     F600138       9
                *         LOCAL ASSEMBLY OPTIONS REFERENCED ONLY IN ≠FTN≠               F600138      10
                *                                                                       F600138      11
         1       CCABT    EQU     1                  SET TO NON ZERO VALUE FOR ABORT CN FTN          75
                *                                    CONTROL CARD ERRORS                FTN          76
                 EM       IFNE    CPERM,0                                               F600250    1165
                 EM       ENDIF                                                         F600250    1169
                                                                                        F600250    1170
        71       LMAX     EQU     57D                LINES / PAGE                       FTN          77
                 PSR      MICRO   1,4,/P213/         PSR LEVEL OF THE COMPILER          F600131     181
         1       CTIMO    EQU     1                                                     F600240       1
                *                                    =0 NO MESSAGE                      F600131     184
                *                                    =1 ISSUE DAYFILE MESSAGE FOR CPU   F600131     185
                 PSR      MICRO   1,4,/P310/          PSR LEVEL                         TITLE         1


                 CDC6466  OCTMIC MACHINE.            MACHINE THAT CODE IS GENERATED FOR F600131     188
                                                                                        FTN          82
                 K        MICRO   1,,/BS9/           *1000B                             FTN          85
                 TV       MICRO   1,,*1/2*           TRIGGER VALUE                      FTN          86
         1       TVS      EQU     1                  LOG2( 1/TV )                       FTN          87
                                                                                        F600077       2
                ***                                                                     F600077       3
                *         MIN.FL,MIN.DFL = MINIMUM FIELD LENGTH TO EXECUTE IN.          F600077       4
                *         MIN.FL = MAX(LWA LOAD)+400B+1000B ROUNDED UP TO NEXT 1000B    F600077       5
                *         WHERE:                                                        F600077       6
                *          MAX(LWA LOAD) = LAST WORD ADDR OF THE OVERLAY LOADS          F600077       7
                *          400B IS FOR MINIMUM RLIST AND COMPS BUFFERS                  F600077       8
                *          1000B IS FOR WORKING STORAGE                                 F600077       9
                *         CURRENTLY PASS 1 ( THE 1.1 OVERLAY ) IS THE LARGEST           F600077      10
                *                                                                       F600077      12
                *         FOR FL > MIN.FL+3K STANDARD SIZE BUFFERS (L.RLIST,L.COMPS)    F600077      13
                *         ARE ALLOCATED FOR RLIST AND COMPS.                            F600077      14
                *                                                                       F600077      15
     42000       MIN.FL   EQU     42≠K≠+TEST*2000B   MINIMUM FL FOR EXECUTION           F600322       1
     61000       MIN.DFL  EQU     MIN.FL+17≠K≠       MINIMUM FL IF D OPTION SELECTED    F600077      17
                                                                                        F600077      18
                 MIN.FL   OCTMIC  MIN.FL                                                F600077      19
                 MIN.FL   MICRO   1,2,/≠MIN.FL≠/                                        F600077      20
                 MIN.DFL  OCTMIC  MIN.DFL                                               F600077      21
                 MIN.DFL  MICRO   1,2,/≠MIN.DFL≠/                                       F600077      22


                *         DEFAULT FILE NAMES                                            FTN          94
```

```
                                                                                     FTN         95
              INPUT    MICRO  1,,$INPUT$                                             FTN         96
              OUTPUT   MICRO  1,,$OUTPUT$                                            FTN         97
              LGO      MICRO  1,,$LGO$           BINARY FILE NAME                    FTN         98
              COMPS    MICRO  1,,$COMPS$         COMPASS IMAGES                      FTN         99
              RLIST    MICRO  1,,$ZZZZZRL$       INTERMEDIATE CODE SCRATCH FILE      F600172     20
              RMAP     MICRO  1,,$ZZZZZRM$       REFERENCE MAP SCRATCH FILE          F600172     21
              OPT      MICRO  1,,$ZZZZZOP$       SUPER MODE SCRATCH FILE             F600172     22
                       IFNE   TEST,0,3                                               F600172     23

              *        DEFAULT BUFFER SIZES                                          FTN         105
              *         MAY BE ADJUSTED BY FTN, DEPENDING ON FIELD LENGTH AND        FTN         106
              *         CONTROL CARD OPTIONS                                         FTN         107
                                                                                     FTN         108
                                                                                     F600250     1173
              BUFL     MICRO  1,, 1002B   MINIMUM BUFFER SIZE                        F600250     1174
                                                                                     F600250     1175
                       RMT                                                           F600250     1176
              L.INPUT  EQU    IBUFL                                                  F600250     1177
              L.OUTPUT EQU    OBUFL                                                  F600250     1178
                       RMT                                                           F600250     1179
                                                                                     F600250     1180
    1002      L.COMPS  EQU    ≠BUFL≠      COMPASS SOURCE IMAGE BUFFER LENGTH         F600250     1181
    2004      L.LGO    EQU    ≠BUFL≠*2    BINARY OUTPUT BUFFER LENGTH                F600250     1182
    2004      L.OPT    EQU    ≠BUFL≠*2    MAX OPT (OPT=2) SCRATCH FILE BUFFER LENGTH F600250     1183
    2004      L.RLIST  EQU    ≠BUFL≠*2    INTERMED LANG (R-LIST) FILE BUFFER LENGTH  F600250     1184
    1002      L.RMAP   EQU    ≠BUFL≠      LONG REFERENCE MAP (R=3) BUFFER LENGTH     F600250     1185
                                                                                     FTN         116
              *        THE LENGTH≠S OF THE ≠INPUT≠ AND ≠LGO≠ BUFFERS MUST BE         FTN         117
              *        .GT. 1000B SINCE THEY MAY BE ON TAPE.                         FTN         118
                                                                                     FTN         119
       3      NOPTLVL  EQU    3                  NUMBER OF LEVELS OF OPTIMIZATION    FTN         120
```

COMPILER PROGRAM LIBRARY STRUCTURE

When a full update is performed on PL7, eight records are produced on the compile file as follows:

	Contents		Deck Names Needed to Compile
1.	FTNMAC		FTNMAC
2.	FTNTEXT		FTNTEXT
3.	Controller	(0,0)	FTN
		(1,0)	LSTPRO.SNAP
	Pass 1--normal	(1,1)	PS1CTL.PH1CTL
	Pass 2	(1,2)	CLOSE2.MACROX
		(1,3)	PS2CTL
4.	Pass 1-1/2	(1,5)	PASS15.CHECK
5.	Pass 1--debug	(1,4)	DBGPHCT.SAVREGS
6.	Debug COPYL deck		FORMDBG
7.	RDUMP		MACROR.RDUMPCP
8.	CMPAR		CMPAR

FTNMAC

This routine is a text collection of the macros necessary to assemble a FTN compiled job using the COMPASS assembler. Assembling the FTNMAC record from the compile file will produce a record suitable for entering the system through EDITLIB. The assembled routine is in system text format and can be used in FTN and COMPASS assemblies through the S parameter specification.

FTNTEXT

This record consists of a collection of macros, micros, and symbol definitions which facilitate the assembly of the FTN version 4.0 compiler. Assembling the FTNTEXT record from the compile file will produce a record in system text format that will be referenced through the GT parameter for compiler assemblies.

(0,0) Overlay (FTN)

This overlay is a batch controller which accomplishes the following:

Breaks down the parameters on the FTN control card

Initializes the compiler to reflect compile time options

Contains basic I/O routines and system communication routines

Handles COMPASS-FTN communication for intermixed COMPASS language program units

(1,0) Overlay (LSTPRO$)

This overlay holds information necessary for communication between passes of the compiler. It contains the symbol/label table lookup routine (needed by pass 1, pass 1-1/2, and the assembly phase of pass 2); a routine to allow the usage of formatted I/O in the FORTRAN coded routines which are a part of the debug and pass 1-1/2 portions of the compiler; and a snap facility active only when the compiler is in test mode.

(1,1) Overlay (PS1CTL$)

This overlay is the first pass of the compiler under normal mode (when the debug mode is not selected by the D control card option); it handles lexical, syntactical and semantic analysis of a FORTRAN program unit. The FORTRAN program unit is translated through a lexical element language (E-list) to an intermediate language (R-list).

(1,2) Overlay (CLOSE2$)

This overlay is the second pass of the compiler. Using the R-list generated by pass 1, this pass produces COMPASS instructions which are assembled (optionally by the COMPASS assembler at the expense of CPU time) into an object deck.

(1,3) Overlay (PS2CTL)

Overlay 1,3 is loaded only if errors occur in the FORTRAN program. This overlay issues full line error messages for both fatal and informative errors.

(1,5) Overlay (PASS15$)

This overlay is selected optionally (OPT=2 on the FTN control card); it is executed between the normal pass 1 and pass 2. This pass (1-1/2) uses the R-list language generated by pass 1 as input and, utilizing flow analysis, produces a modified R-list file for input to pass 2. This file will reflect the movement of invariant code from frequently executed regions to those less frequently executed and it will allow register allocation over loops.

(1,4) Overlay (PASS14$)

This overlay serves the same purpose as overlay (1,1) with the addition that debug statements will be acted upon as specified by the D option in the FTN control card. On the program library, only the routines unique to the debug mode constitute the fifth record. The sixth record of the compile file constitutes a full overlay (1,4).

Debug COPYL Deck

This deck contains a series of zero length COMPASS routines with identifiers of the (1,1) overlay routines needed to complete the (1,4) overlay. By compiling the fifth and sixth records of the compile file to the same object file, the object file may be used as an OLDLIB file for COPYL with the object file of the (1,1) overlay serving as the replacement file to produce a full (1,4) overlay.

RDUMP

This utility routine is to be used with the test mode of the compiler. For a single program unit, RDUMP will dump the R-list file generated by pass 1 of the compiler. (Since an attempt is made to keep this file completely in core, the V (for very small buffers) option on the control card should be used; it is active in test mode only.)

CMPAR

This routine compares two object records and lists discrepancies by loader table. It is unlikely that a FORTRAN program using the FTN4.0 internal assembler will have the same object representation as when it is handled by the COMPASS assembler (though a load from either of the object routines will produce the same core image in non-BSS storage).

INSTALLATION INSTRUCTIONS

The release tape for the FORTRAN Extended 4.0 compiler, PL7, contains 5 files. File 1 is the compiler program library. Files 2, 3, and 4 contain installation decks; file 5 contains an installation verification program. The decks in files 2 through 5 can be obtained by performing the job:

```
Job card.
REQUEST(PL7,E)
REWIND(PL7)
SKIPF(PL7,1,17)              SKIP PL AND BINARY FILES
COPYBF(PL7,PUNCH,3)          INSTALLATION DECKS
COPYBF(PL7,PUNCH,1)          VERIFICATION PROGRAM
UNLOAD(PL7)
6/7/8/9
```

The installation decks provide a method for introducing the FORTRAN Extended 4.0 compiler into a SCOPE 3.4 system. The first job (CINSTAL) updates the program library, producing a new program library tape including supplemental binary files. Deck CINSTAL requires access to the COMPASS program library to acquire the common deck COMPCOM. The second job, CMAINT, updates the program library and supplemental binary files producing a new maintenance form tape. Deck CMAINT also requires access to the COMPASS program library. CINSTAL should be used only for initial installation; CMAINT should be used for subsequent maintenance. Deck FTNC2 must be run following CINSTALL or CMAINT but before attempting installation of the object library.

The release tape for the FORTRAN Extended 4.0 object library, PL8, contains three files. File 1 is the FORTRAN object library PL. File 2 contains a maintenance procedure to be used with SCOPE 3.4; file 3 contains deck LIB4. To obtain these decks, perform a job of the type

```
Job card.
REQUEST(PL8,E)
REWIND(PL8)
SKIPF(PL8,1,17)
COPYBF(PL8,PUNCH,2)
UNLOAD(PL8)
6/7/8/9
```

The installation decks provide a method for updating the program library and generating binary files to be introduced into the system. Deck FTN4LIB is the SCOPE 3.4 procedure to update the program library and generate binaries to be used by FTN Version 4.0 programs. PL8 deck LIB4 must be run following FTN4LIB; upon completion, job SCOPE3 (Section 1) can be run to generate a deadstart tape.

NOTE: The T7000 parameter on the CINSTAL job card must be changed to T20000 if the deck is to be run on a 6200.

```
CINSTAL,CM65000,T7000,MT2.
COMMENT.  THIS JOB UPDATES FORTRAN EXTENDED 4.0 FROM THE RELEASED PL7.
COMMENT.  AN UPDATED PROGRAM LIBRARY, TWO FILES WHICH WILL BE OF USE IN
COMMENT.  MAINTAINING THE COMPILER, AND A FILE FOR EDITLIBING THE COMPILER
COMMENT.  INTO A RUNNING SYSTEM WILL BE PRODUCED ON THE TAPE NEWPL7.
COMMENT.
COMMENT.  THE CONTENTS OF THE RELEASED PL7 ARE --
COMMENT.       FILE 1 -- COMPILER PROGRAM LIBRARY
COMMENT.       FILE 2 -- INSTALLATION DECK
COMMENT.       FILE 3 -- MAINTENANCE DECK
COMMENT.       FILE 4 -- EDITLIB DECK FOR THE COMPILER
COMMENT.       FILE 5 -- VERIFICATION PROGRAM DECK
COMMENT.
COMMENT.  THIS JOB USES THE COMPASS V3.0 PL TO CAPTURE THE COMPASS
COMMENT.  COMMON DECK COMPCOM WHICH IS NECESSARY FOR COMPASS/FTN
COMMENT.  INTERFACE.
COMMENT.
REQUEST(PL2,E,HI)             COMPASS PROGRAM LIBRARY
UPDATE(Q,P=PL2,N=COMPCOM,C=0)
UNLOAD(PL2)
REQUEST(PL7,E,HI)             RELEASE PL7
REQUEST(NEWPL7,N,HI)          TAPE TO RECEIVE MODIFIED PL
LABEL(NEWPL7,W,L=FTN4P0COMP*3P4,D=HI)
REWIND(PL7,NEWPL7)
UPDATE(F,P=PL7,N=NEWPL7,R=C)
COPYBF(ECF,NEWPL7)
UNLOAD(PL7)
COMPASS(I=COMPILE,L=LISTFTN,B=FTNMAC,S=0)
COMPASS(I=COMPILE,L=LISTFTN,B=FTNTEXT,S=0)
COMPASS(I=COMPILE,L=LISTFTN,B=SYSMAIN,S=IPTEXT,G=FTNTEXT)
REWIND(SYSMAIN)
MAP(OFF)
LOAD(SYSMAIN)
NOGO.
EDITLIB(USER)
RETURN(FTN)
LIBRARY(NEWFTN)
FTN(I=COMPILE,L=LISTFTN,GT=FTNTEXT,S=0,B=SYSOPT,OPT=1)
FTN(I=COMPILE,L=LISTFTN,GT=FTNTEXT,S=0,B=FDEBUG,OPT=1)
FTN(I=COMPILE,L=LISTFTN,GT=FTNTEXT,S=0,B=FDEBUG,OPT=1)
LIBRARY.
REWIND(SYSMAIN,FDEBUG)
COPYL(FDEBUG,SYSMAIN,SYSDBG)
REWIND(SYSMAIN,SYSOPT,SYSDBG)
RETURN(FDEBUG)
COPYEF(SYSMAIN,SYSFTN)
SKIPB(SYSFTN)
COPYBF(SYSOPT,SYSFTN)
REWIND(SYSFTN)
COPYBF(SYSFTN,NEWPL7)
SKIPB(SYSFTN)
COPYBF(SYSDBG,SYSFTN)
REWIND(SYSDBG)
COPYBF(SYSDBG,NEWPL7)
REWIND(SYSFTN)
MAP(ON)
LOAD(SYSFTN)
NOGO.
REWIND(FTNMAC,FTN)
```

```
COPYBR(FTNMAC,NEWPL7)
COPYBF(FTN,NEWPL7)
UNLOAD(NEWPL7)
REQUEST(LISTAPE,HI)          TAPE TO RECEIVE COMPILER LISTING
REWIND(LISTFTN)
COPYBF(LISTFTN,LISTAPE)
UNLOAD(LISTAPE)
7/8/9  END OF RECORD
*C COMPCOM
7/8/9  END OF RECORD
*/   PLACE ANY INSTALLATION MODIFICATIONS AFTER THIS CARD.
7/8/9  END OF RECORD
LIBRARY(NEWFTN,NEW)
ADD(FTN,FTN,AL=3)
FINISH.
LIBRARY(SYSOVL,NEW)
ADD($LSTPRC$$$+$FTNMSG$$$,FTN)
FINISH.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
CMAINT,CM65000,T7000,MT2.
COMMENT.  THIS JOB UPDATES THE FORTRAN EXTENDED COMPILER AND CREATES
COMMENT.  A NEW PL7 WITH PROGRAM LIBRARY AND COMPILER BINARIES.
COMMENT.
COMMENT.  THE CONTENTS OF THE NEWPL7 TAPE ARE --
COMMENT.       FILE 1 -- FORTRAN COMPILER PROGRAM LIBRARY
COMMENT.       FILE 2 -- RELOCATABLE BINARIES OF THE NON DEBUG OVERLAYS
COMMENT.       FILE 3 -- RELOCATABLE BINARIES OF THE DEBUG OVERLAY
COMMENT.       FILE 4 -- FTNMAC AND ABSOLUTE BINARIES OF THE COMPILER
COMMENT.
COMMENT.  THIS JOB USES THE COMPASS V3.0 PL TO CAPTURE THE COMPASS
COMMENT.  COMMON DECK COMPCOM WHICH IS NECESSARY FOR COMPASS/FTN
COMMENT.  INTERFACE.
COMMENT.
REQUEST(PL2,E,HI)           COMPASS PL
UPDATE(Q,P=PL2,N=COMPCOM,C=0)
UNLOAD(PL2)
REQUEST(OLDPL7,E,HI)        PL TO BE UPDATED
REQUEST,NEWPL7,N,HI.    TAPE TO RECEIVE MODIFIED PL
LABEL(NEWPL7,W,L=FTN4POCOMP*3P4,D=HI)
REWIND(OLDPL7,NEWPL7)
UPDATE(P=OLDPL7,N=NEWPL7,R=C)
COPYBF(ECF,NEWPL7)
COPYBF(OLDPL7,NIL)
COMPASS(I=COMPILE,L=0,S=0,B=FTNMAC)
COMPASS(I=COMPILE,L=0,S=0,B=FTNTEXT)
FTN(I=COMPILE,GT=FTNTEXT,S=IPTEXT,L=0,B=REPLACE,OPT=1)
FTN(I=COMPILE,GT=FTNTEXT,S=IPTEXT,L=0,B=REPLACE,OPT=1)
FTN(I=COMPILE,GT=FTNTEXT,S=IPTEXT,L=0,B=REPLACE,OPT=1)
COPYL(OLDPL7,REPLACE,SYSMAIN)
COPYL(OLDPL7,REPLACE,SYSDBG)
UNLOAD(OLDPL7)
REWIND(SYSMAIN,SYSDBG)
COPYBF(SYSMAIN,NEWPL7)
COPYBF(SYSDBG,NEWPL7)
SKIPF(SYSMAIN)
```

```
REWIND(SYSDBG)
COPYBF(SYSDBG,SYSMAIN)
MAP(ON)
LOAD(SYSMAIN)
NOGO.
REWIND(FTN,FTNMAC)
COPYBR(FTNMAC,NEWPL7)
COPYEF(FTN,NEWPL7)
UNLOAD(NEWPL7)
7/8/9  END OF RECORD
*C COMPCOM
7/8/9  END OF RECORD
*C FTNMAC,FTNTEXT
*/   PLACE ANY COMPILER MODIFICATIONS AFTER THIS CARD.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
FTNC2,CM55000,T7000,MT1.
COMMENT.
COMMENT. THIS JOB EDITLIBS THE FTN COMPILER INTO THE RUNNING SYSTEM
COMMENT. FROM THE TAPE MADE BY DECK CINSTAL OR CMAINT
COMMENT.
COMMENT.
LABEL(FTNCOMP,R,L=FTN4P0COMP*3P4,D=HI)  MOUNT PL7
REWIND,FTNCOMP.
SKIPF(FTNCOMP,3,17)
COPYBF(FTNCOMP,FTNC)
REWIND FTNC,FTNCOMP.
UNLOAD FTNCOMP.
EDITLIB(SYSTEM)
7/8/9  END OF RECORD
READY(SYSTEM,OLD).
LIBRARY(NUCLEUS,OLD).
REPLACE(FTNMAC,FTNC).
REPLACE(FTN,FTNC,AL=3,FL=47000,FLO=1).
FINISH.
LIBRARY(SYSOVL,OLD).
REWIND(FTNC).
REPLACE($LSTPRO$$$+$PASS14$$$,FTNC).
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

The T1000 parameter on this job card is insufficient on a 6200 or 6400. Please increase the value to T2000.

```
FTN4LIB,CM60000,T1000,MT2.
COMMENT.  THIS SCOPE 3.4 JOB UPDATES THE FORTRAN EXTENDED 4.0 OBJECT
COMMENT.  LIBRARY FROM THE RELEASED PL8 OR FROM A NEWPL8 TAPE.  AN
COMMENT.  UPDATED PROGRAM LIBRARY AND A FILE CONTAINING ASSEMBLED
COMMENT.  BINARIES OF THE LIBRARY WILL BE PRODUCED ON THE TAPE NEWPL8.
COMMENT.
COMMENT.  THE CONTENTS OF THE RELEASED PL8 ARE --
COMMENT.        FILE 1 -- FORTRAN OBJECT LIBRARY PL
COMMENT.        FILE 2 -- 3.4 INSTALLATION DECK
COMMENT.        FILE 3 -- 3.4 EDITLIB DECK
COMMENT.        FILE 4-6 -- 2.0 INSTALLATION DECKS
```

```
REQUEST(PL8,E,HI)        RELEASE PL8 OR NEWPL8 FROM PRIOR RUN
REQUEST,NEWPL8,N,HI.         TAPE TO RECEIVE UPDATED PL
LABEL(NEWPL8,W,L=FTNLIBS*3P4,D=HI)
REWIND(PL8,NEWPL8)
UPDATE(F,P=PL8,N=NEWPL8,R=C)
COPYBF(EOF,NEWPL8)
UNLOAD(PL8)
COMPASS(I=COMPILE,L=LISTLIB,S=SYSTEXT,S=IPTEXT,B=LIBRARY)   I/O-MATH LIBRARY
FTN(I=COMPILE,L=LISTLIB,S=SYSTEXT,S=IPTEXT,B=LIBRARY,OPT=1)FTN DEBUG LIBRARY
REWIND(LIBRARY)
COPYBF(LIBRARY,NEWPL8)
UNLOAD(NEWPL8)
REQUEST(LISTAPE,HI)      TAPE TO RECEIVE LIBRARY LISTING
REWIND(LISTAPE)
REWIND(LISTLIB)
COPYBF(LISTLIB,LISTAPE)
UNLOAD(LISTAPE)
7/8/9  END OF RECORD
*/    PLACE ANY LIBRARY MODIFICATIONS AFTER THIS CARD.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
LIB4,CM55000,MT01,T7000.
COMMENT. THIS JOB EDITLIBS THE FTN LIBRARY INTO THE RUNNING SYSTEM.
LABEL(FTNLIB,R,L=FTNLIBS*3P4,D=HI)MOUNT PL8 MADE BY DECK FTN4LIB
REWIND,FTNLIB.
SKIPF(FTNLIB,1,17)
COPYBF(FTNLIB,FTNL)
REWIND FTNLIB,FTNL.
UNLOAD FTNLIB.
EDITLIB(SYSTEM)
7/8/9  END OF RECORD
READY(SYSTEM,OLD)
LIBRARY(FORTRAN,OLD).
REPLACE(*,FTNL).
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

Dayfile output of the FTN 4.0 verification program should be similar to the following:

```
            SCOPE 3.4
05.44.23.VFTN03Z
05.44.23.VFTN,CM55000,T200.
05.44.23. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
05.44.23.STALLATION OF
05.44.23. FORTRAN EXTENDED 4.0
05.44.23.FTN(G,OPT=1)
05.44.24.         .727 RT SECONDS LOAD TIME
05.44.28.         .356 CP SECONDS COMPILATION TIME
05.44.28.LGO
05.44.33.        4.754 RT SECONDS LOAD TIME
05.44.33.COMPUTATION SUCCESSFUL
05.44.33.     STOP
05.44.33.BKSP(INPUT)
05.44.35.        1.485 RT SECONDS LOAD TIME
05.44.35.RETURN(LGO)
05.44.37.FTN(G,OPT=2)
05.45.02.       11.035 RT SECONDS LOAD TIME
05.45.07.         .434 CP SECONDS COMPILATION TIME
05.45.07.LGO
05.45.12.        4.703 RT SECONDS LOAD TIME
05.45.12.COMPUTATION SUCCESSFUL
05.45.12.     STOP
05.45.12.BKSP(INPUT)
05.45.14.         .975 RT SECONDS LOAD TIME
05.45.14.RETURN(LGO)
05.45.14.FTN(G,OPT=0)
05.45.27.        2.013 RT SECONDS LOAD TIME
05.46.21.         .238 CP SECONDS COMPILATION TIME
05.46.21.LGO
05.47.10.       48.476 RT SECONDS LOAD TIME
05.47.10.COMPUTATION SUCCESSFUL
05.47.10.     STOP
05.47.10.    END OF JOB
05.47.10.CPA      3.492 SEC.
05.47.10.PP      27.262 SEC.
```

COBOL VERSION 4.0

RELEASE MATERIALS

COBOL Version 4.0 release material consists of a magnetic tape containing the program library.

The source code has been resequenced completely.

NEW FEATURES

New features incorporated in version 4.0 include the following:

1. New verbs (REWRITE, DELETE, SKIP) to support IS and DA files and future COBOL standardization.
2. All I/O handled through Record Manager to provide compatibility with other compilers.
3. Compliance with ANSI COBOL standards. Non-ANSI usage and statements can be diagnosed by the compiler.
4. Faster execution times as a result of generating more code inline and more optimization.
5. Compilation and execution of version 3.0 programs under control card option.
6. Selection of 6400 or 6600 object code for more execution efficiency.
7. Dynamic sort area and buffer assignment at execution time, therfore a reduction in field length requirements.

CORRECTIONS

All eligible PSR code has been added to the program library including all code as published through PSR Summary No. 314.

LIMITATIONS

Source decks may need modifying if the following are used:

USE procedures. The USE AFTER STANDARD LABEL Procedure on output files now is executed just before the label is written. Previously, it was executed after the label was written.

Positive sign presence on signed fields. The positive sign will not always be carried. The absence of a negative sign now signals positive. Existing tests must be changed that redefine the field as alphanumeric and compare with the letters A-I or < (less than).

Random files. Version 3.0 random files must be converted to 4.0 format with the supplied conversion routine. (See CONVERSION AIDS discussion in Section 6).

ENTER statements containing a parameter to reference an FET. COBOL 4.0 establishes FIT's which contain a pointer to the FET; the code must be altered to go through the FIT to the FET.

Blank fill on variable length records. For reasons of efficiency COBOL 4.0 does not guarantee blank fill on variable length records, except for Z type records. Version 3.0 blank filled all variable length records. Care must be taken that unused portion of the record area are not expected to contain blanks.

Non-standard labels are no longer in the record area. They are read into each label area only, and any references to the record area will have to be changed.

End of file cards: On INPUT, Version 3 considered 7/8/9 level 15 as end of file. Version 4 considers all 7/8/9 cards as end of file.

When an elementary A/N item of more than 18 characters is moved to a numeric edited field, the rightmost are moved rather than the leftmost.

No editing on a group to a numeric edited field.

Signed numeric to an equal A/N field removes the sign. To an unequal size field, the sign is not removed.

INSTALLATION PARAMETERS

The COBOL compiler uses symbol definitions from IPTEXT for IP.CMU, IP.IMUL, and IP.TYPE (see SCOPE section discussion of IPARAMS). To override these installation parameter values, make the following changes in the COMDECK ASSEMOP when COBOL is assembled.

Feature	Change Required
Generate code optimized for a 6600	*D ASSEMOP.3, 4
Generate code optimized for a 6400	*D ASSEMOP.3
Generate integer multiply instruction code	*D ASSEMOP.6, 7
Generate non-integer multiply instruction code	*D ASSEMOP.6
Generate CMU instructions	*D ASSEMOP.9, 10
Generate non CMU instructions	*D ASSEMOP.9

INSTALLATION PROCEDURES

The release tape, PL9, contains seven files. File one contains the COBOL 4.0 program library. This file includes both compiler and object routines. Files 2-4 contain binary decks of the object time routines and the compiler overlays. Files 5 and 6 contain the installation decks, and file 7 contains the sample installation verification program.

The installation job decks add COBOL to the running system. The first job (COBOL1) uses the release tape as input to create a tape containing four files as output:

File 1 Update version of the COBOL program library

File 2 Relocatable binary records resulting from assembly plus the COBOL system routines

File 3 Overlays forming the COBOL 4.0 compiler that will be installed into the system

File 4 COPYCL routine in absolute form

When the second job (COBOL2) is performed, using either the released tape or the output tape created by the first job (COBOL1), COBOL 4.0 is added to the running system by EDITLIB. Job SCOPE3, described in Section 1, can be used to generate a deadstart tape.

These installation decks can be acquired by performing the job:

```
Job card.
REQUEST(PL9,E)               MOUNT COBOL 4.0 PL
REWIND(PL9)
SKIPF(PL9,4,17)              SKIP COBOL PL AND BINARY
COPYBF(PL9,PUNCH,2)          PUNCH INSTALLATION DECKS
COPYBF(PL9,PUNCH,1)          PUNCH VERIFICATION PROGRAM DECK
UNLOAD(PL9)
6/7/8/9
```

```
COBOL1,CM60000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF COBOL
COMMENT. THE NEW PL9 WILL CONSIST OF FOUR FILES
COMMENT. THE FIRST FILE WILL BE THE COBOL NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY OF THE OBJECT TIME ROUTINES
COMMENT. AND THE RELOCATABLE BINARIES OF COBOL (COMPILER OVERLAYS) AND
COMMENT. COPYCL
COMMENT. THE THIRD FILE WILL BE THE BINARY OF THE COMPILER OVERLAYS
COMMENT. THE FOURTH FILE WILL BE THE BINARY OF THE OVERLAY COPYCL
LABEL(PL9IN,R,L=COBOL4P0*3P4,D=HI) COBOL OLDPL
REQUEST,PL9,N,HI.     SCRATCH FOR NEW PL9
LABEL(PL9,W,L=COBOL4P0*3P4,D=HI)
REWIND(PL9IN,PL9)
UPDATE(F,P=PL9IN,N=PL9,X)
UNLOAD(PL9IN)
COMPASS(I=COMFILE,S=IOTEXT,S=IPTEXT,L=0)
COMPASS(I=COMFILE,S=CPCTEXT,L=0)
COMPASS(I=COMFILE,S=CPCTEXT,L=0)
SKIPF(PL9,1,17)
REWIND(LGO)
COPYBF(LGO,PL9)
REWIND,LGO.
COPYBR(LGO,NIL,90)
COPYBR(LGO,NEWLGO,102)
COPYBR(LGO,COPYLGO,3)
LDSET(LIB=SYSIO)
LOAD(NEWLGO)          GENERATION OF OVERLAY COBOL
NOGO.
REWIND,COBOL.
COPYBF(COBOL,PL9)
LOAD(COPYLGO)         GENERATION OF OVERLAY COPYCL
NOGO.
REWIND,COPYCL.
COPYBF(COPYCL,PL9)
UNLOAD(PL9)
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
COBOL2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS THE COBOL V4.0 BINARIES FROM THE RELEASED
COMMENT. COBOL PL OR THE PL CREATED BY DECK COBOL1.
LABEL(PL9,R,L=COBOL4P0*3P4,D=HI)  MOUNT COBOL 4.0 PL
REWIND,PL9.
SKIPF(PL9,1,17)             SKIP OLDPL
COPYBR(PL9,COBL,90)         COBOL LIBRARY ROUTINES
SKIPF(PL9,1,17)             SKIP RELOCATABLE COMPILER ROUTINES
COPYBF(PL9,COBOV)           COBOL ABSOLUTE COMPILER ROUTINES
COPYBF(PL9,COPYCL)          COPYCL
REWIND,PL9,COBL,COBOV,COPYCL.
UNLOAD(PL9)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
```

```
REPLACE(COBOL,COBOV,AL=3,FL=56000,FLO=1)
REPLACE(COPYCL,COPYCL,AL=3,FL=35000,FLO=1)
FINISH.
LIBRARY(SYSCVL,CLD)
REPLACE(*,COBOV)
FINISH.
LIBRARY(COBCL,CLD)
REPLACE(*,COBL)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

The dayfile output for the COBOL 4.0 verification program is listed below.

```
            SCOPE 3.4
05.44.24.VCOB030
05.44.25.VCOB,CM60000,T20.
05.44.25. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
05.44.25.STALLATION OF
05.44.25. COBOL 4.0
05.44.25.COBOL(LX)
05.44.28.      2.492 RT SECONDS LOAD TIME
05.44.31.COMPILING COBOL-V
05.44.42. 000 E AND 000 T/U  DIAGNOSTICS ISSUED
05.44.42. FIELD LENGTH NEEDED FOR COBOL 052300
05.44.42.       .203 CP SECONDS COMPILATION TIME
05.44.42.END COBOL
05.44.42.MAP(OFF)
05.44.42.LGO.
05.44.51.      7.862 RT SECONDS LOAD TIME
05.44.51.  COBOL 4.0 EXISTS UNDER SCOPE 3.4
05.44.52.    END OF JOB
05.44.52.CPA     1.118 SEC.
05.44.52.PP     11.554 SEC.
```

SORT/MERGE VERSION 4.0

RELEASE DESCRIPTION

SORT/MERGE 4.0 runs under 6000 SCOPE 3.4 and 6000 RECORD MANAGER 1.0.

RELEASE MATERIALS

SORT/MERGE 4.0 is released on program library tape PL10.

HARDWARE CONFIGURATION

SORT/MERGE 4.0 requires the same minimum hardware configuration as SCOPE 3.4. If the Tape Sort option is used, additional magnetic tape units are required: polyphase requires three; balanced requires four.

NEW FEATURES

The following new features were added for user convenience and system reliability:

FREE FIELD CONTROL CARDS

Two SORT/MERGE control card directive types are available to the user. Format 1, an upward 7000 based compatible design, has free field directives beginning in columns 1-72. Format 2 is based on 6000 SORT/MERGE 3.0 control card formats. Format 2 control cards will enable the user to run 3.0 jobs without rewriting them. New SORT/MERGE jobs should be written in Format 1.

UNLIMITED SIZE AND/OR NUMBER OF FILES

SORT/MERGE 4.0 imposes no restrictions on the size of the file or number of files to be sorted in a single run, other than the restrictions under 6RM. Currently 100 input files are allowed.

COLLATING SEQUENCE CHARACTERS

Characters within a collating sequence may be equated by the user.

OWNCODE EXITS

The system provides the user with additional OWNCODE exits, EXIT5 and EXIT6. The EXIT5 option exits to the user when two records with identical keys are found. EXIT6 is required for non-standard labeled tapes in directive Format 1. EXIT6 supplies the non-standard label to the user for validation on input and exits to the user for the non-standard label on output. When directive Format 2 is used, non-standard labels are skipped and not processed.

A NEW SORT/MERGE FORMAT

The 6000 SORT/MERGE 3.0 Macro format has been replaced with a new format which is upward 7000 compatibile.

MODIFICATIONS

1. The 4.0 SORT/MERGE system does not provide the TAG SORT capability.

2. 6RM handles the following:

   Record types
   Block types
   Parity error processing
   Padding character processing
   Label processing
   Multi-reel and multi-file reel processing
   Disk overflow detection (SORT/MERGE still handles disk overflow recovery)

GENERAL DESCRIPTION

SORT/MERGE 4.0 runs under SCOPE 3.4 and 6RM 1.0. The system consists of two control card directive formats and a Macro Sort format. Directive format 1 is based on upward compatibility toward 7000 SORT/MERGE and Format 2 is based on the SORT/MERGE 3.0 control card format. The Macro Sort format is also based on 7000 compatibility. SORT/MERGE 4.0 is a more modularized package, consisting of overlay modules which are in core only when necessary. (For example, a disk sort does not need the tape merge overlay modules.) This product is designed to optimize speed and core space as well as to utilize 6RM and SCOPE 3.4 capabilities.

INSTALLATION PROCEDURE

PL10 contains the following files:

File	Contents
1	SORT/MERGE program library
2	SORT/MERGE relocatable binary decks (macro sort)
3	SORT/MERGE relocatable binary decks (control card sort)
4	SORT/MERGE binary overlays
5	SMTEXT binary
6	SORT1 Installation deck
7	SORT2 Installation deck
8,9,&10	Verification programs

To obtain the installation and verification program decks, perform a job of the type:

```
Job card.
REQUEST (PL10,E)          Assign SORT PL
SKIPF(PL10,5,17)
COPYBF(PL10,PUNCH,5)
UNLOAD(PL10)
6/7/8/9
```

The installation jobs function as follows:

SORT1 — Updates the program library with modifications producing a new program library tape including assembled binary information as supplemental files. This job essentially allows creation of a revised release tape.

SORT2 — Adds SORT/MERGE to the running system. SORT2 can use either the released PL10 or a tape created by job SORT1 as input.

After job SORT2 has been run, job SCOPE3 described in Section 1 can be run to create a deadstart tape of the running system containing SORT/MERGE.

The system text SMTEXT is installed via execution of installation decks SORT1 and SORT2.

```
SORT1,CM55000,T7000,MT2.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF SORT/MERGE
COMMENT.  THE NEW PL10 WILL CONSIST OF FIVE FILES
COMMENT. THE FIRST FILE WILL BE THE NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY OF THE RELOCATABLE ROUTINES FOR
COMMENT.  MACRO SORT
COMMENT. THE THIRD FILE WILL BE THE BINARY OF THE RELOCATABLE ROUTINES FOR
COMMENT. CONTROL CARD SORTS
COMMENT.  THE FOURTH FILE WILL BE BINARY OF THE SORT OVERLAYS
COMMENT. THE FIFTH FILE WILL BE THE BINARY OF SMTEXT
LABEL(PL10IN,R,L=SORT4P0*3P4,D=HI) MOUNT SORT V4.0 PL
REQUEST,PL10,N,HI.    SCRATCH FOR NEW PL10
LABEL(PL10,W,L=SORT4P0*3P4,D=HI)
REWIND(PL10IN,PL10)
UPDATE(F,P=PL10IN,N=PL10,X)
COMPASS(I=COMPILE,S=IOTEXT,S=IPTEXT,B=SRTOBJ,L=0)    COMPILE ALL BINARIES
COMPASS(I=COMPILE,S=0,L=0)                     COMPILE SMTEXT
REWIND,SRTOBJ.
REWIND(LGO)
SKIPF(PL10,1,17)
REWIND(PL10IN)
SKIPF(PL10IN,1,17)
COPYL(PL10IN,SRTOBJ,PL10)    PUT FILE TWO ON NEWPL
REWIND,SRTOBJ.
COPYL(PL10IN,SRTOBJ,SRTOBJ2)
REWIND(SRTOBJ2)
COPYBF(SRTOBJ2,PL10)         PUT FILE THREE ON NEWPL
REWIND(SRTOBJ2)
LDSET(LIB=SYSIO/NUCLEUS)
LOAD(SRTOBJ2)          GENERATION OF SORT OVERLAYS
NOGO.
REWIND,SORTMRG.
COPYL(PL10IN,SORTMRG,PL10)   PUT FILE FOUR ON NEWPL
COPYBF(LGO,PL10)             PUT FILE FIVE ON NEWPL
UNLOAD(PL10,PL10IN)
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
SORT2,CM55000,T7000,MT1.
COMMENT. THIS JOB TAKES THE SORT/MERGE BINARIES FROM PL10 AND EDITLIBS
COMMENT. THEM INTO THE RUNNING SYSTEM. EITHER THE RELEASED VERSION OF
COMMENT. PL10 OR THE VERSION CREATED BY DECK SORT1 MAY BE USED.
LABEL(PL10,R,L=SORT4P0*3P4,D=HI)  MOUNT SORT/MERGE V4.0 PL
REWIND,PL10.
SKIPF(PL10,1,17)             SKIP OLDPL
COPYBF(PL10,SORTM)           GET BINARIES FOR MACRO SORTS
SKIPF(PL10,1,17)             SKIP RELOCATABLES FOR CONTROL CARD SORTS
COPYBR(PL10,SORTC,1)         0,0 OVERLAY
COPYBF(PL10,SORTO)           REST OF OVERLAYS
COPYBF(PL10,SMTXT)           SMTEXT
REWIND,SMTXT.
REWIND,PL10,SORTM,SORTC,SORTO.
UNLOAD(PL10)
EDITLIB(SYSTEM)
COMMENT. *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(*,SORTC,AL=3,FL=60000,FLO=1)
REPLACE(*,SMTXT,AL=0)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(*,SORTC,AL=0)
FINISH.
LIBRARY(COBOL,OLD)
REPLACE(*,SORTM)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

Dayfile output for the verification programs is as follows:

```
             SCCFE 3.4
05.44.51.VSCRT31
05.44.52.VSCRT1,CM65000,T500.
05.44.52.  CIRECTIVE FORMAT 1 VERIFICATION DECK -
05.44.52. SORT/MERGE 4.0
05.44.52.  JOB SORTS 10 RECORDS - ASCENDING ORDER
05.44.52.FILE(INPUT ,RT=Z,BT=C,FL=80,ERL=1)
05.44.54.FILE(OUTPUT,RT=Z,BT=C,FL=80,ERL=1)
05.44.57.SORTMRG(7C)
05.45.40.     1.475 RT SECONDS LOAD TIME
05.47.05. ** INSERTIONS DURING INPUT    *********0
05.47.05. ** DELETIONS DURING INPUT     *********0
05.47.05. ** TOTAL RECORDS SORTED       ********10
05.47.05. ** INSERTIONS DURING OUTPUT   *********0
05.47.05. ** DELETIONS DURING OUTPUT    *********0
05.47.05. ** TOTAL RECORDS OUTPUT       ********10
05.47.05. **END SORT RUN
05.47.06.CPA       .254 SEC.
05.47.06.PP      20.724 SEC.
```

```
             SCCFE 3.4
05.44.36.VSORT32
05.44.37.VSCRT2,CM65000,T500.
05.44.37.  CIRECTIVE FORMAT 2 VERIFICATION DECK -
05.44.37. SCRT/MERGE 4.0
05.44.37.  JOE SORTS 10 RECORDS - ASCENDING ORDER
05.44.37.SORTMRG(6C)
05.45.25.     1.636 RT SECONDS LOAD TIME
05.45.42. ** INSERTIONS DURING INPUT    *********0
05.45.42. ** DELETIONS DURING INPUT     *********0
05.45.42. ** TOTAL RECORDS SORTED       ********10
05.45.42. ** INSERTIONS DURING OUTPUT   *********0
05.45.42. ** DELETIONS DURING OUTPUT    *********0
05.45.42. ** TOTAL RECORDS OUTPUT       ********10
05.45.42. **END SORT RUN
05.45.43.CPA       .234 SEC.
05.45.43.PP      16.522 SEC.
05.45.43.IO        .281 SEC.
```

```
            SCOPE 3.4
05.44.50.VSCRT33
05.44.50.VSCRT3,CM65000,T500.
05.44.50.  MACRO SORT VERIFICATION DECK -
05.44.50. SORT/MERGE 4.0
05.44.50.  JOB SORTS 10 RECORDS - ASCENDING ORDER
05.44.50.COMPASS(S=SMTEXT,S=IOTEXT)
05.44.57. ASSEMBLY COMPLETE.   52700B SCM USED.
05.44.57.    1.333  CPU SECONDS  ASSEMBLY TIME.
05.44.57.LDSET(LIB=COBOL/SYSIO)
05.44.58.LGO.
05.45.06.      8.113 RT SECONDS LOAD TIME
05.45.06.
05.45.06.
05.45.08. ** INSERTIONS DURING INPUT    *********0
05.45.08. ** DELETIONS DURING INPUT     *********0
05.45.08. ** TOTAL RECORDS SORTED       ********10
05.45.08. ** INSERTIONS DURING OUTPUT   *********0
05.45.08. ** DELETIONS DURING OUTPUT    *********0
05.45.08. ** TOTAL RECORDS OUTPUT       ********10
05.45.08. **END SORT RUN
05.45.08.CPA      2.682 SEC.
05.45.08.PP       6.389 SEC.
```

RELEASE DESCRIPTION

QUERY UPDATE (QU) Version 1.0 is contained in the SCOPE 3.4 product set.

RELEASE MATERIALS

QU 1.0 is released on the program library tape PL11. QUIDDL is also included on this tape.

The structure of the release format PL11 is as follows:

Files 1-3	QU, as follows
1	QU program library
2	QU binary - absolute format
3	QU binary - relocatable format
Files 4-6	QUIDDL, as follows
4	QUIDDL program library
5	QUIDDL binary - absolute format
6	QUIDDL binary - relocatable format
Files 7-13	Installation Decks
7,8	QU installation decks
9,10,11	QU, QUIDDL verification decks
12,13	QUIDDL installation decks

HARDWARE CONFIGURATION

QU requires the same minimum hardware configuration as SCOPE.

GENERAL DESCRIPTION

The user, either at a terminal or through batch processing, can insert or delete records in a file, modify and display fields within records, specify selection criteria for record manipulation, and define various modes for operation and input/output options.

Data to be manipulated must be on a mass storage device; it may be organized as sequential, indexed sequential or direct access. Data within a file is referred to by symbolic names which have been recorded in a mass storage directory constructed through use of QUIDDL.

QU is written primarily in SYMPL, but it has several COMPASS routines.

INSTALLATION OPTIONS

The installation of Query Update Version 1.0 does not require selection or modification of any installation parameters.

INSTALLATION PROCEDURES

To obtain the QU installation decks, execute a program of the type:

```
Job card.
REQUEST(PL11,E)
SKIPF(PL11,6,17)               SKIP QU AND QUIDDL PL AND BINARIES
COPYBF(PL11,PUNCH,2)
COPYBF(PL11,PUNCH,3)           PUNCH VERIFICATION JOBS
UNLOAD(PL11)
6/7/8/9
```

Deck QU1 allows regeneration of PL11 to incorporate changes to the program library file which will be reflected in the binary. Successful formation of the absolute overlay may result in externals unsatisfied to the different access methods. This is of no consequence if such access methods are not to be used. Access methods IS 2.0, DA 1.0 should be present in the running system to satisfy externals when the NOGO directive of the deck QU1 is honored. Deck QU2 adds binary from either the released PL11 or a tape created by QU1 to the running system. Once QU2 has completed, job SCOPE3 (section 1) can be used to create a deadstart tape from the running system.

```
QU1,CM64000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF Q/U
COMMENT. AND COPIES THE OLDPL AND BINARY OF QUIDDL TO THE NEW PL11.
COMMENT. THE NEW PL11 WILL CONSIST OF SIX FILES
COMMENT. THE FIRST FILE OF PL11 WILL BE THE QU NEWPL
COMMENT. THE SECOND FILE WILL BE THE ABSOLUTE BINARY OF QU
COMMENT. THE THIRD FILE WILL BE THE RELOCATABLE BINARY OF THE OVERLAY QU
COMMENT. THE LAST THREE FILES WILL BE QUIDDL
LABEL(PL11IN,R,L=QU*QUIDDL*3P4,D=HI)    MOUNT QU PL
REQUEST,PL11,N,HI.              SCRATCH FOR NEW PL11
LABEL(PL11,W,L=QU*QUIDDL*3P4,D=HI)
REWIND(PL11IN,PL11)
UPDATE(F,P=PL11IN,N=PL11)
COMPASS(I=COMPILE,B=QX,L=0,S=IOTEXT,S=IPTEXT)
SYMPL(I=COMPILE,L=0,B=QX)
COMPASS(I=COMPILE,B=QX,L=0)
SYMPL(I=COMPILE,L=0,B=QX)
COMPASS(I=COMPILE,B=QX,L=0)
SYMPL(I=COMPILE,L=0,B=QX)
REWIND(QX)
LDSET(LIB=SYSIO/NUCLEUS)
LOAD(QX)            GENERATION OF OVERLAY QU
NOGO.
REWIND(QU,QX)
SKIPF(PL11,1,17)
COPYBF(QU,PL11)                 COPY BINARY FILE TO NEW PL11
COPYBF(QX,PL11)
REWIND(PL11IN)
SKIPF(PL11IN,3,17)
COPYBF(PL11IN,PL11,3)       COPY QUIDDL TO TAPE
UNLOAD(PL11IN,PL11)
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
QU2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS THE QUERY/UPDATE BINARIES INTO THE RUNNING
COMMENT. SYSTEM FROM THE PL11 TAPE. EITHER THE RELEASED VERSION OF PL11
COMMENT. OR THE VERSION CREATED BY DECK QU1 MAY BE USED.
LABEL(PL11,R,L=QU*QUIDDL*3P4,D=HI)  MOUNT QU PL
REWIND,PL11.
SKIPF(PL11,1,17)            SKIP OLDPL
COPYBF(PL11,QU)             Q/U BINARIES
REWIND,PL11,QU.
UNLOAD(PL11)
EDITLIB(SYSTEM)
COMMENT. *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(*,QU,AL=3,FL=64000,FLO=1)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAMS

The QU verification deck consists of three jobs which use the SCOPE Job Dependency option to control the order of execution. Since these are the same verification decks used by QUIDDL, QUIDDL must also be installed in the system before running the verification program decks. The first job (SCHEMA) creates a directory; the second one (AREA), an indexed sequential file; the third (QU) runs Query Update using the directory and the indexed sequential file.

```
            SCOPE 3.4
05.44.52.SCHEM35
05.44.52.SCHEMA,T100,CM63000,DQU00.
05.44.52.REQUEST(TAPE5,*PF)
05.44.54.QUIDDL(M=C,S=TAPE5)
05.45.12.     3.082 RT SECONDS LOAD TIME
05.45.15.( TAPE5   ASSIGNED TO EST 07 )
05.45.47.QUIDDL COMPLETE
05.45.48.CATALOG(TAPE5,DDLDIRECT,ID=VERIFY,CY=1,C
05.45.48.N=*---*,PW=*---*)
05.45.48.INITIAL CATALOG
05.45.50.RP = 010 DAYS
05.45.51.RETURN(TAPE5)
05.45.52.TRANSF(QU)
05.45.53.     TRANSFERRED TO QU00036
05.45.53.FTN(L=0)
05.46.54.     2.055 RT SECONDS LOAD TIME
05.46.59.      .243 CP SECONDS COMPILATION TIME
05.46.59.COMPASS(L=0)
05.47.03. ASSEMBLY COMPLETE.   46600B SCM USED.
05.47.03.    1.427  CPU SECONDS  ASSEMBLY TIME.
05.47.03.MAP(OFF)
05.47.04.ATTACH(TAPE5,DDLDIRECT,ID=VERIFY)
05.47.06.PF CYCLE NO. = 001
05.47.06.LGO.
05.47.34.     3.710 RT SECONDS LOAD TIME
05.47.36.    STOP
05.47.38.CPA      3.646 SEC.
05.47.38.PP      30.040 SEC.
```

```
            SCCPE 3.4
05.44.51.AREA034
05.44.51.AREA,T100,CM63000,DQU00.
05.44.51.REQUEST(ALICE,*PF)
05.44.52.FTN(L=0)
05.44.54.      1.305 RT SECONDS LOAC TIME
05.44.58.       .186 CP SECONDS COMPILATION TIME
05.44.58.COMPASS(S=IOTEXT,L=0)
05.45.04. ASSEMBLY COMPLETE.   46600B SCM USED.
05.45.04.     1.496  CPU SECONDS  ASSEMBLY TIME.
05.45.05.MAP(OFF)
05.45.05.LGO.
05.45.13.      7.547 RT SECONDS LOAD TIME
05.45.13.( ALICE   ASSIGNED TO EST 10 )
05.45.16.FTN - FATAL ERROR   0065
05.45.18.JOB REFRIEVED
05.45.18. SIS ERROR RECOVERY
05.45.18.FOLLOWING FILES TO BE CLOSED BY SIS
05.45.18. RECOVERY-
05.45.18.           ALICE
05.45.20. (PREVIOUS ERROR CONDITION RESET)
05.45.21.EXIT.
05.45.21.TRANSF(QU)
05.45.23.      TRANSFERRED TO QU00036
05.45.23.CPA      3.142 SEC.
05.45.23.PP      15.882 SEC.


            SCOPE 3.4
05.45.54.QU00036
05.45.55.QU,T200,CM77000,DQU02.
05.45.55.QU.
05.46.07.     11.183 RT SECONDS LOAC TIME
05.46.17. QUERY UPDATE SESSION COMPLETED
05.46.18.REWIND(F1,F2,F3)
05.46.18.COPYSEF(F1,OUTPUT)
05.46.19.COPYSBF(F2,OUTPUT)
05.46.20.COPYSEF(F3,OUTPUT)
05.46.20.REWIND(F3)
05.46.21.QU.
05.46.32.      9.336 RT SECONDS LOAD TIME
05.46.41. QUERY UPDATE SESSION COMPLETED
05.46.41.CPA      2.816 SEC.
05.46.41.PP      22.432 SEC.
```

GENERAL DESCRIPTION

With QUIDDL, a data base administrator defines data in terms of its attributes and the relationships that exist and must be maintained, thus permitting users to store and retrieve data from secondary storage easily and efficiently. The data base consists of all records and areas in mass storage; they are described by a specific schema. The user, in combination with SCOPE 3.4 features, has the ability to:

Structure data physically in the manner most suitable to each application regardless of whether any portion of that data may be used for some other application as well.

Control the physical placement of data.

Declare a variety of data structures.

Interact with data without concern for the mechanics of maintaining data structural associations.

QUIDDL has routines written in SYMPL and COMPASS.

RELEASE DESCRIPTION

QUIDDL Version 1.0 runs under the SCOPE 3.4 operating system; it operates on the same minimum hardware configuration as SCOPE 3.4.

RELEASE MATERIALS

The Update program library for QUIDDL shares release tape PL11 with QU.

A complete catalog of PL11 contents is included in section I-12.

LIMITATIONS

A directory cannot be compressed to eliminate wasted space if UPDATE mode is used.

INSTALLATION PARAMETERS

QUIDDL 1.0 will reference IP.CSET of IPTEXT in the running system at installation time to determine the exact character set.

INSTALLATION PROCEDURES

QUIDDL shares PL11 with QU. The installation decks refer to the release QUIDDL tape as P11A.

To obtain the QUIDDL installation decks execute a program of the type:

```
Job card.
REQUEST(PL11,E)
SKIPF(PL11,11,17)
COPYBF(PL11,PUNCH,2)
UNLOAD(PL11)
6/7/8/9
```

Deck QUDDL1 allows regeneration of the release tape incorporating changes to the program library file which will be reflected in the binary. Deck QUDDL2 adds binary from either the released tape or a tape created by QUDDL1 to the running system. Once QUDDL2 has completed, job SCOPE3 (section 1) can be used to create a deadstart tape from the running system.

```
LOAD(LGO)                GENERATION OF OVERLAY QUIDDL                  023
NOGO.                                                                  024
REWIND(QUIDDL)                                                         025
REWIND(LGO)                                                            026
SKIPF(F11A,4,17)                                                       027
COPYBF(QUIDDL,F11A)                                                    028
COPYBF(LGO,F11A)                                                       029
UNLOAD(P11A)                                                           030
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE                                               032
7/8/9  END OF RECORD
6/7/8/9  END OF FILE


QUDDL2,CM55000,T7000,MT1.                                              001
COMMENT. THIS JOB TAKES TAKES THE RELEASED VERSION OF QUIDDL, OR THE    002
COMMENT. VERSION CREATED BY JOB QUDDL1, AND EDITLIBS THE BINARY INTO   003
COMMENT. THE RUNNING SYSTEM FROM F11A.                                 004
LABEL(F11A,R,L=QU*QUIDDL*3P4,D=HI) MOUNT QUIDDL PL                     005
REWIND,P11A.                                                           006
SKIPF(F11A,4,17)                                                       007
COPYBR(P11A,QUIDC,1)          QUIDDL CONTROL CARD CALLABLE ROUTINE     008
COPYBF(P11A,QUIDO)            QUIDDL OVERLAYS                          009
REWIND,P11A,QUIDC,QUIDO.                                               010
UNLOAD(P11A)                                                           011
EDITLIB(SYSTEM)                                                        012
COMMENT. *** END OF JOB ***                                            013
7/8/9  END OF RECORD
READY(SYSTEM)                                                          015
LIBRARY(NUCLEUS,OLD)                                                   016
REPLACE(*,QUIDC,AL=3,FL=44000,FLO=1)                                   017
FINISH.                                                                018
LIBRARY(SYSOVL,OLD)                                                    019
REPLACE(*,QUIDC,AL=0)                                                  020
FINISH.                                                                021
COMPLETE.                                                              022
ENDRUN.                                                                023
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

The QUIDDL verification decks are shared with QU. See the QU section of this document for description of the verification process.

RELEASE DESCRIPTION

INTERCOM 4.1 in conjunction with the SCOPE 3.4 operating system provides Teletype and CRT terminals with time-shared access to CYBER 70 Series computers. Remote batch jobs may be submitted from terminals equipped with a remote card reader and printer, from a remote computer (1700 or 8231) running an IMPORT package, or from a High Speed Batch Terminal. Programs written in the RUN, FORTRAN Extended, COBOL, ALGOL, COMPASS, and BASIC languages can be submitted from a remote terminal for execution at control points; the user at the remote terminal can interact with the executing program. Program output can be routed to the line printer and card punch at the central site or to a terminal equipped with line printers or card readers. Through the SCOPE permanent file feature, input from a central site magnetic tape or card reader is available to the remote user.

INTERCOM 4.1 further provides the capability for multi-user jobs to be written which can handle many users simultaneously with one copy of a program. The program-text editor, EDITOR, introduced in Version 3.0, utilizes this capability.

HARDWARE CONFIGURATION

In addition to the minimum hardware required by the SCOPE system, INTERCOM 4.1 requires the following equipment for communication and operation.

One of the following:

CRT terminal, model 214-11, 214-12, 217-11, 217-12, 217-13, 217-14, or 711-102,
or a model 33 or 35 KSR or ASR Teletype terminal, or a 713 Teletype compatible terminal,
or a 1700 or 8231 remote computer running an IMPORT package
or a 733-10 High Speed Batch Terminal (HSBT)

Also: A dedicated multiplexer on a dedicated channel 6671, 6676, 6673, or 6674 (6671 for Teletype and/or CRT terminals; 6676 for Teletype terminals only; 6673 or 6674 for high speed connections to remote computer)
Or: A dedicated 7077 Communications Station on a dedicated channel with a 791 Local Communications Controller (LCC) for Teletypes and HSBTs.

Also: Data Sets for communication between the remote terminal and central site: Teletype terminals require 103A Data Sets; CRT terminals require 201A or 201B Data Sets, or CONTROL DATA 358 Transceivers; remote computers and HSBTs require 301B or 303 Data Sets and a TELPAK A communication line, or CONTROL DATA 358 Transceivers.

Hardware Options

Teletype
- Paper tape reader/punch

CRT
- Card reader: 224-11, 12, 13, or 14
- Line printer: 222-11, 12, 13, or 14

HSBT
- Card reader (one additional) 733-120
- Line printer (up to three additional) 733-110
- Card punch 733-101
- Memory increment (if additional peripherals are used beyond the basic HSBT required options, single card reader and line printer) 733-140
- Data Set Adaptor 733-130
- CRT (16x80 or 18x64) 733-150 or 733-152

The High Speed Import for the 8231 and for 1700 are discussed in separate sections of this Installation Handbook and hardware options are included in those sections.

RELEASE MATERIALS

INTERCOM Version 4.1 release material consists of a magnetic tape (PL12) containing the INTERCOM Version 4.1 OLDPL as file one and installation decks as files two and three.

MODIFICATIONS AND DEFICIENCIES

New Features and Modifications

INTERCOM Version 4.1 includes all the interactive commands of INTERCOM Version 3.0 with the exception of SETUP, the filename facility (available through XEQ), and B. All facilities offered by SETUP are available through the multi-user editor features of EDITOR.

The remote batch commands of INTERCOM Version 4.1 are changed as follows:

New Commands	Description
REP	Repeat the printing of a file
KILL	Kill a job in execution or in the input queue
RTN	Return the file now printing to the output queue

New Names	Old Names
REW	AGAIN
BSP	AGAIN,N
DIVERT	OUTPUT

The XEQ command, added in INTERCOM Version 4.1, makes available to the user many of the features of the new LOADER for SCOPE 3.4. It also adds greater consistency in the definition of an INTERCOM language.

The BRESEQ command, added in INTERCOM Version 4.1, allows the user to resequence a BASIC program and gives him control over how the file is resequenced.

Through the new INTERCOM PAUSE capability, a terminal user can improve control of the interactive program at the terminal.

The 1WB driver is capable of supporting the 6673 and 6674 multiplexers and remote 1700 and 8231 computers.

The 1ZZ driver in INTERCOM Version 4.1 is capable of supporting the LCC and High Speed Batch Terminals and Teletypes.

Teletype operation via both the 791 LCC and the 6671 or 6676 multiplexer differs from INTERCOM 3.0 as follows:

> The left arrow (←) no longer is used for backspacing. The backspace character (CTRL H) must be used.
>
> Logical lines longer than 72 characters can be entered via paper tape by terminating continued lines with a LINE FEED as the first control character (LINE FEED, RETURN) and terminating the last physical line with a carriage return as the first control character (RETURN, LINE FEED).
>
> Teletype code has been changed as follows:

Graphic	INTERCOM 3.0 CDC 63 Character Display Code	INTERCOM 3.0 CDC 64 Character Display Code	INTERCOM 3.0 ASCII 64 Character Display Code	INTERCOM 4.1 63 Character Display Code	INTERCOM 4.1 64 Character Display Code
#	65	65	60	60	60
' (apostrophe)	64	64	61	70	70
! (exclamation)	71	71	62	66	66
%	76	63	63	none	63
" (quote)	60	60	64	64	64
_ (underline) or ←	none	76	65	65	65
]	62	62	66	62	62
@	66	66	70	74	74
?	75	75	71	71	71
[	61	61	72	61	61
<	72	72	74	72	72
\	74	74	75	75	75
^ (circumflex) or ↑ (on some TTYs)	70	70	76	76	76
:	63	00	00	63	00

Corrections

All eligible PSR code for INTERCOM Version 2.0 or Version 3.0 published through PSR Summary 320 has been incorporated in Version 4.1.

Limitations

1. EDITOR commands input through the terminal cannot exceed 518 characters; the longest text line allowed by EDITOR is 510 characters. On the Teletype, data input to a user's interactive program can be of any length (subject to the limitations imposed by the size of the user's circular buffer); and it may be a full screen on the CRT.

2. The following SCOPE commands are not allowed as direct INTERCOM commands, although they are allowed as part of a job created through EDITOR to be submitted subsequently for batch execution. REQUEST is allowed only for a permanent file device:

   CKP, RFL, DMPECS, RESTART, LIMIT, LOAD, LIBLOAD, SLOAD, CMLOAD, ECSLOAD, EXECUTE, NOGO, LIBRARY, RPACK, LABEL, and VSN.

   These commands should be given an access level of 7777B when a deadstart tape is made.

3. When the CONNECT command (or CONNEC call) is used, the specified data is routed to or from the terminal each time the file is read or written. When simultaneous operations are to be performed, no more than one file should be connected to a terminal for interactive operations at any given time.

4. Time must be entered into the SCOPE system prior to bringing up INTERCOM. INTERCOM will request a time entry if it has not been done.

5. PASSWRD can be run while INTERCOM is up; normally, however, it should not be done as INTERCOM users may experience poor response to LOGIN, SEND, SITUATE while PASSWRD is running.

6. A single copy of the PP multiplexer driver cannot service both BCD and ANSI 200 User Terminals.

7. Teletype operation via the 791 LCC differs from Teletype operation via the 6671 or 6676 multiplexer.

   Teletype interrupts (CTRL Z followed by the directive A or S) require a carriage return (RETURN) to complete the message before the interrupt is processed. If the continue message (CTRL Z RETURN) is input, interrupted output will continue at the next line; characters in the interrupted line will be lost.

   The delete line character (CTRL X) is recognized on paper tape.

   Physical lines longer than the width of the Teletype carriage cannot be entered. The message LINE TOO LONG will be issued and the user must re-type the complete line.

8. Paper tapes created under INTERCOM Version 3.0 will not run properly under Version 4.1.

9. Pending a modification to 7000 SCOPE 2.0, which should result from PSR S00848, the Q command does not distinguish between a parameter of the form 6x and a parameter of the form 7x.

   The information returned for either form will be the same, and will be dependent on whether or not the 6000 running INTERCOM is the 7000 operator console as well. If it is, 6x and 7x will both perform as 7x is documented. Otherwise, they will both perform as 6x.

GENERAL PROCEDURES

Installation of a complete INTERCOM system requires establishing installation parameters and installing from the INTERCOM OLDPL. The card deck described later can be run at the central site to install INTERCOM. FORTRAN Extended 4.0 and COMPASS 3.0 must be installed before INTERCOM 4.1 can be installed.

CENTRAL MEMORY REQUIREMENTS

The standard installation of INTERCOM Version 4.1 requires about 2000 (decimal) CM/ECS locations for PP programs 1CI, 1BR, 1QP, MUJ, 3TT and all the 3TT and 1CI overlays (except 3CU,3CX). In addition, the PP routine 1PT may be made CM/ECS resident to increase performance; 1PT is on periodic recall at control point zero. If the system has a heavy high speed remote batch utilization, 1XP and its overlays should be made CM or ECS resident. The LCC driver is not dedicated and enters timed recall at control point zero if activity is low. Therefore, 1ZZ and 9ZZ may be made CM/FCS resident as well as the HSBT remote batch processor 1LX and its overlays. These additions increase the total CM or ECS requirement to about 4500 locations.

Whenever INTERCOM is in idle state, approximately 500 additional CM words are required for multiplexer tables and minimum empty buffer chains.

INSTALLATION PARAMETERS

To configure the INTERCOM system for a particular installation:

An equipment status table (EST) entry must be established for each multiplexer dedicated to INTERCOM.

In CMR, a multiplexer table must be defined which contains subtables for each multiplexer dedicated to INTERCOM.

The installation deck must contain an assembly for each variant of the low speed multiplexer driver required.

Parameters in the INTERCOM common deck INTCOM may be changed to affect the characteristics of INTERCOM.

Parameters in the EDITOR common decks IPFTN and IPCOM may be changed to affect the characteristics of EDITOR.

Parameters in the multiuser job common decks MUJCOM and CMUJCOM may be changed to affect the characteristics of multiuser jobs (particularly EDITOR).

INTERCOM COMMON DECK SETTINGS

Release values are shown in the following list of INTERCOM parameters for the common deck INTCOM present on PL12. If these parameters are to be changed, the cards containing the proper code with the CEQU macro should be placed after an *INSERT INTCOM.91 card and inserted into the first update record of the deck INTCM1 after card 035. Alternate tested values are shown in parentheses.

IP.AABT CEQU 76B

Display code for the character recognized by the multiplexer driver as an abort request. This value is for the 63 character set, and a 63B will be assigned for the 64 character set when it is selected.

IP.CTCT CEQU 20

Maximum number of 200 User Terminals at 2400 baud which the PPU multiplexer driver will attempt to service in one cycle. If this number is set too high and the 200 UTs are very active, an attempt may be made to service too many users in one cycle and retransmissions may be observed. If this number is set too low and more than this number of terminals are active, the polling rate at some terminals may be slowed noticeably. In general this number should not require change.

The maximum absolute value which may be defined is 32; 32 is the maximum number of 6671 ports supported by one low speed driver.

IP.CTCT4 CEQU 8

Maximum number of 200 User Terminals at 2400 or 4800 baud which the PPU multiplexer driver will attempt to service in one cycle if at least one 4800 baud terminal is defined for this driver. The effect of this parameter is similar to IP.CTCT; it also should not require change. The maximum value stated for IP.CTCT is applicable here also.

IP.HRCL CEQU 5

Number of seconds delay (when 1WB is in recall) before taking over a PP to check if any terminals are attempting to establish communications. It represents, approximately, the maximum amount of time a user will wait after loading IMPORT before communication is established.

IP.HSYNC   CEQU   7500B

Number of communication cycles 1WB will wait for a sync acknowledge from terminal before causing error indication and retransmission. The value of this parameter will vary with the length of the longest transmission line attached to the highspeed multiplexer.

The duration of a communication cycle is approximately 250 microseconds. The release value should be large enough to provide for the longest transmission line possible in the continental United States (3500 miles).

IP.HINFW   CEQU   7500B

Number of communication cycles 1WB will wait for first input word from terminal before causing error indication and retransmission. The value of this parameter will vary with the length of the largest transmission line attached to the high speed multiplexer. The value of this parameter is similar to IP.HSYNC and will vary accordingly.

IP.HCRXT   CEQU   100

Number of consecutive retransmissions allowed a given terminal before ending communication with the terminal and attempting to restart from the initiate-communication phase.

A single transmission generally requires between 10 and 250 milliseconds.

IP.1LX   CEQU   1

If 1, one copy of 1LX will be called to process all LCC users. If 2, one copy of 1LX will be called for each LCC.

IP.PRIX   CEQU   3777B   (7000B)

Non-zero, indicates the priority given to input files read from remote site. If zero, priority will be taken from Job card.

IP.IACES CEQU 12

Number of bits in the 11-bit user table access field assigned for the user access level. The remainder will be used for permission bits. Both access level and permission bits are used to determine if a user has access to a specific utility or routine.

When a request is made to use a command, the user's access level is checked to determine whether it is greater than or equal to the access level of the command. If it is, the permission bits of the user are compared against the permission bits of the command. If they match, the user is given access to the command. If either test fails, permission is not granted.

Example:

In a university environment all undergraduates are assigned access Level 5; therefore, all commands having an access level of 5 or less are available to the undergraduates. Only commands X, Y, and Z however, are available to members of the Computer Science Department. This may be controlled by assigning a portion of the access field to the access level with the remainder used for permission bits. Then such commands could be assigned a permission value of 3, which also would be assigned only to members of the Computer Science Department.

IP.ID CEQU 1

If one, the INTERCOM user id is used as the default permanent file id by the commands STORE, FETCH, and DISCARD. If zero, the permanent file id must be specified by the INTERCOM user.

IP.IDFL CEQU 55000B

Default field length assigned to a user's program, or any control card when field length override is specified in the library directory, and user has not entered a field length (EFL).

IP.IGCON CEQU 0

Maximum number of 1700/274 Graphics consoles known to the system; should be set to zero if Graphics is not defined in the system. IP.IGCON has a maximum possible value of 24.

IP.IGS CEQU 0 (1)

If one 1700/274 Graphics is defined as existing within the system; if zero 1700/274 Graphics is not present in the system.

IP.IHEAD CMICRO 0,(CONTROL DATA INTERCOM 4.1)

Header output by 1PT when a remote terminal dials into the INTERCOM system.

IP.ISCRN CEQU 1

Specifies the default size of CRT screens at the installation: if 0, a 13 x 80 screen is assumed; if 1, a 50 x 20 screen.

IP.MALOC CEQU 4000B

A 12-bit octal value defining the allocation style for files created by a multi-user job. Bit 11 always is set to one to indicate that a permanent file device is requested. The bits indicating the allocation style are bits 5-0. This value is placed in the File Name Table entry generated for new multi-user job files, in byte C.FALLOC.

IP.MPRIT CEQU 4000B

Priority of output files directed to the central site by the remote batch command DIVERT.

IP.N6671 CEQU 2

Number of 6671 multiplexers for which the driver is assembled. 0,1, or 2. No individual driver can support more than 2 multiplexers total.

IP.N6676 CEQU 0

Number of 6676 multiplexers for which the driver is assembled. 0,1, or 2. If set to 2, IP.N6671 must be set to zero. Refer to the section on the Installation of INTERCOM for a further discussion on the required settings for these two parameters.

IP.N791 CEQU 3

Maximum number of LCC's per channel that the LCC driver can support: 1, 2 or 3.

IP.TILL CEQU 55B

An illegal character input from Teletype will be converted to IP.TILL.

IP.TSL CEQU 10B

Default time limit in seconds for execution of a user's program, if the user has not entered a time limit (ETL).

IP.1PT CEQU 1

1PT will not be called if this parameter is set to zero; if set to 1, 1PT will be called only if CRT terminals are configured. If set to 2, 1PT will always be called when low speed multiplexer drivers are called.

CROSS REFERENCE LISTING

The following cross-reference listing shows the routines that reference each INTCOM symbol:

*********..	LOGIN								
INTCOM									
*********..									
IP.AABT	8ZM	9ZM							
IP.CTCT	1ZM								
IP.CTCT4	1ZM	8ZM							
IP.HRCL	1WB								
IP.HSYNC	1WB								
IP.HINFW	1WB								
IP.HCRXT	1WB								
IP.1LX	1LX	1I1							
IP.PRIX	1XP	1LX	1PT						
IP.IACES	1CI								
IP.ID	STORBEG	DISBEG	FETBEG						
IP.IDFL	1CI								
IP.IGCON	1I1	GBJ	GEJ	1GJ	2GJ	1GR			
IP.IGS	1XP	1CI	1I1	GBJ	GEJ	1GJ	2GJ	1GR	
IP.IHEAD									
IP.ISCRN	1CI								
IP.MALOC	FAD								
IP.MPRIT	1XP	1LX	1CI	1DS	TBL				
IP.N6671	1ZM	8ZM	9ZM						
IP.N6676	1ZM	8ZM	9ZM						
IP.N791	1ZM	8ZM	9ZM	1XP	1LX	1WB	0ZZ	1ZZ	1CI	1[illegible]
	3TT	1BR	1QP	1QM	1PT	1ID	1IM	1DS	T76	TBL
	FNT	IUP	IAP	MES	MUJ	MAC	FAD	GBJ	GEJ	1GJ
	2GJ	1GR	IUID	STORBEG	DISBEG	FETBEG				
IP.TILL	8ZM								
IP.TSL	1CI								
IP.1PT	1I1								

SCOPE IPARAMS SETTINGS

These parameters must be set at *INSERT IPARAMS.15 when SCOPE 3.4 is installed (deck SCOPE1).

IP.ILCMD CEQU 1

If set to 1, the last word in the user table (W.IINS reserved for the installation) will store the last command entered by each user for display on the DSD Q display. If 0, it will not be used for this purpose.

IP.IUSID CEQU 2RAH

Defines the first user id available for assignment by the program PASSWRD. The value of this parameter is determined by the number of highspeed multiplexers with sub-tables defined in the system and the number of HSBT terminals defined in the system. The highspeed multiplexers use two id's per 6673 or 4 id's per 6674, starting with user id AB. The LCC's use one id per HSBT line defined in the LCC mux subtables.

This user ID is the lowest available to be assigned a user at a lowspeed terminal. Every 1700 or 8231 remote highspeed batch terminal connected to the system must have its own user ID assigned to it.

IP.1M1 CEQU 3

Maximum number of active low speed multiplexer drivers allowed in the system simultaneously. Should be zero if the system has no 6671 or 6676 multiplexers. The setting never should exceed three. This entry is dependent on the number of channels configured for use by low speed multiplexer drivers. One copy of a driver may service only one channel.

IP.1WB CEQU 2

Maximum number of active highspeed multiplexer drivers allowed in the system simultaneously. Should be zero if the system has no 6673 or 6674 multiplexers. The setting never should exceed two; it is determined in the same manner as IP.1M1.

IP.1ZZ CEQU 0

Maximum number of active LCC drivers allowed in the system simultaneously. Should be zero if system has none; it should never exceed two.

EST ENTRY

The EST table established when deck SCOPE1 is run to install SCOPE3.4, must contain an entry for each multiplexer dedicated to INTERCOM. The channel referenced in this entry must be dedicated to the INTERCOM multiplexers on that channel. A separate copy of the driver in a dedicated PP is required to service each channel assigned to INTERCOM. For non-allocatable equipment, the EST uses the 6000 macro which has been modified as follows:

```
type 6000  channel,sync,unit,onoff,ipoint
```

Macro parameters used by INTERCOM:

type	DC for 6671, YC for 6676, SC for 6673 or 6674, CS for 791
channel	Channel for multiplexer or 7077 Communication Station
sync	Equipment number for multiplexer or 7077 SAC/CSM I/O channel for 791
unit	Not used by INTERCOM
onoff	Switch, on = 0, off = 1
ipoint	Index to INTERCOM multiplexer table

A typical EST entry might appear as follows:

```
*I  CMR.2108

DC  6000  3,5,0,0,MUX1-T.ITABL
```

This entry notifies the multiplexer driver that a 6671 with equipment number 5 is on channel 3; and the index to the multiplexer table is MUX1-T.ITABL, where MUX1 is the symbol on the card defining the multiplexer subtable for this 6671, and T.ITABL is the beginning of the multiplexer table.

Typical EST entries for two LCC 791's on SAC/CSM channels 0 and 1 connected to a 7077 connected to channel 4 would appear as follows:

```
CS   6000      4,0,0,0,MUX2-T.ITABL
CS   6000      4,1,0,0,MUX3-T.ITABL
```

CONFIGURATION PARAMETERS (INTERNAL TO CMR)

This parameter defines the length of the INTERCOM multiplexer table. It must be set at *INSERT CMR.964 when SCOPE 3.4 is installed. The default value is:

```
L.ITABL  CEQU  19
```

This parameter should be changed to reflect the size of the multiplexer table for each installation. The length of the table can be determined from the following formula:

```
L.ITABL = 2+2*N76+N71+N71PORTS+N73+N74+NG+2*N91+N91PORTS
```

N76	number of 6676 multiplexers dedicated to INTERCOM
N71	number of 6671 multiplexers dedicated to INTERCOM
N71PORTS	total number of 6671 ports defined
N73	number of 6673 multiplexers dedicated to INTERCOM
N74	number of 6674 multiplexers dedicated to INTERCOM
NG	number of 6673 or 6674 multiplexers defined for GRAPHICS support
N91	number of 791's
N91PORTS	total number of 791 ports defined

CMR MULTIPLEXER TABLE

The CM resident INTERCOM multiplexer table is used by INTERCOM to provide data on the hardware configuration of the installation and to record parameters. It consists of two dedicated parameter words and one or more subtables assigned to the multiplexers serviced by INTERCOM.

The first two words of the multiplexer table, the parameter words, start at location T.ITABL in CMR. They are already assembled into CMR. The subtables follow the parameter words in any order convenient to the installation. The first subtable must be defined at *INSERT CMR.2238 when SCOPE 3.4 is installed. Each subtable has a relative pointer in the EST entry for that multiplexer. The upper bound of the multiplexer subtable may not extend beyond 7777B.

CMR MULTIPLEXER SUBTABLE GENERAL FORMAT

Each multiplexer subtable contains one entry to define the type of multiplexer, one entry for each port defined on that multiplexer if a 6671 or a 791, or a single entry if a 6676, or no further entries if a 6673 or 6674 does not support graphics consoles or a single entry if a 6673 or 6674 supports graphics consoles. The address of the entry describing the multiplexer is the same address used in the EST entry defining that multiplexer. A subtable for a 6671 multiplexer might be defined as follows:

```
MUX1     MUX71    4
         CRT
         CRT
         CRT
         TTY
```

When a 6671 multiplexer is configured it is advisable to place the highest speed terminals on the lowest ports and to place any empty ports at the high number port positions. Thus, the 6671 should be configured 4800 baud terminals first, the 2400 and 2000 baud terminals, then TTY's, then empty ports. The MUX71 macro port count parameter can be set to exclude the empty ports and increase driver efficiency. This saving is especially important when a driver is to support both a 6671 and a 6676.

A subtable for the 6673/6674 multiplexer might be defined as follows:

```
MUX4      MUX73        0, 3, G
```

The mux is defined with terminals attached to ports 0 and 3, and it also is described as capable of graphics support.

MULTIPLEXER DEFINITION ENTRIES

INTERCOM recognizes four types of multiplexers, the 6671, 6676, 6673 and 6674, and one type of communications subsystem, the 791 (LCC). They are defined with the following macros:

```
MUX71     no. of ports
MUX76     no. of ports
MUX73     P0,P1,G
MUX74     P0,P1,P2,P3,G
MUXLCC    No. of ports, memory size
```

The parameter (number of ports) indicates the highest numbered port + 1 which INTERCOM is to service. For the 6673 and 6674, the parameters Pi are the port numbers which are attached to the multiplexer. The parameter G, if specified, is the letter G and designates this multiplexer as being defined for Graphics support. The memory size parameter for the LCC indicates the size of the LCC memory. It should be either the characters 4k or 8k. If omitted 8k is assumed. A 6671 with ports 0, 1 and 3 attached to data sets should be defined as:

```
MUX71     4
```

A 791 with ports 0, 1, and 3 attached to data sets should be defined as:

```
MUXLCC    4
```

Since a 6676 multiplexer can have only TTY ports, the MUX76 macro does not require port definition entries. The MUX73, MUX74, and MUX76 macro generate all multiplexer subtable entries necessary to completely define the multiplexers.

PORT DEFINITION ENTRIES

Currently, six types of ports are recognized by the CMR macros for a 6671 multiplexer only. They are defined with the following macros:

TTY	Teletype Model 33 or 35 or CDC 713
CRT	BCD CRT or 200 User Terminal, 2400 baud
CRTA	ANSI CRT or 200 User Terminal or CDC 711 2400 baud
CRT4	BCD CRT or 200 User Terminal, 4800 baud
CRTA4	ANSI CRT or 200 User Terminal or CDC 711 4800 baud
EMPTY	EMPTY Port (not serviced by INTERCOM)

The port definition entries immediately follow the entry for the corresponding 6671 multiplexer. Each entry defines one port, beginning with port 0 as the first entry, the second is port 1, and so on. All ports through the highest to be serviced by INTERCOM on that multiplexer must be defined. Thus, if the number of ports parameter on the MUX71 macro is 10B, then 8 port definiton entries must follow even though some may not be used. Unused ports should be defined with the EMPTY macro.

Because the INTERCOM low speed multiplexer driver cannot handle both BCD and ANSI terminal types on the same channel, these terminal mixes should not be specified. Refer to the section on the installation of INTERCOM for allowable terminal combinations. Terminals to be operated at 4800 baud must operate in full duplex mode.

Port definition entry macros without parameters indicate that the ports are servicing normal dial-up telephone circuits. Site addresses for dial-up CRTs not on party lines are assumed to be zero by the INTERCOM system. They must be set manually to zero at the terminal. It is possible also to define hard-wired and party-line connections by adding parameters to any of the CRT definition macros as described below.

Five types of ports are recognized by the CMR macro for an LCC. They are defined with the following macros:

TTY	Teletype model 33 or 35 or CDC 713
LSBT	Low Speed Batch Terminal
MSBT	Medium Speed Batch Terminal
HSBT	High Speed Batch Terminal
EMPTY	Empty port (not processed by INTERCOM)

Note, however, that INTERCOM 4.1 has not been tested using Low, Medium, and/or High Speed Batch Terminals; therefore, these three terminal types are not supported with this initial release of INTERCOM 4.1.

PORT DISTRIBUTIONS FOR LOW SPEED MULTIPLEXERS

MULTIPLEXER CONFIGURATION	HARDWARE LIMITS		SOFTWARE LIMITS			
One PPU	200 UTs	TTYs	2400 Baud 200 UTs	TTYs	4800 Baud 200 UTs	TTYs
6671	16		16		16	
		16		16		16
2X6671	32		28-32		Not Supported	
	16	16	14-16	16	Not Supported	Not Supported
		32		32		32
6676		64		64		64
2X6676		128		100-128		100-128
6671, 6676	16	64 80	16	64 80	10-12	25-40 80

The chart indicates the hardware limitations for each low speed multiplexer configuration. These limitations show the maximum number of 200 UT ports that can be defined and, the maximum number of TTY ports that can be defined. Each 200 UT port can be a party-line port and support more than one 200 UT.

For each hardware configuration, the chart indicates the best estimates of what the software will support satisfactorily. Under heavy loads, terminals may suffer some degradation.

PARTY LINES

Any of the CRT definition macros may be used to define party-line configurations by adding parameters to the macro call. The general form is:

```
        CRT     sa0,sa1,...sa15
```

A list of site addresses indicates the port is to service a party-line to which terminals at those site addresses may be connected. Up to 16 site addresses, 0 to 17B, may be specified. The macro will stop scanning after the 16th parameter, or at the first null parameter. Site addresses may be specified in any order.

For example, a BCD 2400 baud CRT or 200 User Terminal party-line with six possible site addresses might be defined as follows:

```
        CRT  5,1,0,10,15,6
```

HARD-WIRED CONNECTIONS

Because of the way INTERCOM handles hard-wired (non dial-up) terminals, they must be specified as party-lines with only one site address (usually zero for only one terminal; however, the site address specified in the macro should match the site address of the terminal). A terminal connected to a 6671 with Control Data 358-2 transceivers should be defined as follows:

```
        CRT  0
```

Example:

A system with two multiplexers, an LCC, and a number of different types of terminals might be defined as follows:

EST Entries:

```
    DC    6000        3,0,0,0,MUX1-T.ITABL
    YC    6000        3,1,0,0,MUX2-T.ITABL
    CS    6000        4,0,0,0,MUX3-T.ITABL
```

Multiplexer Table:

```
    T.ITABL   VFD       6/1LA,6/0,12/0,36/0
              VFD       60/0
    MUX2      MUX76     10B
    MUX1      MUX71     12B
              CRT
              CRT
              CRT       0
              CRT       0
              CRT       0
              CRT       0
              CRT       0,1,2,3
              CRT       4,5,6,7
              EMPTY
              TTY
    MUX3      MUXLCC    3
              HSBT
              TTY
              TTY
    MUX4      MUX73     0,3,G
```

GRAPHICS MULTIPLEXERS

A 6673 or 6674 multiplexer may be designated as a GRAPHICS multiplexer by inclusion of the G parameter. In this case the GCON macro should be specified immediately following. This macro has the following form:

```
GCON      p0, p1, p2,... p23
```

The pi are GRAPHICS console numbers defining the 274 consoles attached via this multiplexer. In the 2-digit console number, the first digit designates the port through which this console is accessible, 0 or 1 for 6673, 0, 1, 2 or 3 for 6674; and the second digit designates the console number on a particular remote system 0-6.

LCC PROGRAMS

The INTERCOM 4.1 LCC initializer uses the LCC multiplexer subtables to determine which variants of the LCC programs to load before the LCC driver is brought up. INTERCOM 4.1 assumes the proper variants are available on the SCOPE system library and are disk resident. The following table indicates the names of the LCC programs that the LCC initializer will search for.

LCC memory size	8k	4k
HSBT ports only defined	0ZD	0ZE
Teletype ports only defined	0ZF	0ZG
HSBT and Teletype ports defined	0ZJ	0ZK*

*Configuration not supported by LCC programs

In addition the LCC autoload program 0ZA must also be available on the system library and be disk resident.

All these programs are available in binary format as part of a separate release of the LCC programs. They may be added to the running system using the following job:

```
Job card.
EDITLIB(SYSTEM)
7/8/9
READY(SYSTEM,OLD)
REPLACE(*,INPUT)
COMPLETE.
ENDRUN.
7/8/9
Binary decks of LCC programs
6/7/8/9
```

DRIVER TYPE SELECTION

If the variant of the low speed driver which supports BCD 200 User Terminals (2400 baud) and Teletypes on the 6671 multiplexer only is not required, card number 40 must be changed to *DEFINE TYPEx, where x is A to N determined from the following table. Find the column containing the combination of Y's which corresponds with your equipment configuration, then obtain the letter x from the bottom row. Any equipment configuration not included on the chart is either an impossibility (e.g., CRTs on a 6676), or it contains more terminal types than can be supported by one variant of the driver (e.g., both BCD and ANSI 200 UT).

IP.N6671≠0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y
IP.N6676≠0														Y	Y	Y	Y	Y	Y	Y	Y
BCD 200 UT(2400 baud)	Y		Y					Y	Y							Y	Y				
TTY		Y						Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
BCD 200 UT(4800 baud)			Y	Y					Y	Y							Y	Y			
ANSI 200 UT(2400 baud)					Y	Y					Y	Y							Y	Y	
ANSI 200 UT(4800 baud)						Y	Y					Y	Y							Y	Y
x	A	B	C	C	D	E	E	F	G	G	H	I	I	J	J	K	L	L	M	N	N

Each hardware channel dedicated to INTERCOM is serviced by only one copy of the driver in a dedicated PPU. In general, a different variant of the driver is required for each hardware channel dedicated to INTERCOM with either a different multiplexer mix or a different terminal mix supported. There are two exceptions: As indicated in the table, the driver variant which drives CRTs at both 2400 and 4800 baud is the same variant which drives CRTs at 4800 baud only; whereas the driver which only drives CRTs at 2400 baud requires a different variant. Likewise, the same variant which drives TTYs on the 6671 and 6676 multiplexers also drives TTYs on the 6676 multiplexer alone; whereas the 6671 alone requires a different variant. Thus, if an installation has two channels dedicated to INTERCOM, one with a 6671 and a 6676 and the other with a single 6676, and only TTYs are supported, only one variant of the driver need be assembled, variant J.

Note that only differences in the type of multiplexers supported, and not differences in the number of multiplexers supported, will require different variants of the driver to be assembled for different channels. However, the symbols IP.N6671 and IP.N6676 must be set to the number of multiplexers of the given type on the channel with the greatest number of that type multiplexer. In the example of the preceding paragraph, the J variant of the driver should be assembled with IP.N6671 set to one and IP.N6676 set to one. Similarly, an installation could assemble a variant of the driver with IP.N6676 set to two, even though only one 6676 is to be supported.

For each additional variant of the low speed driver required, the following cards must be inserted in the installation deck INTCM1 (see Installation Procedures, page I-14-35).

After card 017

```
UPDATE(P=PL12,C=COMPx,Q)  UPDATE DRIVER 1Zx
COMPASS(I=COMPx,S=PPTEXT,S=IPTEXT,L=0)  ASSEMBLE DRIVER 1Zx
```

After card 046

```
*IDENT YYYYYx
*DEFINE TYPEx
*INSERT INTCOM.91
*/
*/   INCLUDE INTERCOM 4.1 INSTALLATION PARAMETER CHANGES HERE
*/   INCLUDING REDEFINITION OF IP.N6671 AND IP.N6676 IF NEEDED
*/
*COMPILE 1M1
7/8/9    END OF RECORD
```

where x is one of the above.

The UPDATE deck name for the low speed driver is 1M1, however, each variant of the driver produced will have a different name of the form 1Zx where x is a character from A through N indicating the terminal mix supported by that driver. (The character x will correspond to that used on the *DEFINE card.) Similarly, the driver's overlays, 8Zx and 9Zx, will have names indicating the terminal mix supported.

When INTERCOM is first initiated, the INTERCOM initialization routine, 1I1, initiates the drivers as dictated by the multiplexers defined in the EST and the port definitions defined in the multiplexer subtables. If all equipments (multiplexers) on a channel are turned off when INTERCOM is initiated, no driver will be initiated to service that channel; however, the multiplexer subtables for all of the equipments will be examined and initialized by 1I1. If some (but not all) of the equipments on a channel are turned off when INTERCOM is initiated, the variant of the driver initiated will be the same as if all equipments were turned on. The multiplexer subtables of all of the equipments, on or off, will be read and initialized.

The user should make certain that only one EST entry points to each multiplexer subtable whether the equipment is on or off.

Installation deck INTCM1 also will compile the relocatable multi-user job subroutines (deckname MUJSUBS). Deck INTCM2 will not add them to the running system for reasons of size and expected infrequency of use. MUJSUBS always must be included on the COMPILE file, however, when EDITOR IS compiled and loaded, so that references to the muj subroutines from EDITOR will be satisfied. If a full UPDATE is done, the subroutines will be included on the COMPILE file. If an UPDATE,Q is done and the EDITOR is to be modified, the UPDATE input must include a *COMPILE MUJSUBS. (EDITOR does not use FTNMUJ or COBOMUJ, the decknames for the FORTRAN Extended and COBOL muj preprocessors.)

After the password files are established and the time has been initialized, INTERCOM should be brought up at control point zero with the console type-in INTERCOM. The INTERCOM system will be ready to service remote terminal users.

COMMAND TABLE STRUCTURE (1CI OVERLAY 3CT)

Prior to INTERCOM installation, release values in the command table of 3CT may be changed to add a new command or multi-user job. These modifications are accomplished with the BATCH, TABLE, and MUJ macros, and should be installed at 1CI.4173.

BATCH MACRO

The BATCH macro is used on all remote batch commands such as READ, GO, etc. Macro format:

```
BATCH  name,ordinal,overlay,level,perm
```

name	Name of remote batch command
ordinal	Command ordinal (within overlay specified by next parameter)
overlay	Overlay of 1CI containing command
level	Access level
perm	Permission bit value

TABLE MACRO

The TABLE macro is used for commands other than batch type which require special processing. Otherwise, placing them in the NUCLEUS directory is sufficient.

TABLE	name, level, FL, TL,,,PERM
name	Name of command
Level	Access level
FL	Field length required
TL	Time limit (in power of two seconds)
PERM	Permission bits for access to command (not used in release version)

MUJ MACRO

This macro defines a table entry for multi-user jobs such as EDITOR. If a new multi-user job is added by the installation, the MUJ macro must define a table entry in the 3CT table. A corresponding entry also must be made in the 1QP MUJTABL to completely define a muj (refer to the description of the 1QP table below). MUJ macro format:

```
        MUJ  name,ordinal,overlay,level,perm
```

name	Name of muj (each defined muj must begin with a different letter, as this letter is used as part of the muj id and must be unique)
ordinal	Command ordinal (within overlay specified by next parameter)
overlay	Section of 1CI containing code to process muj
level	Access level
perm	Permission bit value

In addition to the MUJ macro, the following cards must be inserted in the overlay 3CT for each muj added by the installation:

```
*INSERT  1CI.4092
         LJM    MUJn
*INSERT  1CI.4214
 MUJn    LDN    1QP muj ordinal
         UJK    PROMUJ
```

The first *INSERT places an entry in the jump table in 3CT. The position of the LJM instruction in the jump table constitutes the command ordinal as specified in the MUJ macro. For the EDITOR, this ordinal is 4; the next available entry to be added would be 8, and so on. The second *INSERT adds code for loading the 1QP muj ordinal (position of this muj in 1QP MUJTABL) and for jumping to process the request.

Muj Table Structure (1QP)

Each multi-user job as defined in the command table of 3CT also must be defined in the muj table of 1QP, MUJTABL. The position of an entry in MUJTABL is defined as the 1QP muj ordinal. Entries are made with the macro MUJTBL.

```
MUJTBL  name,fl,swpin,swpout,editor
```

name	Name of the muj
fl	Field length of muj (actual value)
swpin	Delay, in 1CI cycles (depends on IP.TICI, released for 1/2 second), between discovery of need to swap in the muj and actual entry into the scheduling queue. This value increases response time to muj requests (when the muj is swapped out) but allows requests to accumulate; so that when the muj is in, it is more likely to process multiple users. Maximum of 4095.
swpout	Delay, in 1CI cycles, between discovery of need to swap out muj and actual swap out. A high value setting essentially dedicates the muj at a control point.
editor	1 muj EDITOR 0 otherwise

The parameters swpin, swpout, and editor may be null, and default values 1,0, and 0, respectively, will be assumed.

Table Changes and Release Settings

Changes to the tables in routines 3CT and 1QP should be included in the UPDATE record at card 044 of the installation deck INTCM1. The following list shows the release values and UPDATE identifiers.

```
                                                                       1CI
TBEG     EQU    *                                                      1CI      4114
         MUJ    EDITOR,4,8SP,1                                         1CI      4115
         TABLE  CATALOG,0,20B                                          1CI      4116
         TABLE  ATTACH,1,20B                                           1CI      4117
         TABLE  PURGE,0,20B                                            1CI      4118
         TABLE  EXTEND,1,20B                                           1CI      4119
         TABLE  RENAME,1,20B                                           1CI      4120
         TABLE  ALTER,1,20B                                            1CI      4121
         TABLE  SETP,1,20B                                             1CI      4122
         TABLE  MODE,1,10B                                             1CI      4123
         TABLE  PAGE,1,400B,7                                          1CI      4124
         BATCH  TAPE,1,8SP                                             1CI      4125
         BATCH  LOCK,2,8SP                                             1CI      4126
         BATCH  SAVEFL,3,8SP                                           1CI      4127
         BATCH  REDUCE,6,8SP                                           1CI      4128
         BATCH  DMP,5,8SP                                              1CI      4129
         TABLE  MAP,1,10B                                              1CI      4130
         BATCH  IMPORT,7,8SP                                           1CI      4131
*                                                                      1CI      4132
PN       EQU    3                    END OF PAUSE TYPE COMMANDS        1CI      4133
CN       EQU    14                                                     1CI      4134
```

```
*                                      BEGINNING OF COMMAND BATCH TYPE ONLY        1CI       4135
*                                                                                  1CI       4136
         BATCH  S,CN+13,2CX                                                        1CI       4137
         BATCH  G,PN,2CX                                                           1CI       4138
         BATCH  R,CN+1,2CX                                                         1CI       4139
         BATCH  E,PN+10,2CX                                                        1CI       4140
         BATCH  C,CN+4,2CX                                                         1CI       4141
         BATCH  P,PN+6,2CX                                                         1CI       4142
         BATCH  D,1,2CX                                                            1CI       4143
         BATCH  DROP,1,2CX                                                         1CI       4144
         BATCH  SWITCH,2,2CX                                                       1CI       4145
         BATCH  GO,PN,2CX                                                          1CI       4146
         BATCH  H,PN+1,2CX                                                         1CI       4147
         BATCH  COMMENT,PN+2,2CX                                                   1CI       4148
         BATCH  EFL,PN+3,2CX                                                       1CI       4149
         BATCH  ETL,PN+4,2CX                                                       1CI       4150
         BATCH  DIVERT,PN+5,2CX                                                    1CI       4151
         BATCH  PRIOR,PN+6,2CX                                                     1CI       4152
         BATCH  EVICT,PN+7,2CX                                                     1CI       4153
         BATCH  KILL,PN+8,2CX                                                      1CI       4154
         BATCH  M,PN+9,2CX                                                         1CI       4155
         BATCH  MESSAGE,PN+9,2CX                                                   1CI       4156
         BATCH  SCREEN,CN,2CX                                                      1CI       4157
         BATCH  READ,CN+1,2CX                                                      1CI       4158
         BATCH  END,PN+10,2CX                                                      1CI       4159
         BATCH  REW,CN+3,2CX                                                       1CI       4160
         BATCH  BSP,CN+3,2CX                                                       1CI       4161
         BATCH  REP,CN+5,2CX                                                       1CI       4162
         BATCH  RTN,CN+6,2CX                                                       1CI       4163
         BATCH  WAIT,CN+7,2CX                                                      1CI       4164
         BATCH  DSP,CN+8,2CX                                                       1CI       4165
         BATCH  OFF,CN+9,2CX                                                       1CI       4166
         BATCH  ON,CN+10,2CX                                                       1CI       4167
         BATCH  REVERT,CN+11,2CX                                                   1CI       4168
         BATCH  OFFLINE,CN+12,2CX                                                  1CI       4169
         BATCH  SUP,CN+14,2CX                                                      1CI       4170
         BATCH  SUSPEND,CN+13,2CX                                                  1CI       4171
         BATCH  CONTIN,CN+4,2CX                                                    1CI       4172
         TABLE  QU,1,500B,7                                                        1CI       4173
 TEND    EQU    *                                                                  1CI       4174

*CALL INTCOM
 MUJTABL BSS    0                      DEFINE BEGINNING OF TABLE                   1QP        597
         MUJTBL  EDITOR,40000,0,2,1                                                1QP        598
         END
```

6000 OPERATOR CONSOLE COMMANDS

The following commands and one special display (Q) are available to the 6000 operator to aid in his control of the INTERCOM environment. Full details may be seen in the SCOPE Operator's Guide.

DIVERT	Diverts file or files to and/or from remote and/or central.
ILOCK	Prevents users from logging into INTERCOM.
INTERCOM.	Activates INTERCOM (other variants to activate one part of INTERCOM).
IUNLOCK	Cancels ILOCK condition.
M,ID,MESSAGE	Sends message to INTERCOM user id.

The Q display, an INTERCOM display, exists in DSD. This display indicates the operating environment of INTERCOM and lists all logged in users along with their status and current activity. For a complete description of the operator commands and the Q display, refer to the SCOPE Operator's Guide.

MUJ SYSTEM ERRORS

INTERCOM multi-user jobs (e.g. EDITOR), upon encountering hardware and/or software errors, produce diagnostic dumps. These dumps contain a header MUJ SYSTEM ERROR xx. This message is sent to the system dayfile and to each user currently using the muj. Values of xx less than 50 indicate error conditions encountered by the system muj subroutines; values 50 or greater denote errors detected by the multi-user job itself.

Number	Issued By	Error
1	USER	User area lost internally
2	SERVICE	User area lost internally
3	SERVICE	Bit KWCOM should not be set for this value of MMACT (FATAL)
4	SWAPOK	Error (from CIO) on last user area swap
5	SWAPOK	Illegal CIO function code on last user area swap (FATAL)
6	SWAPOK	User area lost on swap-out (FATAL)
7	not used	
8	not used	
9	USER	Muj returning user area not assigned to it. (User error) (FATAL)
10	USER	Invalid ACTN code sent by muj (User error) (FATAL)
11	USER	Invalid information from 1QP
12	USER	Internal logic error
13	USER	A non-ready user was marked as ready
14	LUNSRCH	Logical unit number was specified in call to USERFO, but corresponding file was not declared on muj PROGRAM card (User error) (FATAL)
15	not used	
16	USER	Muj is returning user not assigned to it. (User error) (FATAL)
17	USER	User's files cannot be returned when user leaves muj
50	SYSERR	One of various EDITOR errors. Refer to dump to determine which routine called SYSERR
51	SYSERR	Same as 50, except EDITOR debugging code is on
52	WRTPRN	Input/output error occured on file EDITFIL

EDITOR INSTALLATION PARAMETERS

EDITOR uses two common decks IPFTN (FORTRAN) and IPCOM (COMPASS) to hold installation parameters. Generally, a change to one common deck requires a corresponding change to the other. With the exception of arrays which must be dimensioned for FORTRAN in common deck IPFTN, the values of installation parameters are not defined in IPFTN. IPFTN merely allocates storage for these definitions. The definitions are DATA statements in the BLOCK DATA subprogram IPFILL.

IPCOM contains EQU's which define the installation parameters. Since many parameters are of such a nature that a change in one implies a change of another, a dependency chart is included below to aid the installation.

Summary of the steps to be taken to change an EDITOR installation parameter:

1. Change the DATA Statement in IPFILL or the EQU in IPCOM, or both, as indicated by the parameter description.

2. Consult the dependency chart for any dependent installation parameters that require change, and change them as in step 1.

3. Consult the dependency chart for dimensions of arrays in IPFTN. If they are affected, change them as indicated in the table, Array Dimensions in IPFTN.

Any changes which cause the size of the EDITOR to increase may require an increase in the field length defined for EDITOR in the MUJTABL for 1QP. The following list shows the release values and UPDATE identifiers for IPFILL, IPCOM, and IPFTN.

EDITOR Installation Parameters

In this table, -* in the Range column indicates where a parameter has essentially no absolute upper limit. The installation determines the practical upper limit based on considerations such as EDITOR size and expected number of users.

Parameter Name	Defined In IPFILL	IPCOM	Description	Range	Release Value
NLINE	X		Default first line number for CREATE, EDIT, RESEQ	6L000001 to 6L999999	6L000100
NINCR	X		Default line number increment for ADD, CREATE, EDIT, RESEQ	1-999998	10
NUAS	X	X	Number of user area buffers	1-* Large number decreases response time if there are many users	3
NFILES	X	X	Maximum number of user files which may be attached at any one time	1-* Should equal NBBS because EDITOR assigns a big buffer for each attached file	2
NBBS	X	X	Number of big buffers (used for EDIT, SAVE, RUN)	1-* Increase if many EDITs, SAVEs, RUNs anticipated	2
NPBS	X	X	Number of pool buffers. Each is 64*NPRUS words	2-* Increase when heavy file modifications or long text lines expected, generally NPBS≥NUAS	3
NUSERS	X	X	Maximum number of users simultaneously using EDITOR	1-* Vary with expected usage of EDITOR	30

Parameter Name	Defined In IPFILL	IPCOM	Description	Range	Release Value
NPRUS	X	X	Number of 64-word PRUs in one block in edit file	1-* Large number decreases response time for commands which process large files, but it also increases amount of central memory required for EDITOR by 64 words for each pool buffer and 64 words for each user area buffer	2
NSUA	X		Size of user area; must be modified in IPFILL if NPRUS is changed. NSUA=67+64*NPRUS. Size does not include portion of user area used for tabs, return jump links, and edit file index	131-*	195
JTABS	X		Number of word in user area which holds tab values; must be modified in IPFILL if NPRUS is changed. JTABS=67+64*NPRUS	131-*	195
JNDXHDR	X		Number of index header word in user area; must be modified in IPFILL if NPRUS is changed. JNDXHDR=JTABS+(NTBSMAX+4)/5	132-*	197

Parameter Name	Defined In IPFILL	IPCOM	Description	Range	Release Value
JINDEX	X		Number of first word in edit file index in user area; must be modified in IPFILL if NPRUS is changed. JINDEX=JNDXHDR+1	133-*	198
JRJLNKS	X		Number of first word in return jump link area in user area; must be modified if NPRUS is changed. JRJLNKS=JINDEX+NSINDEX	153-*	218
NSINDEX	X	X	Number of index entries for each user's edit file	1-* Increase for editing very large files	20
NTBSMAX	X	X	Maximum number of tab settings permitted by FORMAT command	1-509 Must be ≥ NTBSFTN, NTBSCOM, NTBSCOB, NTBSALG, NTBSDEF	10
XNPCENT	X		Percent to which each block of user's edit file is filled by EDIT (Padding factor)	.01-1.00 Decrease if heavy file modification is expected	.90
NTABFTN	X		FORTRAN tab character	1LA-1L;	1L;
NTABCOM	X		COMPASS tab character	1LA-1L;	1L;
NTABCOB	X		COBOL tab character	1LA-1L;	1L;
NTABALG	X		ALGOL tab character	1LA-1L;	1L$
NTABDEF	X		Default tab character	1LA-1L;	1L;

Parameter Name	Defined In IPFILL	IPCOM	Description	Range	Release Value
NTBSFTN	X		Number of FORTRAN tabs defined	0-509	1
NTBSCOM	X		Number of COMPASS tabs defined	0-509	3
NTBSCOB	X		Number of COBOL tabs defined	0-509	5
NTBSALG	X		Number of ALGOL tabs defined	0-509	5
NTBSDEF	X		Number of Default tabs defined	0-509	1
NCHFTN	X		Maximum no. of characters in FORTRAN line	1-510	72
NCHCOM	X		Maximum no. of characters in COMPASS line	1-510	72
NCHCOB	X		Maximum no. of characters in COBOL line	1-510	72
NCHALG	X		Maximum no. of characters in ALGOL line	1-510	72
NCHDEF	X		Maximum no. of characters in default format	1-510	72
NCHBAS	X		Maximum no. of characters in BASIC line	1-510	72

Parameter Name	Defined In IPFIL	IPCOM	Description	Range	Release Value
FTNTABS	X		Consecutive stream of bits, each 12 define a tab position for FORTRAN format. Must be ascending order	1-511 (each tab)	00070000000000000000B
COMTABS	X		Same as above, for COMPASS	1-511	00130022004400000000B
COBTABS	X		Same as above, for COBOL	1-511	00100014002000240030B
ALGTABS	X		Same as above, for ALGOL	1-511	00070012001500200023B
DEFTABS	X		Same as above, for Default format	1-511	00070000000000000000B
NSBB		X	Size of big buffers used for EDIT, SAVE, RUN (does not include FET)	64-* Increase for very large files	257

EDITOR Array Dimensions in IPFTN

Array Name	Usage	Array Dimension
FTNTABS	FORTRAN tabs	(NTBSFTN+4)/5
COMTABS	COMPASS tabs	(NTBSCOM+4)/5
COBTABS	COBOL tabs	(NTBSCOB+4)/5
ALGTABS	ALGOL tabs	(NTBSALG+4)/5
DEFTABS	Default tabs	(NTBSDEF+4)/5
MMUJTBL	Storage needed by muj subroutine tables	3*NUSERS + NFILES + NUAS + 6*(NPBS+1)
MUAS	User area buffers May never exceed 4095 decimal	NUAS* (size of full user area) where: (size of full user area) = (NTBSMAX+4)/5 + NSINDEX +1 + NSUA + NSRJLNK Note: NSRJLNK should not have to be changed by an installlation
MBBS	Big buffers	NBBS*NSBB + NBBS*6
MPBS	Pool buffers	NPBS*64*NPRUS
MBBMA	Big buffer management area	NBBS
MPBMA	Pool buffer management area	NPBS

EDITOR Dependency Chart

If changed	Check parameters in IPFILL and/or IPCOM	and arrays in IPFTN
NLINE		
NINCR		
NUAS		MMUJTBL,MUAS
NFILES	NBBS	MMUJTBL
NBBS	NFILES	MBBS,MBBMA
NPBS		MPBS,MPBMA,MMUJTBL
NUSERS*		MMUJTBL
NSINDEX	JRJLNKS	MUAS
NTBSMAX	JNDXHDR,JINDEX,JRJLNKS	MUAS
XNPCENT		
NTABFTN		
NTABCOM		
NTABCOB		
NTABALG		
NTABDEF		
NTBSFTN	NTBSMAX, FTNTABS	FTNTABS
NTBSCOM	NTBSMAX, COMTAABS	COMTABS
NTBSCOB	NTBSMAX, COBTABS	COBTABS
NTBSALG	NTBSMAX, ALGTABS	ALGTABS
NTBSDEF	NTBSMAX, DEFTABS	DEFTABS
NCHFTN		
NCHCOM		
NCHCOB		
NCHALG		
NCHDEF		
NCHBAS		
FTNTABS	NTBSFTN	FTNABS
COMTABS	NTBSCOM	COMTABS
COBTABS	NTBSCOB	COBTABS
ALGTABS	NTBSALG	ALGTABS
DEFTABS	NTBSDEF	DEFTABS
NSBB		MBBS
NDEBUG		
NPRUS	JTABS,JNDXHDR,JINDEX, JRJINKS,NSUA	MPBS,MUAS

*When NUSERS is increased, the user should consider changing also the size of the TERMIN and TERMOUT tables in the muj subroutines MUJSUBS. See INTERCOM MULTI-USER JOB CAPABILITY Programming System Bulletin (60358300) under the heading "Changing Size of TERMIN and TERMOUT."

EDITOR DEBUG CODE

If EDITOR encounters hardware and/or software problems, diagnostic printout is produced. If the problem is considered fatal, all EDITOR users are detached. The content of the diagnostic printout depends upon the error encountered and the setting of NDEBUG. In any event, the diagnostic printout should accompany any PSR relating to a MUJ SYSTEM ERROR. See also MDEBUG below.

MULTI-USER JOB INSTALLATION PARAMETERS

The multi-user job (muj) subroutines use two common decks, MUJCOM and CMUJCOM. Both contain storage allocation for an array, ECSBUF. The MUJCOM deck in FORTRAN code contains a DIMENSION statement; the CMUJCOM deck in COMPASS code contains a BSS statement. This array is used by the muj peripheral processor routines, FAD, to read information from Extended Core Storage (ECS). Array length must be (n*64+1) central memory words. The value of n may be selected by the installations, depending on the expected use of ECS for storage of user swap files (if ECS will be used, n should be at least 2) and on the number of local files allowed for an INTERCOM user. As a guide, n may be increased by one for each 20 local files allowed per user. The upper limit for n is dependent on the amount of storage to be used for ECS buffer in the muj, and the size of the swap buffer in FAD.

The peripheral processor routine FAD contains two parameters relevant to allocation of space for ECSBUF. ECSBFLN (near FAD.659) is a COMPASS EQU instruction. It must be equated to the number of central memory words in the ECSBUF array. SWAPBF (near FAD.650) is a table FAD uses to read the ECSBUF array into PP memory. The value of ECSBFLN, and thus the size of the ECSBUF array in MUJCOM and CMUJCOM, must not be greater than 1+ (length of SWAPBF)/5.

MDEBUG

Symbol MDEBUG in common deck CMUJCOM controls muj debugging code (0=off, 1=on). It should be set to 1 if EDITOR installation parameter NDEBUG is set to 1.

INSTALLATION PROCEDURES

Two installation decks are included on the released program library as files 2 and 3. Deck INTCM1 assembles the released program library adding the created binary to the PL tape as supplemental files. The release tape does not contain assembled binary. Deck INTCM2 uses EDITLIB to enter the binary created by deck INTCM1 into the running system. Deck INTCM1 will require modification if the single default low speed driver type is not the correct variant or multiple low speed driver types are desired. If the hardware configuration does not contain 200 User Terminals or teletype devices (e.g., LCC only system), assembly of the low-speed multiplexer driver can be avoided by changing card number 17 of deck INTCM1 to COPYBR(COMPILE,NIL).

```
Job card.
REQUEST(PL12,E)
REWIND(PL12)
SKIPF(PL12,1,17)           Skip PL
COPYBF(PL12,PUNCH,2)       Punch installation decks
UNLOAD(PL12)
6/7/8/9
```

With deck INTCM1, one variant of the PP multiplexer low speed driver will be produced which will drive BCD 200 User Terminals (2400 baud) and Teletypes on the 6671 multiplexer. The exact number and type of multiplexers supported by the driver is determined by the settings of the IP.N6671 and IP.N6676 symbols in the INTERCOM common deck INTCOM. Instructions follow the deck listing regarding how to change the driver definition or increase the number of drivers defined.

Deck INTCM2 suggests CM residency for selected PP routines. Sites having ECS may wish to move some of these PP routines to ECS by employing the method discussed in the Deadstart portion of the SCOPE section.

NOTE: The T7000 parameter on the INTCOM1 job card must be changed to T20000 if the deck is to be run on a 6200.

```
INTCM1,CM65000,T7000,MT2.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF INTERCOM
COMMENT. THE NEW PL12 WILL CONSIST OF FIVE FILES
COMMENT. THE FIRST FILE WILL BE THE NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY OF THE PP ROUTINES
COMMENT. THE THIRD FILE WILL BE THE RELOCATABLE BINARY OF THE OVERLAYS
COMMENT. THE FOURTH FILE WILL BE THE BINARIES OF THE INTERACTIVE GRAPHICS
COMMENT. SYSTEM
COMMENT. THE FIFTH FILE WILL BE THE ABSOLUTE BINARIES OF THE OVERLAYS
LABEL(PL12IN,R,L=INTERCOM4P1*3P4,D=HI)INTERCOM OLDPL
REQUEST,PL12,N,HI.    SCRATCH FOR NEW PL12
LABEL(PL12,W,L=INTERCOM4P1*3P4,D=HI)
REWIND(PL12IN,PL12)
UPDATE(F,P=PL12IN,N=PL12,C=0)
UPDATE(F,P=PL12)
UNLOAD(PL12IN)
COMPASS(I,S=PPTEXT,S=IPTEXT,L=0)                 DRIVER
COMPASS(I=COMPILE,S=SCPTEXT,S=IPTEXT,S=SCHTEXT,S=STATEXT,L=0)
FTN(I=COMPILE,S=CPCTEXT,S=PFMTEXT,S=IPTEXT,S=IOTEXT,B=LGO1,L=0,OPT=1)
COPYBR(COMPILE,NIL)
FTN(I=COMPILE,S=CPCTEXT,B=LGO2,L=0,OPT=1)  IGS ROUTINES
FTN(I=COMPILE,S=CPCTEXT,B=LGO2,L=0,OPT=1)  IGS ROUTINES
REWIND,LGO,LGO1,LGO2.
SKIPF(PL12,1,17)
COPYBF(LGO,PL12)
COPYBF(LGO1,PL12)
COPYBF(LGO2,PL12)
REWIND,LGO1.
LOAD,LGO1.
NOGO.
REWIND(INTRCOM)
COPYBF(INTRCOM,PL12)
```

```
UNLOAD(PL12)
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
*ID YNKMUJST
*YANKDECK,MUJSTAT                        (EDITOR DOES NOT USE MUJSTAT)
*IDENT,YYYF
*DEFINE,TYPEF       SELECTS DRIVER TYPE
*I INTCOM.91
 IF.N6671 CEQU   1
 IF.N6676 CEQU   0
*COMPILE,1M1
7/8/9  END OF RECORD
*/  PLACE MULTIPLE DRIVER UPDATE MODIFICATION RECORDS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

Deck INTCM2 suggests CM residency for selected PP routines. Sites having ECS may wish to move some of these PP routines to ECS by employing the method discussed in the Deadstart portion of the SCOPE section.

```
INTCM2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS THE INTERCOM V4.1 BINARIES FROM THE NEW PL12
COMMENT. CREATED BY THE DECK INTCM1.
LABEL(PL12,R,L=INTERCOM4P1*3P4,D=HI)  MOUNT INTERCOM V4.1 PL
REWIND,PL12.
SKIPF(PL12,1,17)            SKIP OLDPL
COPYBF(PL12,INTPP)          INTERCOM PP ROUTINES
SKIPF(PL12,1,17)            SKIP OVERLAYS IN RELOCATABLE FORMAT
COPYBF(PL12,IGS)            BINARIES OF IGS
COPYBF(PL12,INTCP)          INTERCOM CP ROUTINES
REWIND,INTPP,IGS,INTCP,PL12.
UNLOAD(PL12)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
REPLACE(*,INTPP)
MOVE(1BR,CM)
MOVE(1QP,CM)
MOVE(3TT,CM)
MOVE(3T1,CM)
MOVE(3T2,CM)
MOVE(1CI,CM)
MOVE(3CI,CM)
MOVE(3CT,CM)
MOVE(3CF,CM)
LIBRARY(NUCLEUS,OLD)
REPLACE(LOGIN,INTCP,AL=1,FL=20400,FLO=0)
REPLACE(CONNECT,INTCP,AL=3,FL=200,FLO=0)
REPLACE(SITUATE,INTCP,AL=3,FL=16100,FLO=0)
REPLACE(DISCONT,INTCP,AL=3,FL=200,FLO=0)
REPLACE(BATCH,INTCP,AL=1,FL=16300,FLO=0)
REPLACE(G,INTCP,AL=3,FL=14400,FLO=0)
REPLACE(PAGE,INTCP,AL=3,FL=32400,FLO=0)
REPLACE(SEND,INTCP,AL=3,FL=20100,FLO=0)
REPLACE(LOGOUT,INTCP,AL=1,FL=12300,FLO=0)
REPLACE(CONVERT,INTCP,AL=3,FL=22300,FLO=0)
REPLACE(ERESEQ,INTCP,AL=3,FL=40000,FLO=0)
```

```
REPLACE(TEACH,INTCP,AL=1,FL=31400,FLO=0)
REPLACE(FILES,INTCP,AL=1,FL=10300,FLO=0)
REPLACE(STORE,INTCP,AL=1,FL=14200,FLO=0)
REPLACE(DISCARD,INTCP,AL=1,FL=12000,FLO=0)
REPLACE(FETCH,INTCP,AL=3,FL=14000,FLO=0)
REPLACE(XEQ,INTCP,AL=3,FL=15300,FLO=0)
REPLACE(PASSWRD,INTCP,AL=7777,FL=46400,FLO=0)
REPLACE(ASSETS,INTCP,AL=1,FL=10200,FLO=0)
REPLACE(READ,INTCP,AL=3,FL=7400,FLO=0)
REPLACE(ERRORS,INTCP,AL=3,FL=26000,FLO=0)
REPLACE(TESTLF,INTCP,AL=3,FL=2000,FLO=0)
REPLACE(EDITOR,INTCP,AL=3,FL=37000,FLO=0)
FINISH.
LIBRARY(SYSOVL,OLD)
REWIND(INTCP)
REPLACE(Q6000+Q7000,INTCP)
REPLACE(ERFTN+ERRUN,INTCP)
FINISH.
LIBRARY(IGS274,OLD)
REPLACE(*,IGS,AL=0)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

PASSWORD FILE CREATION

Access to the INTERCOM system is controlled by passwords. The user must specify a valid password to log in to the INTERCOM system. Two types of passwords exist:

Restricted passwords: When he logs in, the user must specify a valid user-name associated with the given password. The installation defines valid password/user-name combinations. A user id (two alphanumeric characters) is assigned by the PASSWRD utility, and it is permanently associated with the password/user-name. This user id is assigned from a pool of available user id's; it is marked as available again only when the password/user-name is deleted.

Unrestricted passwords: The user may specify any user-name when he logs in; the user-name is not validated. However, when a user first logs in under a given user-name, a user id is associated by the LOGIN utility with that password/user-name combination. Thereafter, this user id is associated with the password/user-name combination, until the password/user-name is deleted from the system.

Through the INTERCOM routine PASSWRD, the installation defines valid restricted password/user-name combinations and valid unrestricted passwords and accounting values to be associated with the password/user-names or passwords. PASSWRD must be called from a data deck submitted to the central site as a batch job. The routine will create two permanent files (or edit existing files). One file, with the permanent file name INTERCOMPASSWORDS, contains all unrestricted passwords, all restricted password/user-names, and all accounting information. The other file, with the permanent file name INTERCOMUNRESTRICTED, contains a bit map defining assigned user id's; it also contains all unrestricted password/user-name combinations currently specified at LOGIN time.

A total of 1296 user ID's (all possible 2-character combinations of A-Z, 0-9) are available to the INTERCOM system. User ID's are assigned permanently to password/user-name combinations. Installations with many users should:

Instruct users of unrestricted passwords always to use the same character string for user-name when logging in.

Make use, on a regular basis, of the editing facilities in PASSWRD to delete all unrestricted user-names, etc, freeing user ID's.

While a user is in the process of logging in, he is assigned a temporary ID. Temporary ID's begin with one of the following special characters: + - * / ( ) $ =

The following deck structure can be used to run the PASSWRD routine, creating a password permanent file:

```
Job card.
PASSWRD.
7/8/9
NEW
ADD
.
.
6/7/8/9
```

The following deck structure can be used to modify existing password permanent files:

```
Job card.
PASSWRD.
7/8/9
OLD
ADD or
DEL
.
.
.
6/7/8/9
```

This mode of PASSWRD operation will update the existing permanent files by adding new or deleting old entries. If both files do not exist a PF ERROR=12B will abort the run.

To protect against unauthorized modification of the password files, the PASSWRD utility requests permission from the console operator before any modifications are made.

Between the NEW (or OLD) card and the 6/7/8/9 card appear the parameter cards which specify the new entries or the editing requirements. After a NEW card, only ADD parameter cards may appear; after an OLD card, either ADD or DEL parameter cards may appear. The ADD card creates a new entry, or replaces an old entry which has the same password/user-name. The DEL card deletes one or more entries. The NEW card may be used to delete existing files entirely and to construct new ones.

The format for an ADD parameter card is:

```
ADD U=username,P=password,F=flength,T=time,A=acclevl,N=nfiles
```

username — User name (1-10 alphanumeric characters) must be specified for restricted passwords; it must be blank or omitted for unrestricted passwords.

password — Password (1-10 alphanumeric characters) must be specified. It must be the only unrestricted password of this name defined by the installation. If it is restricted, it must be the only password/user-name of this particular combination defined by the installation. (If the password or password/user-name have been previously defined, the ADD card will function as a replace.)

flength — Maximum field length available to the user (1-6 octal digits). If blank or omitted, 60000 octal CM words will be assumed. This value may not exceed IP.MFL.

time — Time limit for user's session (1-4 octal digits, also defines the maximum ETL for individual jobs). If blank or omitted, 500 octal seconds will be assumed.

acclevl — Access level/permission bits for the user (1-4 octal digits). This value defines which programs the user can access. If blank or omitted, an access level of 5 is assumed.

nfiles — Number of files this user is permitted to attach as local files at any one time (1-2 octal digits). If blank or omitted, 24 (octal) files will be allowed. This value may not exceed 76B.

All parameters start after column four on the ADD and DEL cards. They may be specified in any order and should be separated by delimiters (special characters).

The DEL card is used to delete one or more entries from one or both of the permanent files. It has two formats:

```
DEL U=username,P=password

DEL I=id
```

username — May take three forms: 1-10 alphanumeric characters, blank, or the character string *NAMES. If the first form is used, the password/user-name combination (restricted or unrestricted) will be deleted; and the user ID will become available. If the second form is used, all entries in the two files with the given password will be deleted. All user ID's associated with these entries will become available; the password will no longer be defined. The third form may be used only if the specified password is unrestricted. All entries in the unrestricted password file with the given password will be deleted, and the associated ID's will be made available. The password will still be defined.

password — Password to be processed. Whether an unrestricted password is deleted or not depends on the username parameter. If password is *NAMES, all user-names for all unrestricted passwords will be deleted from the permanent files; and the user ID's for these user-names will become available. The unrestricted passwords will still be defined.

ID — User ID; may be used as a shorthand notation to specify the password/user-name associated with this user ID. The given password/user-name entry (restricted or otherwise) is deleted and the user ID becomes available. If the password is unrestricted, it will still be defined.

VERIFICATION PROCEDURE

INTERCOM is brought to control point zero when INTERCOM is entered at the console after the operator has entered the time.

The verification procedure cannot proceed unless a permanent file has been established containing the user passwords.

The following sample from a Teletype terminal session will indicate if INTERCOM 4.1 is installed correctly. The underlined characters are typed by the user.

```
CONTROL DATA INTERCOM 4.1
DATE  07/19/72
TIME  14.11.41.

PLEASE LOGIN
LOGIN
ENTER USER NAME-USERA
[redacted] ENTERPASSWORD-
```

```
 07/19/72    LOGGED IN AT  14.12.30.
             WITH USER-ID BT
             EQUIP/PORT 54/03
COMMAND- FILES
NONE
COMMAND- EDITOR
..FORMAT,FORTRAN
..FORMAT,SHOW
CH= 72 TAB CHAR=; TAB COL=    7
..CREATE
   100=;PROGRAM A(OUTPUT)
   110=;PRINT 1
   120=1;FORMAT(*  TEST*)
   130=;END
   140==
..SAVE,B
..RUN,FTN
       .084 CP SECONDS COMPILATION TIME
 TEST
      END  A
..BYE,BYE
COMMAND- FILES
--LOCAL FILES--
```

```
   B          $INPUT       $OUTPUT      LGO
COMMAND- RETURN,B
COMMAND- RETURN,LGO
COMMAND- FILES
--LOCAL FILES--
  $INPUT      $OUTPUT
COMMAND- EDITOR
..FORMAT,BASIC
..FORMAT,SHOW
CH= 72 TAB CHAR=; TAB COL=    0
..10 PRINT # TYPE A NUMBER OR 0 (ZERO) TO END #
20 INPUT X
25 IF X=0 THEN 80
30 F=1
40 FOR I = 1 TO X
50 F = F * I
55 NEXT I
60 PRINT # FACTORAL # X; # IS #F;
70 GO TO 10
80 END
SAVE,BASIN
..BYE
COMMAND- REWIND(BASIN)
COMMAND- CONNECT(BASOUT)
COMMAND- BASIC(I=BASIN,K=BASOUT)
```

```
 TYPE A NUMBER OR 0 (ZERO) TO END
\      2
 FACTORAL  2      IS  2      TYPE A NUMBER OR 0 (ZERO) TO END
\      3
 FACTORAL  3      IS  6      TYPE A NUMBER OR 0 (ZERO) TO END
\      4
 FACTORAL  4      IS  24     TYPE A NUMBER OR 0 (ZERO) TO END
\      0
COMMAND- DISCONT,BASOUT
COMMAND- FILES
--LOCAL FILES--
  $INPUT      $OUTPUT      BASIN       BASOUT
COMMAND- LOGOUT
CP TIME       2.819
PP TIME      21.796
CONNECT TIME     0 HRS. 11 MIN.
 07/19/72  LOGGED OUT AT 14.23.42.<
```

RELEASE DESCRIPTION

HARDWARE CONFIGURATION

PERT/TIME requires a minimum hardware configuration of one computer with at least 65K central memory, three tape units, and the other minimum equipment required by SCOPE.

RELEASE MATERIALS

PERT/TIME is released on program library PL13.

LIMITATIONS

Since the file names TAPE1 through TAPE6 are used internally, no user file name may be TAPE1-6.

CORRECTIONS

All eligible PSR code as published through PSR Summary 312 has been added to the program library.

GENERAL PROCEDURES

PERT/TIME 1.2 runs under SCOPE 3.4 and is on the PL13 program library tape.

INSTALLATION PARAMETERS

None for PERT/TIME.

INSTALLATION PROCEDURES

PL13 contains 4 files. File 1 is the program library; file 2 contains PERT in overlay format. Files 3 and 4 contain the installation and verification program decks.

To obtain these decks, perform the job:

```
Job card.
REQUEST(PL13,E)
REWIND(PL13)
SKIPF(PL13,2,17)
COPYBF(PL13,PUNCH,2)          PUNCH INSTALLATION AND VERIFICATION DECKS
UNLOAD(PL13)
6/7/8/9
```

```
PERT,CM110000,T7000,MT2.                                          001
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF PERT.         002
COMMENT. THE NEW PL13 WILL CONSIST OF TWO FILES                   003
COMMENT. THE FIRST FILE WILL BE THE PERT NEWPL                    004
COMMENT. THE SECOND FILE WILL BE THE ABSOLUTE BINARY OF PERT      005
COMMENT.                                                          006
LABEL(PL13IN,R,L=PERT*3P4,D=HI)  MOUNT PERT PL                    007
REQUEST,PL13,N,HI.     SCRATCH FOR NEW PL13                       008
LABEL(PL13,W,L=PERT*3P4,D=HI)                                     009
REWIND(PL13IN,PL13)                                               010
UPDATE(F,P=PL13IN,N=PL13)                                         011
UNLOAD(PL13IN)                                                    012
RUN(S,,,COMPILE,NONE,PERTBIN,100000)                              013
LOAD(PERTBIN)           GENERATION OF OVERLAY PERT66              014
NOGO.                                                             015
REWIND,PERT66.                                                    016
REWIND(PERTBIN)                                                   017
SKIPF(PL13,1,17)                                                  018
COPYBF(PERT66,PL13)                                               019
UNLOAD(PL13)                                                      020
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE                                          022
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

The installation deck PERT provided as release tape file 3 compiles the PERT/TIME product, forms the necessary overlays, and writes these overlays on the program library tape. PERT must be executed from this overlay tape; PERT cannot be executed from the running system. It is not possible to build a PERT/TIME binary tape or to execute programs on a computer with only 32K memory.

To execute from the overlay tape, the following deck is required:

```
PERT,T1000,CM120000,TP1.

REQUEST TAP,E.

REWIND(TAP)

SKIPF(TAP,1,17)

COPYBF(TAP,PERT66)

PERT66.

7/8/9
          PERT networks
7/8/9

6/7/8/9
```

If a master tape is made, insert after card 3:

```
REQUEST TAPE6,MT.
```

If an old master tape is input and a new master tape saved, insert the following after card 3:

```
REQUEST TAPE4,MT.        OLD MASTER
REQUEST TAPE6,MT.        NEW MASTER
```

If either option is used the TP1 job card parameter must be increased.

Successive PERT/TIME networks may be processed by batching as follows:

```
K
L
M
X
Y
W
A    PERT BATCHED NETWORKS
•
•
•
A
Z
K
L
•
•
•
Z
```

(Input for each network starts with the K control card and ends with the Z control card).

I-15-4

CLASSIFICATION

PERT/TIME
ACTIVITY REPORT

REPORTING ORGN. CONTRACT NO.
333

PERT VERIFICATION PROGRAM
1ST SORT KEY PREDECESSOR EVENT NO.
2ND SORT KEY SUCCESSOR EVENT NO.
3RD SORT KEY LEAST SLACK
4TH SORT KEY EXPECTED DATE (TE)

TERM-
REPORT DATE- 9/17/64
RELEASE DATE- 9/17/64

	EVENT				ACTIV.	DATE		DATE		REMAINING		
	PRED.	SUCC.	ACTIVITY DESCRIPTION	PROB.	TIME	EXPECTED	ALLOWED	COMP/SCHED	SLACK	TIME	ORG.	ACCOUNT NO.
	00	01			0.0		9/25/64	A 9/17/64	1.1	0.0	ORG1	012345678900
I	01	02		.97	7.6	11/10/64	11/18/64		1.1	7.6	ORG4	01
I	01	03		.99	3.6	10/13/64	11/19/64		5.3	3.6	ORG3	001234567890
I	01	04		.99	5.3	10/26/64	11/16/64		3.1	5.3	ORG1	51
	02	05		.78	4.3	12/11/64	12/18/64		1.1	11.9	ORG4	02
	02	09		.99	4.3	12/11/64	4/23/65		18.6	11.9	ORG1	31
	03	05		.99	4.1	11/11/64	12/18/64		5.3	7.7	ORG4	03
	03	06		.99	4.1	11/11/64	12/24/64		6.0	7.7	ORG1	01
	04	07		.99	2.3	11/10/64	12/ 3/64		3.1	7.6	ORG2	53
	04	08		.99	3.8	11/20/64	1/13/65		7.1	9.1	ORG1	52
	05	10		.76	4.0	1/12/65	1/19/65		1.1	15.9	ORG5	023456718560
	06	10		.99	3.0	12/ 3/64	1/19/65		6.3	10.7	ORG1	03
	06	11		.99	3.0	12/ 3/64	1/18/65		6.0	10.7	ORG2	81
	07	11		.99	6.0	12/23/64	1/18/65		3.1	13.6	ORG2	52
	08	12		.99	4.3	12/22/64	2/12/65		7.1	13.4	ORG1	54
	09	13		.99	0.0	12/11/64	4/23/65		18.6	11.9	ORG1	31
	10	15		.75	5.0	2/16/65	2/23/65		1.1	20.9	ORG2	01
	11	16		.98	3.5	1/20/65	2/10/65		3.1	17.1	ORG2	81
	12	14		.99	3.1	1/15/65	3/ 5/65		7.1	16.5	ORG5	51
	13	18		.99	1.1	12/18/64	4/30/65		18.6	13.0	ORG4	71
	14	17		.99	4.6	2/17/65	4/ 7/65		7.1	21.1	ORG2	41
	15	19		.74	2.8	3/ 8/65	3/15/65		1.1	23.7	ORG5	61
	15	22		.99	6.8	4/ 5/65	6/14/65		10.0	27.7	ORG1	71
	16	20		.98	4.6	2/22/65	3/15/65		3.1	21.7	ORG2	82
	17	21		.99	2.5	3/ 5/65	4/26/65		7.1	23.6	ORG1	42
	18	22		.99	6.1	2/ 3/65	6/14/65		18.6	19.1	ORGJ	72
	19	20		.74	0.0	3/ 8/65	3/15/65		1.1	23.7	ORG2	82
	20	22		.99	7.6	4/29/65	6/14/65		6.4	31.3	ORG4	71
	20	25		.70	14.5	6/16/65	6/24/65		1.1	38.2	ORG1	11
	20	27		.99	13.6	6/10/65	8/ 2/65		7.5	37.3	ORG4	21
	21	24		.99	3.0	3/26/65	5/17/65		7.1	26.6	ORG2	41
	22	23		.99	-0.0	4/29/65	6/14/65		6.4	31.3	ORG2	72
	23	26		.99	4.1	5/27/65	7/12/65		6.4	35.4	ORG2	22
	23	27		.99	5.1	6/ 3/65	8/ 2/65		8.4	36.4	ORG1	21
	24	25		.99	5.6	5/ 5/65	6/24/65		7.1	32.2	ORG5	42
	25	27		.69	5.5	7/26/65	8/ 2/65		1.1	43.7	ORG1	11
	26	27		.99	3.0	6/17/65	8/ 2/65		6.4	38.4	ORG5	23

CLASSIFICATION

60307400 A

VERIFICATION PROGRAM

Output from the PERT/TIME Verification program follows. The time required to run this program is less than two minutes of 6400 computer clock time.

```
              SCCFE 3.4
05.45.08.VPERT37
05.45.09.VPERT,CM120000,T7000,TP1.
05.45.09. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
05.45.09.STALLATION OF
05.45.09. PERT/TIME
05.45.09.REQUEST,PL13,E,HI. LATEST PERT/TIME OLDP
05.45.09.L
05.45.52.( MT25 ASSIGNED)
05.45.52.REWIND,PL13.
05.45.53.SKIPF(PL13,1,17)
05.45.57.MT25 VOLUME SERIAL NUMBER IS  004246
05.46.02.COPYBF(PL13,PERT66)
05.46.14.$MT 25  RD RVD TAPE PAR ERR
05.46.14.$MT 25  PRU = 00000221
05.46.17.RETURN(PL13)
05.46.18.PERT66.
05.46.27.      8.464 RT SECONDS LOAC TIME
05.47.36.STOP
05.47.39. END CF JOB
05.47.39.CPA      4.049 SEC.
05.47.39.PP      60.938 SEC.
```

RELEASE DESCRIPTION

SIMSCRIPT Version 2.0 operates under SCOPE 3.4 on the same minimum configuration as SCOPE.

RELEASE MATERIALS

The release materials for SIMSCRIPT Version 2.0 are included on program library tape PL14.

CORRECTIONS

All eligible PSR code as published through PSR Summary No. 312 has been added.

LIMITATIONS

SIMSCRIPT Version 2.0 operates under SCOPE 3.4 in conjunction with COMPASS 3.0 and either FORTRAN Extended 3.0 or FORTRAN 2.3.

Jobs interfacing with FORTRAN 2.3 require a card of the form LIBRARY (RUN2P3, SYSMISC) preceding the LGO directive. Jobs interfacing with FORTRAN Extended 3.0 require the V3.0 FTN object library to be installed and a card of the form LIBRARY (SYSMISC) preceding the LGO directive.

INSTALLATION PARAMETERS

None

INSTALLATION PROCEDURES

PL14 contains 6 files. File 1 contains the SIMSCRIPT program library; file 2 contains the relocatable binary of the execution time routines; file 3 contains the compiler overlays in absolute binary format. Files 4 and 5 contain the installation decks; file 6 contains the verification program.

To obtain the SIMSCRIPT installation and verification decks, perform the following job:

```
Job card.
REQUEST(PL14,E)               MOUNT SIMSCRIPT PL
REWIND(PL14)
SKIPF(PL14,3,17)
COPYBF(PL14,PUNCH,3)          PUNCH INSTALLATION AND VERIFICATION DECKS
UNLOAD(PL14)
6/7/8/9
```

Job SIMSC1 is a maintenance deck which creates a release format tape containing a revised program library and compiled binary. Job SIMSC2 can be used to enter SIMSCRIPT into the running system through EDITLIB either from the released tape or from the tape created by SIMSC1. Job SCOPE3, described in section 1, then should be run to generate a deadstart tape of the running system.

```
SIMSC1,CM65000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF SIMSCRIPT
COMMENT. THE NEW PL14 WILL CONSIST OF THREE FILES
COMMENT. THE FIRST FILE WILL BE THE SIMSCRIPT NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY
COMMENT. OF THE RELOCATABLE CP ROUTINES
COMMENT. THE THIRD FILE WILL BE THE BINARY OF THE OVERLAY SIMS
LABEL(PL14IN,R,L=SIMSCRIPT*3P4,D=HI)  SIMSCRIPT PL
REQUEST,PL14,N,HI.       SCRATCH FOR NEW PL14
LABEL(PL14,W,L=SIMSCRIPT*3P4,D=HI)
REWIND(PL14IN,PL14)
SKIPF(PL14IN,2,17)
COPYBF(PL14IN,SIMS)
REWIND,SIMS.
UPDATE(F,P=PL14IN,N=PL14)
UNLOAD(PL14IN)
SIMS(I=COMPILE,L=0)   SIMS CREATES COMPASS SOURCE ON FILE MAPTP
COMPASS(I=MAPTP,L=0,S=CPCTEXT,S=IPTEXT)
REWIND,LGO.
SKIPF(PL14,1,17)
COPYBF(LGO,PL14)
REWIND,LGO.
COPYBR(LGO,SIMBIN,85)
REWIND,SIMS.
LOAD(SIMBIN)
NOGO.
REWIND,SIMS.
COPYBF(SIMS,PL14)
UNLOAD(PL14)
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
SIMSC2,CM55000,T7000,MT1.
COMMENT. THIS JOB EDITLIBS THE SIMSCRIPT BINARIES FROM THE PL14 TAPE,
COMMENT. EITHER THE RELEASED VERSION OR THE VERSION CREATED BY DECK
COMMENT. SIMSC1, INTO THE RUNNING SYSTEM.
LABEL(PL14,R,L=SIMSCRIPT*3P4,D=HI) MOUNT PL14
REWIND,PL14.
SKIPF(PL14,1,17)             SKIP CLDPL
COPYBR(PL14,NIL,85)          SKIP RELOCATABLE COMPILER ROUTINES
COPYBF(PL14,SIMLB)           SIMSCRIPT LIBRARY ROUTINES
COPYBF(PL14,SIMC)            SIMSCRIPT COMPILER
REWIND,PL14,SIMLB,SIMC.
UNLOAD(PL14)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(*,SIMC,AL=3,FL=47000,FLO=1)
FINISH.
LIBRARY(SYSMISC,OLD)
REPLACE(*,SIMLB)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

The verification deck provided with the release validates SIMSCRIPT. The validation consists of a SIMSCRIPT job with a report. The time required to run the validation job is about one minute. The output includes the actual validation (magic squares) and dayfile as follows:

```
THIS IS A 3 BY 3 MAGIC SQUARE. ALL OF THE ROWS,COLUMNS AND DIAGONALS SUM TO 15


4      3      8

9      5      1

2      7      6


THIS IS A 5 BY 5 MAGIC SQUARE. ALL OF THE ROWS,COLUMNS AND DIAGONALS SUM TO 65


11     10      4     23     17

18     12      6      5     24

25     19     13      7      1

 2     21     20     14      8

 9      3     22     16     15
```

```
            SCOPE 3.4                        07/14/72
21.15.49.VSIMS00
21.15.50.VSIMS,CM60000,T1000.
21.15.50. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
21.15.50.STALLATICN OF
21.15.50. SIMSCRIPT
21.15.50.SIMS.
21.16.04.    12.738 RT SECONDS LOAD TIME
21.16.11.COMPASS(I=MAPTP,L=0)
21.16.18. ASSEMBLY COMPLETE.   46600B SCM USED.
21.16.18.    4.555  CPU SECONDS  ASSEMBLY TIME.
21.16.18.MAP(OFF)
21.16.18.LDSET(LIB=SYSMISC)
21.16.19.LGO.
21.16.23.     4.191 RT SECONDS LOAD TIME
21.16.24.   END OF JOB
21.16.24.CPA      9.227 SEC.
21.16.24.PP       6.847 SEC.
21.16.24.IO       1.004 SEC.
```

RELEASE DESCRIPTION

8231 IMPORT HIGH SPEED (HS) version 1.0 operates in conjunction with INTERCOM 4.1 under SCOPE 3.4. It is coded in OSAS-A format and may be assembled on the central computer using IMPASSE (IMPORT Assembler). Off-line utility functions of card-to-printer and card-to-punch (if the configuration includes a card punch) are provided.

This system provides a remote job submission center, which accepts jobs for 6000 SCOPE 3.4 in the same unit record format as the central site and provides identical unit record output within hardware limitations.

HARDWARE CONFIGURATION

The minimum configuration for the 8231 IMPORT HS Terminal is as follows:

8090 Computer (8K is minimum core requirement for IMPORT and buffers)

161 On-Line I/O Typewriter

3681 Data Channel Converter

3447/405 Card Reader

3256/501 Line Printer

8529-B Data Set Adapter on dedicated Buffered Data Channel (BDC)

Hardware Options

A 160-A Computer may be substituted for the 8090. A 3446/415 Card Punch may be included in the 8231 Terminal configuration.

Communication Link

Each terminal is connected to the central site by a communication link consisting of two DATAPHONE Data Sets 301B or 303, and one TELPAK A communication line, or their logical and physical equivalent.

Logical Unit Assignments

Equipment numbers for the 8231 should be assigned as follows:

Card Reader	3
Card Punch	4
Line Printer	5

RELEASE MATERIALS

An updated program library for 8231 IMPORT HS 1.0 is contained on program library tape PL15.

Corrections

All eligible PSR code as published through PSR Summary No. 312 has been added to the program library.

ADDITIONAL INFORMATION

IMPASSE

IMPASSE (Import Assembler) is designed to assemble 8231 IMPORT HS programs under the SCOPE Operating System on CONTROL DATA 6000 Series computers. It is intended for installations using 8231 HS terminals which do not have adequate peripheral equipment for program assembly.

IMPASSE is coded in FORTRAN, except for the subroutine SHIFT, which is coded in COMPASS. 8231 IMPORT HS is written in standard OSAS-A format.

In the assembly process, IMPASSE reads input from file TAPE1 and produces binary output on file TAPE2 and an assembly listing, including a symbol reference table, on the OUTPUT file.

IMPASSE produces binary output for the 8231 HS terminal system.

The terminal identification in 8231 IMPORT is not used under INTERCOM 4.1; the central site assigns an ID and notifies IMPORT.

Job Input Limit Bypass

The statement ?DVT,,LP may be entered by the operator to divert all line printer output to the central site. When this statement is entered, the job input limit will be ignored and all line printer output will be diverted to the central site until the system is reloaded.

GENERAL PROCEDURES

Installation of 8231 IMPORT HIGH SPEED version 1.0 requires the following:

1. Setting of installation parameters for:

   Connect Codes
   Equipment numbers
   Number of active jobs

2. Assembling 8231 IMPORT HS.

3. Bootstrapping IMPORT on the 160A computer by using the binary deck (output from the assembly).

INSTALLATION PARAMETERS

Connect codes for terminal equipment are assembled at the beginning of 8231 IMPORT HS as shown below. The card format for IMPASSE differs from that of COMPASS; the location field begins in column 2, the operation field in column 10, and the variable field in column 15.

```
LPCN     EQU   5000   Line Printer is equipment 5 (8231IMP.80)

CRCN     EQU   3000   Card Reader is equipment 3 (8231IMP.81)

CPCN     EQU   4000   Card Punch is equipment 4 (8231IMP.82)
```

An equipment number may be changed by replacing the appropriate EQU card. If the configuration does not include a card punch, the CPCN card may be made any value by EQU, but the CPUNCH card must be made equal to 0 by EQU. The default value assumes the card punch is present.

```
CPUNCH   EQU   0      Card Punch is absent
                                             (OHP28.1)
CPUNCH   EQU   1      Card Punch is present
```

The installation may choose any value as a control on the number of jobs active at any time from the terminal. The default value is 25.

```
JOBLIM    EQU    25      Number of jobs active (QHP21.1).
```

The installation may choose either the 026 or the 029 character set at installation time only. The 026 character set is the default value. This cannot be overridden from the terminal.

To use the 026 characters, set:

```
CONTB EQU DISBCD
```

To use the 029 characters, set:

```
CONTB EQU DISB29
```

This card is located at OHP36.110.

Logical Terminal Assignments

A terminal line number is determined by the physical connection.

INSTALLATION PROCEDURES

PL15 contains three files: file 1 contains the IMPASSE assembler and 8231 IMPORT HS program library in UPDATE format, file 2 contains the installation deck, and file 3 the verification program.

The installation and verification decks as listed below can be obtained by performing the job:

```
Job card.
REQUEST(PL15,E)
REWIND(PL15)
SKIPF(PL15,1,17)
COPYBF(PL15,PUNCH,2)        Punch installation and verification decks
UNLOAD(PL15)
6/7/8/9
```

Ident I864CS, as included on the released 8231 IMPORT HS program library, activates selection of the 64-character sets at the remote site. This ident is deactivated however by an update YANK directive included in the installation deck. The installation deck as provided forces selection of the 63-character set; to select the 64-character sets, the YANK card must be removed.

1. To assemble 8231 IMPORT HS, the following job is run on the 6000 computer:

```
IMPHS,CM55000,T7000,MT02.
COMMENT.  THIS JOB CREATES A NEW PL15 AND A
COMMENT.  BINARY DECK OF 8231 IMPORT
LABEL(PL15IN,R,L=8231IMPORT,D=HI)
REQUEST(PL15,N,HI)     SCRATCH FOR NEW PL15
LABEL(PL15,W,L=8231IMPORT,D=HI)
REWIND(PL15IN,PL15)
UPDATE(F,P=PL15IN,N=PL15)
UNLOAD(PL15IN)
UPDATE(Q,P=PL15)             UPDATE IMPASSE ASSEMBLER
UPDATE(Q,C=TAPE1,P=PL15)     UPDATE IMPORT 8231 HS
RUN(S,,,COMPILE,,,40000)     COMPILE IMPASSE
RETURN(PL15)
LDSET(LIB=RUN2P3/SYSIO/NUCLEUS)
LGO.                         EXECUTE IMPASSE
COMMENT. THE BINARY OF IMPORT HS IS WRITTEN ON TAPE2
REWIND(TAPE2)
COPYBF(TAPE2,PUNCHB)         PUNCH 160A BINARY DECK
7/8/9  END OF RECORD
*/ ADD CORRECTION HERE
7/8/9  END OF RECORD
*COMPILE,IMPASSE
7/8/9  END OF RECORD
*IDENT YNK64CS
*YANK I864CS
*IDENT  IMP8231
*/  FOLLOWING CARDS ARE SAMPLE ONLY
*DELETE CHP21.1
 JOBLIM  EQU  25               ADD JOB LIMIT DESIRED
 CPUNCH  EQU  1                1 IF CARD PUNCH, 0 IF NONE
*DELETE CHP28.1
*DELETE 8231IMP.80,8231IMP.82
 CRCN    EQU  3000
 CPCN    EQU  4000
 LPCN    EQU  5000
*DELETE CHP36.110
 CCNTB   EQU  DISBCD
*COMPILE,8231IMP
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

The output from this job will be a listing of 8231 IMPORT HS plus a binary deck of IMPORT for the 160A computer. The card numbered 1 in column 80 is a one-card loader read by the bootstrap program (step 2) which loads the program following it. This should be first card in the deck. The EOR card following card 101(octal) must be discarded; the next EOR card remains as the last card. The IMPORT program cards that precede the second EOR card must be retained. The deck is ready for loading into the 8231 system.

2. Enter the following program at location 7524 in bank 1:

   a. While holding BANK CLEAR button, press MASTER CLEAR.

   b. Set ENTER/SWEEP switch to ENTER.

   c. Set REL BANK to 1 (set bit 0).

   d. Set register P to 7524.

   e. Set register Z to the following values, pressing the RUN/STEP switch to STEP once after each value is entered. The value of the P register advances by one with the addition of each entry.

```
0061
7500
6004
2200
x000      (x is card reader/equipment number)
7677
7500
6020
7202
7656
7536
```

   f. Both P and A registers should contain 7536.

   g. Neutralize all switches and MASTER CLEAR.

3. Read the IMPORT binary deck:

   a. Set REL BANK to 1.

   b. Set register P to 7524.

   c. Place the IMPORT program binary deck in card reader.

   d. Press MOTOR POWER, RELOAD MEMORY, and READY buttons.

   e. Set RUN switch to run.

   f. When all cards are read in, neutralize RUN/STEP switch.

   g. Press MASTER CLEAR and set RUN/STEP switch to RUN.

   h. Computer will halt if typewriter switches are not up or if the DSC is not operating properly.

   i. //IMPORT READY is typed out and IMPORT HS will wait for operator to specify action to be taken.

If loading of the IMPORT binary deck stops before the last card is read, start again at step 1.

A MASTER CLEAR followed by a RUN will re-initialize IMPORT.

If END is typed, IMPORT can be placed in operation by placing the RUN/STEP switch to RUN.

VERIFICATION PROGRAM

Remote terminal output from the 8231 IMPORT HS verification program which verifies communication with computer at the central site follows. This job requires less than one minute of 6400 computer clock time.

```
            SCOPE 3.4                        07/14/72
21.15.53.V823100
21.15.53.V8231,CM10000,T100.
21.15.53.
21.15.53. THIS VERIFIES CORRECT INTERACTION OF 82
21.15.53.31 IMPORT HS
21.15.53. VIA INTERCOM 4.1
```

RELEASE DESCRIPTION

Participants of the APT Long Range Program have released Standard APT Systems to the public domain; therefore, Control Data 6000 APT Version 2.2 is available to all 6000 users. Version 2.2 supersedes Version 2.1.

HARDWARE CONFIGURATION

125K(octal) of core is required to create the APT version 2.2 system, and 65K(octal) of core is required to execute it. A minimum of 65K(octal) of core is required to compile the APT version 2.2 system.

Three tape drives are required if all APT features are used; otherwise fewer tapes may suffice.

RELEASE MATERIALS

APT 2.2 is released under SCOPE 3.4 on program library tape PL16.

MODIFICATIONS

The release materials are now in UPDATE 1.2 program library format.

CORRECTIONS

All eligible PSR code (except AT20025, allowing compilation by FORTRAN Extended 4.0) published through PSR Summary 312 has been added to APT.

ADDITIONAL INFORMATION

Post-processors can be handled in one of two ways: they can be installed on the SCOPE system library by making one entry per post-processor into the LBSRCH table. The routine DISPAT can be modified to call OVERLAY and run the post-processor from a separate overlay tape.

INSTALLATION PROCEDURES

The APT release tape contains five files. File one is the program library, file two is the compiled binary in relocatable format, file three is compiled binary in absolute overlay format, files four and five are the installation decks, and file six is the installation verification program. To obtain these card decks perform a job of the type:

```
Job card.
REQUEST PL16,E.     MOUNT APT PL.
REWIND PL16.
SKIPF(PL16,3,17).
COPYBF(PL16,PUNCH,3)
UNLOAD PL16.
6/7/8/9
```

Deck APT1 will serve as a program library maintenance deck in that it allows regeneration of the APT program library and binary file. Deck APT2 will use EDITLIB to enter APT into the running system either from the release tape or from a tape created by deck APT1.

```
APT1,CM125000,TP02,T3500.                                                   001
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF APT 2.2                 002
COMMENT. THE FIRST FILE OF THE NEW PL16 WILL BE THE PROGRAM LIBRARY,        003
COMMENT. FILE TWO WILL BE THE RELOCATABLE BINARY PRIOR TO OVERLAY           004
COMMENT. FORMATION, AND FILE THREE WILL BE THE FORMED OVERLAYS.             005
LABEL(PL16IN,R,L=APT2P2*3P4,D=HI)                                           006
REQUEST PL16,N,HI.  SCRATCH FOR NEW PL16                                    007
LABEL(PL16,W,L=APT2P2*3P4,D=HI)                                             008
REWIND(PL16,PL16IN)                                                         009
UPDATE(D,F,P=PL16IN,N=PL16)                                                 010
UNLOAD PL16IN.                                                              011
RUN(S,,1001,COMFILE,LIST,LGO,500000)                                        012
REWIND PL16,LGO.                                                            013
SKIPF(PL16,1,17)                                                            014
COPYBF(LGO,PL16,1)                                                          015
REWIND LGO.                                                                 016
LOAD(LGO)                                                                   017
NOGO.                                                                       018
REWIND LIBTAP.                                                              019
COPYBF(LIBTAP,PL16,1)                                                       020
UNLOAD PL16.                                                                021
7/8/9  END OF RECORD
*/   ADD CORRECTIONS HERE                                                   023
*/                                                                          024
*/                                                                          025
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
APT2,CM55000,T7000,MT1.
COMMENT. THIS JOB TAKES THE APT V2.2 BINARY OVERLAYS AND EDITLIBS THEM
COMMENT. INTO THE RUNNING SYSTEM FROM THE PL16 TAPE.
LABEL(PL16,R,L=APT2P2*3P4,D=HI)  MOUNT PL16
REWIND,PL16.
SKIPF(PL16,2,17)
COPYBF(PL16,LIBTAP)
REWIND,PL16,LIBTAP.
UNLOAD(PL16)
EDITLIB(SYSTEM)
COMMENT. *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(APT,LIBTAP,AL=3,FL=65000,FLO=1)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(*,LIBTAP,AL=0)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

Output from the verification program for APT 2.2 follows. Approximately 10 seconds of real time are required to run this program.

```
              SCOPE 3.4                      07/14/72
21.15.55.VAPT000
21.15.55.VAPT,CM65000,T1000.
21.15.55. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
21.15.55.STALLATION OF
21.15.55. APT
21.15.55.APT.
21.17.25.      3.211 RT SECONDS LOAD TIME
21.17.35.    END OF JOB
21.17.35.CPA     1.186 SEC.
21.17.35.PP     10.986 SEC.
21.17.35.IO      1.475 SEC.
```

```
  RULED SURFACE TEST                                                              CARD NO.      TAPE NO.    2
CUTTER/    .500000
                                                                                  CARD NO.      TAPE NO.    4
 FROM /STPT                                                                       CARD NO.      TAPE NO.    6
                       X               Y               Z
                   0.0000000       4.0000000       0.0000000
 DS IS/CIR2                                                                       CARD NO.      TAPE NO.    8
                       X               Y               Z
                    .5029351       3.5808874       0.0000000
CIR2  (    0) = CIRCLE/   3.0000    1.5000    0.0000    3.0000                    CARD NO.      TAPE NO.   10
 DS IS/CIR2                                                                       CARD NO.      TAPE NO.   11
                       X               Y               Z
                    .5754674       3.6650385       0.0000000
                    .6521910       3.7480087       0.0000000
                    .7317524       3.8282618       0.0000000
                    .8140553       3.9057007       0.0000000
                    .8990003       3.9802319       0.0000000
                    .9864847       4.0517654       0.0000000
                   1.0764029       4.1202146       0.0000000
                   1.1686460       4.1854968       0.0000000
                   1.2631027       4.2475331       0.0000000
                   1.3596587       4.3062485       0.0000000
                   1.4581973       4.3615721       0.0000000
                   1.5585995       4.4134369       0.0000000
                   1.6607438       4.4617804       0.0000000
                   1.7645069       4.5065440       0.0000000
                   1.8697633       4.5476737       0.0000000
                   1.9763858       4.5851198       0.0000000
                   2.0842455       4.6188370       0.0000000
                   2.1932120       4.6487844       0.0000000
                   2.3031537       4.6749261       0.0000000
                   2.4139377       4.6972302       0.0000000
                   2.5254300       4.7156700       0.0000000
                   2.6374959       4.7302231       0.0000000
                   2.7500000       4.7408718       0.0000000
 DS IS/RS1                                                                        CARD NO.      TAPE NO.   14
                       X               Y               Z
                   2.7500000       3.6556676       3.6002807
                   2.7500000       3.2325055       5.0000479
FINI                                                                              CARD NO.      TAPE NO.   16
END OF FART PROGRAM
```

RELEASE DESCRIPTION

6000 BASIC version 2.0 runs under SCOPE 3.4. The minimum hardware configuration to operate BASIC 2.0 in batch mode is the minimum configuration required for SCOPE. The minimum hardware configuration to operate BASIC 2.0 from a terminal is the minimum configuration for INTERCOM version 4.1.

RELEASE MATERIALS

The release materials for BASIC 2.0 are provided on program library tape PL17.

CORRECTIONS

All eligible PSR code as published through PSR Summary No. 312 has been included on the BASIC program library.

ADDITIONAL INFORMATION

BASIC 2.0 is designed to be operated primarily from a terminal with a normal mode of operation of compile to core and execute. Thus, the BASIC compiler also includes the run time system which consists of one overlay. BASIC 2.0 can be operated in batch mode allowing the generation of load-and-go files, etc, from a BASIC program. The installation deck also creates a relocatable copy of the run-time system which is used only if a BASIC program is executed via the loader.

INSTALLATION PARAMETERS

BASIC 2.0 has no installation parameters. However, the following procedures may be used to make BASIC 2.0 compatible with BASIC 1.0 in the two areas where they differ.

1. In BASIC 1.0, unary minus was performed first. In BASIC 2.0, it is performed following exponentiation. Thus -2**2 = 4 in BASIC 1.0 and -4 in BASIC 2.0. To perform unary minus first, the correction identifier UNMIN should be removed with the *YANK directive.

2. In BASIC 1.0, the lower limit was zero for array dimensions; in BASIC 2.0, it is one. To allow a lower limit of zero, the correction identifier DIM should be removed with the *YANK directive.

INSTALLATION PROCEDURES

PL17 contains five files: File 1 contains the program library; file 2 contains the compiler in absolute binary; file 3 contains the compiler in relocatable binary; files 4 and 5 contain installation decks; file 6 contains the installation verification deck.

The two installation decks and the verification program may be obtained by performing the following job:

```
Job card.                     MOUNT BASIC 2.0 PL
REQUEST(PL17,E)
REWIND(PL17)
SKIPF(PL17,3,17)              SKIP PL AND BINARY DECKS
COPYBF(PL17,PUNCH,2)          PUNCH INSTALLATION DECKS
COPYBF(PL17,PUNCH,1)          PUNCH VERIFICATION PROGRAM DECK
UNLOAD(PL17)
6/7/8/9
```

Job BASIC1 is a maintenance deck which can be use to create a revised release format tape containing a modified program library and assembled binary. Job BASIC2 can be used to enter BASIC into the running system through EDITLIB either from the released tape or from the tape created by deck BASIC1. Job SCOPE3, described in section 1, then should be run to capture the running system on a deadstart tape.

```
BASIC1,CM56000,T7000,MT02.                                                   001
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF BASIC                    002
COMMENT. THE NEW PL17 WILL CONSIST OF THREE FILES                            003
COMMENT. THE FIRST FILE WILL BE THE BASIC NEWPL                              004
COMMENT. THE SECOND FILE WILL BE THE BINARY OF BASIC                         005
COMMENT. THE THIRD FILE WILL BE THE RELOCATABLE BINARY OF THE BASIC OVERLAYS 006
LABEL(PL17IN,R,L=BASIC2P0*3P4,D=HI)  MOUNT BASIC PL                          007
REQUEST,PL17,N,HI.    SCRATCH FOR NEW PL17                                   008
LABEL(PL17,W,L=BASIC2P0*3P4,D=HI)                                            009
REWIND(PL17IN,PL17)                                                          010
UPDATE(F,P=PL17IN,N=PL17,X)                                                  011
UNLOAD(PL17IN)                                                               012
COMPASS(I=COMPILE,S=CPCTEXT,L=0,B=BASIC00)                                   013
COMPASS(I=COMPILE,S=CPCTEXT,L=0,B=BASRTS,S=IPTEXT)                           014
COMPASS(I=COMPILE,S=CPCTEXT,S=IPTEXT,L=0)                                    015
COMPASS(I=COMPILE,S=CPCTEXT,L=0)                                             016
COMPASS(I=COMPILE,S=CPCTEXT,L=0)                                             017
REWIND,BASIC00.                                                              018
REWIND,BASRTS.                                                               019
REWIND(LGO)                                                                  020
COPYBR(BASIC00,BASOVER,2)                                                    021
COPYBR(BASRTS,BASOVER,9)                                                     022
COPYBF(LGO,BASOVER)                                                          023
LOAD(BASOVER)             GENERATION OF OVERLAY BASIC                        024
NOGO.                                                                        025
REWIND,BASIC.                                                                026
REWIND,BASRTS.                                                               027
REWIND(BASOVER)                                                              028
SKIPF(PL17,1,17)                                                             029
COPYBR(BASIC,PL17)                                                           030
COPYBF(BASRTS,PL17)                                                          031
COPYBF(BASOVER,PL17)                                                         032
UNLOAD(PL17)                                                                 033
*/  ADD CORRECTIONS HERE                                                     035
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
BASIC2,CM55000,T7000,MT1.                                                    001
COMMENT. THIS JOB TAKES THE BASIC BINARIES FROM PL17 AND EDITLIBS            002
COMMENT. THEM INTO THE RUNNING SYSTEM. EITHER THE RELEASED VERSION OF        003
COMMENT. PL17 OR THE VERSION CREATED BY THE DECK BASIC1 MAY BE USED.         004
LABEL(PL17,R,L=BASIC2P0*3P4,D=HI)  MOUNT PL17                                005
REWIND,PL17.                                                                 006
SKIPF(PL17,1,17)           SKIP OLDPL                                        007
COPYBR(PL17,BASIC,1)       BASIC CONTROL CARD CALLABLE ROUTINE               008
COPYBF(PL17,BASICL)        BASIC LIBRARY                                     009
REWIND,PL17,BASIC,BASICL.                                                    010
UNLOAD(PL17)                                                                 011
EDITLIB(SYSTEM)                                                              012
COMMENT. *** END OF JOB ***                                                  013
7/8/9  END OF RECORD
READY(SYSTEM)                                                                015
LIBRARY(NUCLEUS,OLD)                                                         016
REPLACE(*,BASIC,AL=3,FL=40000,FLO=1)                                         017
FINISH.                                                                      018
LIBRARY(SYSMISC,OLD)                                                         019
REPLACE(*,BASICL)                                                            020
FINISH.                                                                      021
COMPLETE.                                                                    022
ENDRUN.                                                                      023
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

The verification program supplied with the release compiles and executes two BASIC programs in batch mode. The first verifies that the compiler has been installed correctly and the second that the relocatable version of the run time system has been installed correctly.

Less than one minute is required to run the verification program deck after BASIC 2.0 has been installed.

Output from the verification program follows:

```
BASIC LIBRARY INSTALLED CORRECTLY

THIS TEST DECK USED COMPILE TO LGO MODE

THANK YOU FOR INSTALLING BASIC 2.0



BASIC-6000     (2.0)                    BASICXX               07/18/72   21.15 HRS

         20  PRINT≡BASIC LIBRARY INSTALLED CORRECTLY≡
         30  PRINT
         40  PRINT≡THIS TEST DECK USED COMPILE TO LGO MODE≡
         50  PRINT
         60  PRINT≡THANK YOU FOR INSTALLING BASIC 2.0≡
         100 END



BASIC INSTALLED CORRECTLY

THIS DECK USED COMPILE AND EXECUTE MODE



  08/23/72  SCOPE 3.4   SVSN58CLEVEL DE  08/10/72
05.45.18.VBASI4B
05.45.18.VBASIC,CM55000,T100.
05.45.18. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
05.45.18.STALLATION OF
05.45.18. BASIC 2.0
05.45.18.BASIC(L)
05.46.43.BASIC(L,B,N)
05.46.45.MAP(OFF)
05.46.45.LGO.
05.46.49.     3.075 RT SECONDS LOAD TIME
05.46.51.   END OF JOB
05.46.51.CPA      .184 SEC.
05.46.51.PP      5.211 SEC.
05.46.51.IO       .287 SEC.
```

RELEASE DESCRIPTION

ALGOL-60 Version 3.0 operates under the SCOPE 3.4 operating system. The minimum hardware configuration for ALGOL-60 is the minimum required by SCOPE.

RELEASE MATERIALS

The program library for ALGOL 3.0 is contained on the release tape PL18. This tape also contains installation materials fcr ALGOL 3.0.

MODIFICATIONS

The released default options during compilation include the F option (full ALGOL-60 checking). Under ALGOL 2.0 it was necessary to select F explicitly on the ALGOL control card. The ALGOL 2.0 default option now is represented in ALGOL 3.0 as the Q option. To achieve object code of maximum efficiency for debugging programs, it is necessary to select Q explicitly on the ALGOL control card. Installation parameters are provided for retaining the ALGOL 2.0 characteristic of producing Q option code by default.

A new control card parameter of the form, option=0, is provided for removing installation default parameters. Also, a message indicating all active options is produced after each compilation.

It is no longer necessary for every call to a code bodied procedure to provide the same number of actual parameters.

The restrictions on the kind of arithmetic expression permitted for control of an implied-fcr-loop in an I/O list have been removed.

The N option is provided to suppress array bounds checking when the F option is active.

CORRECTIONS

ALGOL 3.0 incorporates all eligible PSR corrective code published in PSR Summaries up to and including Summary Number 312.

LIMITATIONS

The SCOPE control card REDUCE cannot be used when ALGOL programs are executed, since they use the space following the program as the stack area for all variables and for input-output buffers. For the same reason the REDUCE,OFF. command should be used whenever necessary for interactive execution.

Segment mode loading (ALGOL control card options S, U, R, G) has been deactivated: attempts to use these options will result in job termination.

INSTALLATION PARAMETERS

Installation options are available in the following areas:

Default compiler options

Default execution-time options

Graphic (character set) dependency

The default compiler options are handled by a macro, ALGOL. The parameters of the ALGOL macro are the compiler control card options required to be on by default (without specification). The release tape contains the following ALGOL call:

```
ALGOL     L,X,F
```

To change these compiler defaults, the macro parameters must be modified as follows:

```
*DELETE,V3CCARD.115
          ALGOL            default options
*COMPILE,ALGOL,ALG0
```

The default execution time options are handled by a macro, OPTIONS. The parameters are execution time options which are required to be on by default (without specification). The parameters are provided in the same format as on an OPTIONS card. The release tape contains the following OPTIONS call:

```
OPTIONS     S=0,D=0,C=61,E=V
```

To change these defaults, the parameters must be modified as follows:

```
*DELETE,V3DEFB0.156
          OPTIONS          default options
*COMPILE,ALGLB00
```

If the compiler is dependent on the installation graphic set selection, IP.CSET, the punched card codes corresponding to certain ALGOL symbols (less than, left bracket, and right bracket) will differ between installations having different values of IP.CSET. The 64-character set will be card code compatible with ALGOL 2.0. The release tape is configured to override graphic set dependency, to maintain compatible source decks, and to preserve maximum character representation of ALGOL symbols. Override is achieved by the following instructions:

```
IPCHOVR     SET     1     override true
```

To establish a dependency on the value of IP.CSET the override must be turned off as follows:

```
*DELETE,V3CSET.42
 IPCHOVR     SET     0
*COMPILE,ALGOL,ALG1
```

INSTALLATION PROCEDURES

The ALGOL 3.0 release tape contains eight files. File 1 contains the program library in UPDATE 1.2 format. File 2 contains relocatable binary of the object time routines. File 3 contains the absolute compiler overlays. File 4 contains binary of ALGTEXT. File 5 contains the relocatable binary of the compiler. File 6 contains a maintenance program deck. File 7 contains an installation program deck. File 8 contains a verification program deck. The three program decks can be obtained by running the following job.

```
Job card.
REQUEST(PL18,E)              ALGOL 3.0 Release Tape
SKIPF(PL18,5,17)             Skip PL and binaries
COPYBF(PL18,PUNCH)           Punch maintenance program.
COPYBF(PL18,PUNCH)           Punch installation program.
COPYBF(PL18,PUNCH)           Punch verification program.
UNLOAD(PL18)
6/7/8/9
```

The maintenance program, ALGOL1, produces an updated copy of the first five files of the release tape; it should be used for introducing installation parameters into the program library. Deck ALGOL1 requires access to the SCOPE program library tape.

The installation program, ALGOL2, uses the tape produced by the maintenance program, or the release tape if no modifications are required, to enter ALGOL 3.0 into the running system through EDITLIB.

System text ALGTEXT is installed by execution of decks ALGOL1 and ALGOL2.

```
ALGOL1,CM56000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF ALGOL
COMMENT. THE NEW PL18 WILL CONSIST OF FIVE FILES
COMMENT. THE FIRST FILE WILL BE THE ALGOL NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY OF THE RELOCATABLE
COMMENT. CP ROUTINES AND THE PP ROUTINES
COMMENT. THE THIRD FILE WILL BE THE BINARY OF THE OVERLAYS
COMMENT. THE FOURTH FILE WILL BE THE BINARY OF ALGTEXT
COMMENT. THE FIFTH FILE WILL BE THE RELOCATABLE BINARY OF THE ALGOL OVERLAYS
REQUEST,PL1,E,HY.        LATEST SCOPE PL
UPDATE(P=PL1,N=OPL,C=0,Q)
UNLOAD(PL1)
LABEL(PL18IN,R,L=ALGOL3P0*3P4,D=HI)ALGOL OLDPL
REQUEST,PL18,N,HI.     SCRATCH FOR NEW PL18
LABEL(PL18,W,L=ALGOL3P0*3P4,D=HI)
REWIND(PL18IN,PL18)
UPDATE(F,P=PL18IN,N=PL18,X)
UNLOAD(PL18IN)
COMPASS(I=COMPILE,B=ALGOL,L=0,S=CPCTEXT,S=IPTEXT)
COMPASS(I=COMPILE,S=SCPTEXT,B=ALGO,L=0,S=CPCTEXT,S=IPTEXT)
COMPASS(I=COMPILE,S=0,X=OPL,B=ALTEXT,L=0)
REWIND(ALGO)
LOAD(ALGO)        GENERATION OF ALGOL OVERLAYS
NOGO.
REWIND(ALGOCMP)
```

```
REWIND(ALGC,ALTEXT)
REWIND(ALGOL)
SKIPF(PL18,1,17)
COPYBF(ALGC,PL18)
COPYBF(ALGCCMF,PL18)
COPYBF(ALTEXT,PL18)
COPYBF(ALGOL,PL18)
UNLOAD(PL18)
7/8/9  END OF RECORD
*/ PRODUCE RANDOM OPL
*C CPCTEXT
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
ALGOL2,CM55000,T700,MT1.
COMMENT. THIS JOB TAKES THE RELEASED VERSION OF ALGOL V3.0, OR THE
COMMENT. VERSION CREATED BY JOB ALGOL1, AND EDITLIBS THE BINARIES INTO
COMMENT. THE RUNNING SYSTEM.
LABEL(PL18,R,L=ALGOL3P0*3P4,D=HI)  MOUNT PL18
REWIND,PL18.
SKIPF(PL18,1,17)             SKIP CLDPL
COMMENT.
COPYBF(PL18,ALGLB)           ALGOL LIBRARY ROUTINES
COPYBR(PL18,ALGC,1)          ALGOL CONTROL CARD CALLABLE ROUTINE
COPYBF(PL18,ALGC)            ALGOL OVERLAYS
COPYBF(PL18,ALGTXT)          ALGTEXT
UNLOAD(PL18)
REWIND(ALGLB,ALGC,ALGO)
REWIND,ALGTXT.
EDITLIB(SYSTEM)
COMMENT. *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
*/
LIBRARY(NUCLEUS,OLD)
REPLACE(ALGOL,ALGC,AL=3,FL=40000,FLO=1).
REPLACE(ALGTEXT,ALGTXT,AL=0)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(*,ALGC,AL=0)
FINISH.
LIBRARY(SYSMISC,OLD)
REPLACE(*,ALGLB)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

Dayfile output from running the installation verification program should appear similar to the following:

```
ALGOL IS BEST

             SCOPE 3.4
21.15.55.VALGOLO
21.15.55.VALGOL,T100,CM56000.
21.15.56.ALGOL(L,X)
21.15.58.      1.653 RT SECONDS LOAD TIME
21.16.01.          0        332
21.16.01.MAP(OFF)
21.16.01.LGO.
21.16.04.      2.604 RT SECONDS LOAD TIME
21.16.06.      END OF JOB
21.16.06.CPA       .587 SEC.
21.16.06.PP       4.624 SEC.

END OF ALGOL RUN  *V3.0*
```

NORMAL OPERATING MODE

1. Press MC button.
2. Press LOAD ADDR button.
3. Enter 6400 via keyboard.
4. Press STEP button.
5. Press Bank 2 button.
6. Press OPR MEM button.
7. Press PROC MODE button.
8. Press RUN button.
9. Press ON LINE button.
10. Dial 6000 computer via dataphone.
11. When connection is made, type LOGIN on the keyboard and press SEND button.
12. Proceed as for normal INTERCOM communication.

RELEASE DESCRIPTION

SIMULA Version 1.0 runs under the SCOPE 3.4 operating system. The minimum hardware configuration required for SIMULA is the same as the minimum configuration required by SCOPE.

RELEASE MATERIALS

The updated program library for SIMULA is contained on the release tape PL19.

CORRECTIONS

All eligible PSR code as published through PSR Summary No. 312 has been added to the program library.

LIMITATIONS

Code procedures and direct files are not implemented.

The SCOPE control card REDUCE cannot be used when SIMULA programs are executed because the space following the program is used as the stack area for variables.

Segment mode loading (SIMULA control card options S, U, R, G) has been deactivated: attempts to use these options will result in job termination.

SIMULA expects compiler input to conform to the SCOPE 63 character set (IP.CSET=IP.C63). Accordingly, if the system is configured to a different character set, input must be translated to conform to the expected display code values for colon, less than, left bracket, and right bracket.

GENERAL PROCEDURES

To install SIMULA, the program library release tape can be used as input to produce a deadstart tape containing SIMULA.

INSTALLATION PARAMETERS

None.

INSTALLATION PROCEDURES

The SIMULA 1.0 release tape contains six files. File 1 contains the program library. Files 2 and 3 contain relocatable binary object time routines and absolute compiler overlays, respectively. File 4 contains relocatable binaries of the compiler. Files 5 and 6 contain the installation decks; file 7 contains the verification program deck. To obtain these decks, perform the job:

```
Job card.
REQUEST(PL19,E)
REWIND(PL19)
SKIPF(PL19,4,17)           Skip PL and Binary Files
COPYBF(PL19,PUNCH,2)       Punch Installation Decks
COPYBF(PL19,PUNCH,1)       Punch Verification Program
UNLOAD(PL19)
6/7/8/9
```

Job SIMULA1 is a maintenance deck which can be used to create a revised release format tape containing a modified program library and assembled binary. Job SIMULA2 can be used to enter SIMULA into the running system through EDITLIB either from the released PL19 or from the tape created by deck SIMULA1. Job SCOPE3, described in section 1 then should be run to capture the running system on a deadstart tape.

```
SIMULA1,CM56000,T7000,MT02.
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF SIMULA
COMMENT. THE NEW PL19 WILL CONSIST OF FOUR FILES
COMMENT. THE FIRST FILE WILL BE THE SIMULA NEWPL
COMMENT. THE SECOND FILE WILL BE THE BINARY OF THE RELOCATABLE
COMMENT. CP ROUTINES AND THE PP ROUTINES
COMMENT. THE THIRD FILE WILL BE THE ABSOLUTE BINARY OF THE OVERLAYS
COMMENT. THE FOURTH FILE WILL BE THE RELOCATABLE BINARY OF THE OVERLAYS
LABEL(PL19IN,R,L=SIMULA1P0*3P4,D=HI)  MOUNT SIMULA PL
REQUEST,PL19,N,HI.    SCRATCH FOR NEW PL19
LABEL(PL19,W,L=SIMULA1P0*3P4,D=HI)
REWIND(PL19IN,PL19)
UPDATE(F,P=PL19IN,N=PL19,X)
UNLOAD(PL19IN)
COMPASS(I=COMFILE,S=IPTEXT,B=SIMB,L=0)
COMPASS(I=COMFILE,S=PPTEXT,B=SIMLB,L=0,S=IPTEXT)
REWIND,SIMB.
LOAD(SIMB)            GENERATION OF SIMULA OVERLAYS
NOGO.
REWIND,SIMULAC.
REWIND,SIMLB.
REWIND(SIMB)
SKIPF(PL19,1,17)
COPYBF(SIMLB,PL19)
COPYBF(SIMULAC,PL19)
COPYBF(SIMB,PL19)
UNLOAD(PL19)
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
SIMULA2,CM55000,T700,MT1.
COMMENT. THIS JOB TAKES THE RELEASED VERSION OF SIMULA V1.0, OR THE
COMMENT. VERSION CREATED BY THE JOB SIMULA1, AND EDITLIBS THE BINARIES
COMMENT. INTO THE RUNNING SYSTEM FROM PL19.
LABEL(PL19,R,L=SIMULA1P0*3P4,D=HI)  MOUNT PL19
REWIND,PL19.
SKIPF(PL19,1,17)            SKIP OLDPL
COMMENT.
COPYBF(PL19,SIMLB)          SIMULA LIBRARY ROUTINES
COPYBR(PL19,SIMUC,1)        SIMULA CONTROL CARD CALLABLE ROUTINE
COPYBF(PL19,SIMUO)          SIMULA OVERLAYS
REWIND,PL19,SIMLB,SIMUC,SIMUO.
UNLOAD(PL19)
EDITLIB(SYSTEM)
COMMENT.
COMMENT. *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
*/
LIBRARY(NUCLEUS,OLD)
REPLACE(*,SIMUC,AL=3,FL=40000,FLO=1)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(*,SIMUC,AL=0)
FINISH.
LIBRARY(SYSMISC,OLD)
REPLACE(*,SIMLB)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

Output from the SIMULA verification program is listed below: The time required to run this program is approximately 5 seconds of 6400 computer clock time.

```
LIBRARY 0 EXISTS
LIBRARY 1 EXISTS
LIBRARY 2 EXISTS
LIBRARY 3 EXISTS
GARBAGE COLLECTOR EXISTS


  END OF SIMULA RUN

1 GARBAGE COLLECTIONS
```

```
           SCOPE 3.4                        07/14/72
21.15.59.VSIMULO
21.15.59.VSIMUL,T100,CM56000.
21.15.59. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
21.15.59.STALLATION OF
21.15.59. SIMULA 1.0
21.15.59.SIMULA(L,X)
21.17.15.     1.161 RT SECONDS LOAD TIME
21.17.18.MAP(OFF)
21.17.18.LGO.
21.17.23.     4.406 RT SECONDS LOAD TIME
21.17.23.     END OF JOB
21.17.24.CPA      .906 SEC.
21.17.24.PP      4.148 SEC.
```

RELEASE DESCRIPTION

1700 MSOS IMPORT HS Version 1.0 runs under the 1700 MSOS 3.0 operating system and operates in conjunction with INTERCOM 4.1 which runs under the SCOPE 3.4 operating system. ` 1700 MSOS IMPORT HS consists of modules written in 1700 Assembly language.

The 1700 computer is used as a remote terminal center which accepts jobs for 6000 SCOPE in the same unit record format as the central site and provides identical unit record output within hardware limitations.

Six simultaneous data streams, in addition to operator messages, are possible between the terminal and the central site. The data streams are restricted only by hardware configurations.

RELEASE MATERIALS

The 1700 MSOS IMPORT HIGH SPEED release consists of an UPDATE format program library tape including both MSOS IMPORT and the CLASS assembler.

Product Description

A brief description of each module of IMPORT HS designed to operate with MSOS 3.0 is given below:

IMMINT

IMMINT is the standard MSOS Manual Interrupt module, which has been modified to divert all commands prefixed with a question mark (?) to CRIMPT (the Core Resident IMPORT) module. It also has been modified to accept messages of the length required by IMPORT HS.

CRIMPT

CRIMPT first determines whether or not IMPORT HS has been loaded. If so, CRIMPT diverts control to IMPORT. If not, CRIMPT loads IMPORT.

IMPORT

IMPORT goes through initialization and starts central to terminal message processing.

MESIMP

MESIMP processes all terminal operator type-in messages. Directives are constructed and transmitted in accordance with typed-in instructions. Data streams are also defined, initialized, and terminated via this module.

TRAFIC

TRAFIC controls orderly routing of transmissions to and from the central site. Special directives and operator messages are given priority over data stream transmissions.

TCPROC

TCPROC controls terminal to central site data streams. These streams allow jobs to be submitted from either tape or cards. These jobs are identical in format to those submitted at the central site.

CTPROC

CTPROC processes all central site to terminal output (card punch and/or line printer) data.

TPR

TPR is the tape to printer utility routine. Data from the central site intended for the printer can be assigned to a tape unit and later printed out on the printer at the 1700 site.

TPU

TPU is the tape to punch utility routine. Data from the central site can be assigned to a tape unit and later punched cards can be produced at the 1700 site.

CTA

CTA is the card to tape utility routine. This routine copies card images onto magnetic tape so that data can be prepared at the 1700 site for subsequent processing at the central site.

SPACE

The MSOS SPACE program must be modified if normal MSOS job processing is to occur concurrently with IMPORT HS processing. This modification adds to allocatable core the amount of core expected to be used by IMPCRT HS. If this modification is not made, IMPORT HS will cause core-swapping which locks out job processing.

DR1747

This program operates as a normal MSOS device driver to handle read and write requests through the 1747 Multiplexer to the central site computer.

HARDWARE CONFIGURATION

The suggested minimum configuration for the 1700 MSOS IMPORT HS Terminals follows. This configuration does not allow concurrent job processing.

1704 4K Computer

1708 8K Storage Module

1705 Interrupt Data Channel

1706 Buffered Data Channel

1747 Multiplexer

1711 or 1713 Teletypewriter

1726/405 or 1728/430 Card Reader/Punch

1740/501 or 1742/HR300 Line Printer

1573 Line Sync Clock

1750 Data and Control Terminal

1738 Disk Controller

853/854 Disk Pack

To allow concurrent job processing, the above minimum configuration may include the following; and depending on the number of defined data streams, an extra 4K of core storage may be required.

1721/1722 Paper Tape Reader

1723/1724 Paper Tape Punch

Additional Equipment

MSOS IMPORT HS will allow up to six concurrent data streams which may use any combination of the following:

- 601/1731/1706 Buffered Magnetic Tape Units
- 608/609/1732/1706 Buffered Magnetic Tape Units
- 1726/405 Card Reader
- 1728/430 Card Reader/Punch
- 1742/HR300 Line Printer
- 1740/501 Line Printer
- 1729-2 Card Reader
- 1729 Card Reader

Communication Link

Each terminal is connected to the central site by a communication link consisting of two DATAPHONE Data Sets 301B or 303, and one TELEPAK A communication line or their logical and physical equivalent.

Logical Unit Assignments

Any logical unit assignments acceptable to MSOS are suitable for IMPORT HS.

MODIFICATIONS AND DEFICIENCIES

A proper base has been established for introduction of 64-character set code.

Limitations

Before installation of 1700 MSOS IMPORT HS is attempted, all PSR summaries for INTERCOM 4.1 and MSOS 3.0 should be reviewed.

16K SYSTEM

SYSBUF Modification Kit - The 28 extra entries added to the scheduler stack SCHSTK are not required in a 16K system.

When allocatable core is oversubscribed, jobs are written out on disk; and core used by these jobs is made available for allocation.

In a 16K system, job processing is swapped out as soon as IMPORT is called. If several data streams are defined and released, small unusable fragments of allocatable core may remain. Defining the remaining data streams could result in the message //NO CORE before all six data streams are defined. Although the total fragments of core available could complete the remaining stream, not enough space is available in one block.

This same situation can occur if magnetic tape reading or writing takes place before all six data streams are defined. If IMPORT is called during job processing, some core is not available to IMPORT.

In a 16K system, there may not be sufficient space to load the mass storage resident manual interrupt processor, MIPRO, which processes non-IMPORT commands. If the operator enters a non-IMPORT command, or mistakenly omits the ? prefix from an IMPORT command, MSOS attempts to load MIPRO to process the message. The core allocator finds no available core, and stacks the request until core is available. Under MSOS, such requests are not timed out and the request is never honored. No further manual interrupt is possible, so the operator cannot communicate with IMPORT, and a data stream cannot be released to make additional core available. This situation can also occur in any system where job processing has been locked out and a statement to job processing has been input.

In such a case, the operator can change the equipment number on the Data Set Controller to retain the active background job processing. This causes the 1747 driver to reject and type DSC REJ. Then IMPORT types //COM LOST, releases its core, and types IMPORT OFF. Core will be available then to complete the processing of the TTY command and job processing continues.

INTERCOM treats this situation as though the IMPORT operator entered ?END. INTERCOM rewinds all attached output files putting them back into the output queue and discards all incomplete input files.

ADDITIONAL INFORMATION

1. 1726/405 Card Reader Operation

   When a 1726/405 Card Reader is used, IMPORT HS cannot read the last card in the hopper. To ensure that the job is read completely, an extra end-of-file card (6/7/8/9 punches in column one) should be added to the end of the deck. IMPORT HS ignores multiple end-of-file cards, so extras will cause no difficulty.

2. Job Completion Messages

   When a job finishes printing, 1700 MSOS IMPORT HS will produce the following messages on the teletypewriter:

   //LUxx jobname DONE   where xx is the logical equipment device used

3. Job Input Limit Bypass

   Only one terminal operator command affects the job limit. The various formats of the ?DVT command follow:

Command	Effect on job counter
?DVT	reset to zero
?DVT,JOBNAME	decreased by one
?DVT,JOBNAME,LP	decreased by one
?DVT,,LP	counter is bypassed (no limit)
?DVT,JOBNAME,CP	no effect
?DVT,,CP	no effect

4. Utility Routines

   The utility routines are executed as normal unprotected jobs and normally are resident on the program library. For installation procedures, see the general description.

5. If an output device is defined and activated, but no output is available for the device, this message is typed out.

   //LUxx WAITING ON CENTRAL

   This message also may be output periodically when no visible activity is taking place to let the user know the output streams are still functioning.

6. When ?WAIT is entered, the //OK. response indicates the wait bit has been set for the data stream; however, the data stream cannot enter the wait condition until any active requests are completed. Therefore, during the WAITING ON CENTRAL condition, a maximum of 30 seconds may elapse before ?WAIT becomes effective. The ?GO command is invalid during this period. WAITING ON CENTRAL is the delay required for the data stream driver to time out.

GENERAL PROCEDURES

1700 MSOS IMPORT HS is composed of modules written in 1700 Assembly Language. It runs under the standard MSOS 3.0 operating system. To install MSOS IMPORT, modules must be added to an existing working MSOS system. Utility functions are provided.

INSTALLATION PROCEDURES AND PARAMETERS

The 1700 MSOS IMPORT HS release tape consists of 5 files. File 1 contains the source of MSOS IMPORT in Update 1.2 program library format. File 2 contains Update 1.2 program library format source of CLASS. A definition how to use CLASS to assemble MSOS IMPORT source on the 6000 is included at the end of this chapter. File 3 contains BCD cards of the SYSBUF Modification Kit to be used in modifying a workable MSOS SYSBUF to accommodate MSOS IMPORT. File 4 contains binary of MSOS IMPORT. File 5 contains a MSOS IMPORT Verification program.

To obtain the contents of files 3, 4, and 5 in card form, perform a job of the following for the 6000.

```
Job card.
REQUEST PL20,E.
REWIND PL20.
SKIPF(PL20,2,17)
COPYBF(PL20,PUNCH)
COPYBF(PL20,TEMP)
COPYBF(PL20,PUNCH)
REWIND TEMP.
DISPOSE(TEMP,P8)
UNLOAD PL20.
6/7/8/9
```

To install MSOS IMPORT, an existing working MSOS system containing the desired hardware configuration must be available. MSOS IMPORT modules are added to these systems. Modify the existing MSOS system as follows:

Replace the MSOS module MINT with the IMPORT variant of the module IMMINT, the Manual Interrupt statement decoder. To Core Resident MSOS add the routine, CRIMPT. The following items should be added to MSOS SYSBUF (these cards are contained in the SYSBUF Modification Kit):

```
IMPSPD    ADC    DUMMY
IMPTC     ADC    DUMMY
IMPCT     ADC    DUMMY
IMSTR1    ADC    DUMMY,DUMMY,DUMMY,DUMMY,DUMMY
IMLAST    ADC    DUMMY
```

The cards specified above are contained in the SYSBUF Modification Kit; they are added as the last items of table LOG1A after the card EQU LAST(*) and before the card NUMLU.

The card with the comment IMPORT LOG1 ENTRIES should be added at the end of the LOG1 table.

Three cards, the first of which has the comment IMPORT LOG2 ENTRIES, should be added at the end of the table LOG2.

Twenty-eight cards, the first of which has the comment 28 EXTRAS FOR IMPORT, should be added at the end of the scheduler stack SCHSTK, (before NUM 0,0 $FFFF,0 scheduler stack entry 24).

Six cards, the first of which has the comment UP TO 6 ENTRIES MAY BE PLUGGED BY IMPORT, should be added to the end of the diagnostic timer table, DGNTAB.

The remaining cards of the IMPORT SYSBUF Modification Kit should be added at the end of SYSBUF. Some of the following may require change depending on the site configuration of the remote terminal:

EQU IMPCOM(4)	IMPORT comment device logical unit
EQU R26(0)	If R26=0, all BCD cards are read as if punched by an 026 keypunch. However, by punching the digits 29 in columns 79-80 of a job card or a 7/8/9 end of record card, the remaining BCD cards in that job are read as if punched by an 029 keypunch. The mode can be changed again by punching the digits 26 in another 7/8/9 end of record card. If R26=1, all BCD cards are read as if punched by an 029 keypunch. However, by punching the digits 26 in columns 79-80, the remaining BCD cards in that job are read as if punched by an 026 keypunch. The mode can be changed again by punching the digits 29 in another 7/8/9 end of record card.
EQU P26(0)	If P26=0, all Hollerith cards will be punched in 026 keypunch format. If P26=1, all Hollerith cards will be punched in 029 keypunch format.
EQU IMPRI(6)	Priority level at which IMPORT is to run
EQU LUDSC(22)	Logical unit number of Data Set Controller
EQU PRIDSC(10)	Priority level at which the Data Set Controller is to run
EQU BSIZE(165)	Default transmission buffer size; the only alternate is 325
IMTID ALF 1,ME	Default terminal ID. Although this parameter was essential for communication with EXPORT HS, it is ignored in communication with INTERCOM.

INSTALLATION OF THE 1747 DSC DRIVER (DR1747)

An entry of the form, ADC PH1747, should be added to table LOG1A at the positions corresponding to the interrupt line used by the 1747. Entry of the form, NUM 0, should be added to the table LOG1.

An entry of the form, NUM $FFFF, should be added to the table LOG2.

The mask table, MASKT, should be modified by adding a 1 bit to the interrupt mask for all priorities lower than those at which the 1747 is to run.

The entry, ADC PH1747, should be added to the table, DGNTAB.

The code for the 1747 Interrupt Response Routine and PHYSTAB follows:

```
        ENT     I1747
I1747   LDQ     =XPH1747       Interrupt response routine
        JMP*    (PH1747+2)
        EXT     IN1747
        EXT     CN1747
        EXT     EX1747
PH1747  NUM     $120A      0
        ADC     IN1747     1   Initiator entry
        ADC     CN1747     2   Interrupt entry
        ADC     EX1747     3   Hang up entry
        NUM     -1         4   Diagnostic clock counter
        ADC     0          5   Logical Unit or logical equipment
                                   designator
        ADC     0          6   PARAM conv.
        ADC     $1401      7   Q SETTING  WES:
                               W: converter (bits 11, 12)
        NUM     $641           E: equipment code (bits 7-10)
                               S: director (bit 0-1)
        ADC     0          9   Switches
        ADC     0         10   Core
        ADC     0         11   LAST LCC + 1
        ADC     0         12   Status
        ADC     0,0,0  13-15   Used by MSOS
        ADC     0         16   TEMP
ISAVE   ADC     0         17   ISAVE
        RTJ-    ($F4)     18   TTYREQ Teletypewriter request for diagnos-
                                      tic printouts after equipment
                                      failures
        RTJ     $C88      19
        ADC     TTYCMP    20
        ADC     0         21   TTYTHD
        ADC     4         22
        ADC     0         23
        ADC     0         24
        JMP-    ($EA)     25
TTYCMP  LDA*    ISAVE     26
        STA-    I         27
        JMP+    0         28 and 29
```

LOCORE MODIFICATIONS

The following card should be added to the program LOCORE.

```
EXT I1747
```

In addition, the interrupt slot for the interrupt line used by the 1747 should be set up as follows:

```
LINExx     NUM 0
           RTJ- ($FE)
           NUM PRI priority level
           ADC I1747
```

PRIORITY LEVEL ASSIGNMENTS

The multiplexer driver should run at a priority level higher than other I/O devices not subject to malfunction when their interrupts are not serviced promptly. It should be lower than that used by devices subject to such malfunction. The priority level at which IMPORT is run should be lower than that used by the Core Allocator, but not so low as to cause unnecessary delays at the central site.

IMPORT MASS STORAGE RESIDENT MODULE

This module is composed of the following routines:

IMPORT

TRAFIC

MESIMP

TCPROC

CTPROC

To add this module, the following modifications must be made to the system initializer input deck.

*YM,IMPORT,xx

where xx is the ordinal number of this Mass Storage Resident module.

This card should be added to the other *YM cards at the beginning of the deck after the decks comprising the Mass Storage Resident module preceding IMPORT.

*M IMPORT

This is followed by the binary cards for the above named five IMPORT programs. Program IMPORT must be first.

*S,xx,p,M

xx is the ordinal of the IMPORT module.

p is the priority level at which IMPORT runs

letter M (not a variable)

This card should be added to the other *S cards at the end of the initialization input deck. LIBEDT processes these cards by assigning the specified priority level to the module.

If simultaneous job processing is required, allocatable core area, AREAC, in program SPACE must be increased by the amount of core IMPORT will use when the streams have been defined. IMPORT occupies 5100 (decimal) core locations.

Each line printer stream will require:

- 1 100-word PHYSTAB
- 2 69-word device buffers
- 2 165- or 325-word transmission buffers

Each card punch stream will require:

- 1 100-word PHYSTAB
- 2 60-word device buffers
- 2 165-or 325-word transmission buffers

Each card read stream will require:

- 1 83-word PHYSTAB
- 2 60-word device buffers
- 2 165- or 325-word transmission buffers

Assembly Options

The user has the option of changing the transmission buffer size default value at installation time.

TYPE-IN	ACTION
?IM carriage return	TRANSMISSION BUFFER SIZE=165
?IM,,325	TRANSMISSION BUFFER SIZE=325

JOBMAX contains the maximum number of jobs allowed in the system at any one time. When this number is reached, IMPORT will stop reading cards until a job is printed and leaves the system. The value in JOBMAX is arbitrary and can be made significantly larger or smaller to suit the needs of the application. The standard value is 25(decimal). To modify this value by assembly, change the NUM value for JOBMAX. To modify it by using hexadecimal corrections, set the contents of the last cell in SYSBUF to contain the desired maximum value. This value may range from 1 to 32k.

The binary deck as provided is correctly set up to handle the 63-character set. If either of the 64-character sets is to be selected, modules MESIMP, TCPROC, and CTPROC must be reassembled and new binary decks produced, including the following changes:

```
*D  CHAROPT.6

C64I  EQU  C64I(1)

*C  MESIMP, CTPROC, TCPROC
```

Procedures for preparation of input to the MSOS Macro Assembler 2.0 are described in Section 25, 1700 SOURCE. Procedures for use of the CLASS Assembler on the CDC CYBER/70 are described at the end of this section.

Utility Routine Installation

Press: AUTO-LOAD

1. Move the STEP/RUN switch to RUN

   Message: PP

   Type: *

   Press: Carriage Return

   Press: Manual Interrupt

   Message: MI

   Type: *LIBEDT

   Message: LIB
   IN

2. Load binary card decks in card reader:

   Type: *K,I1u   1u is input device

3. LIBEDT responds when loading is complete.

   Message: IN

   Type: *L, n...n   n...n is program name at execution time

4. To execute the program:

   Type: *n...n

   This loads the utility being called into core and begins execution.

1747 DSC CE Diagnostics

If any of the following messages appear on the teletypewriter, a customer engineer should be consulted.

BDC BUSY

DSC BUSY

DSC NOT READY

NO CARRIER

TEST MODE

DSC REJECT

BDC NOT READY

(BDC is Buffered Data Channel)
(DSC is Data Set Controller)

These diagnostics are produced primarily for the customer engineer. They generally indicate a hardware failure rather than a software failure.

Logical Terminal Assignments

A terminal line number is determined by the physical connection. The unique two-character identifier associated with each line is established by INTERCOM when the terminal establishes communication.

SYSBUF MODIFICATION KIT

A complete listing of the SYSBUF Modification Kit is provided for general information only.

```
MSOSIMF,CM10000,T50,P17.
COMMENT.  THIS JOB VERIFIES 1700 MSOS AND INTERCOM HIGH SPEED INSTALLATION
7/8/9  END OF RECORD
       NAM  MOD KIT FOR SYSBUF
*THIS MODULE IS NOT INTENDED TO BE ASSEMBLED
*IT CONTAINS CARDS TO BE ADDED TO SYSBUF WHEN MSOS IMPORT
*IS ADDED TO A SYSTEM
*
*        THE FOLLOWING 6 CARDS ARE TO BE ADDED AS THE LAST 9 ENTRIES IN LOG1A
IMFSPD   ADC      DUMMY                IMPORT LOG1A ENTRIES
IMFTC    ADC      DUMMY
IMFCT    ADC      DUMMY
IMSTR1   ADC      DUMMY,DUMMY,DUMMY
         ADC      DUMMY,DUMMY
IMLAST   ADC      DUMMY
*
*      THESE ENTRIES ARE TO BE ADDED TO LOG1
         NUM      0,0,0,0,0,0,0,0,0    IMPORT LOG1 ENTRIES
*
*
*      THESE ENTRIES ARE TO BE ADDED TO LOG2
         NUM      $FFFF,$FFFF,$FFFF    IMPORT LOG2 ENTRIES
         NUM      $FFFF,$FFFF,$FFFF
         NUM      $FFFF,$FFFF,$FFFF
*
*
*      THESE  ENTRIES ARE TO BE ADDED TO THE SCHEDULER STACK-SCHSTK
         ADC      0,0,*+2,0            28 EXTRAS FOR IMPORT
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
         ADC      0,0,*+2,0
```

```
*
*
*       THESE ENTRIES ARE TO BE ADDED TO THE DIAGNOSTIC TIMER TIMER TABLE-DGNTAB
        NUM  $FFFF          UP TO 6 ENTRIES MAY BE PLUGGED BY IMPORT
        NUM  $FFFF
        NUM  $FFFF
        NUM  $FFFF
        NUM  $FFFF
        NUM  $FFFF
*
*
*       THESE ENTRIES ARE THE 1747 DRIVER INTERRUPT RESPONSE ROUTINE AND
*       PHYSTAB-NOTE THAT ENTRY 7 SHOULD BE SET TO THE CONVERTER AND
*       EQUIPMENT NUMBER AT YOUR SITE
        ENT  I1747
I1747   LDQ  =XPH1747
        JMP* (PH1747+2)
        EXT  IN1747
        EXT  CN1747
        EXT  EX1747
PH1747  NUM  $120A     0
        ADC  IN1747    1
        ADC  CN1747    2
        ADC  EX1747    3
        NUM  -1        4
        ADC  0         5 LU
        ADC  0         6 PARAM CONV.
        ADC  $1401     7 Q FOR EQUIP 8 BDC 1
        NUM  $641      8
        ADC  0         9
        ADC  0         10
        ADC  0         11 LAST LOC+1
        ADC  0         12 STATUS
        ADC  0         13 NO MASS MEM
        ADC  0         14 NO MM LENGTH
        ADC  0         15 RETURN INFORMATION FNR
        ADC  0         16 TEMP
ISAVE   ADC  0         17 ISAVE
        RTJ- ($F4)     18 TTYREQ
        NUM  $C88      19
        ADC  TTYCMP    20
        ADC  0         21 TTY THREAD
        ADC  4         22
        ADC  0         23
        ADC  0         24
        JMP- ($EA)     25
TTYCMP  LDA* ISAVE     26
        STA- I         27
        JMP+ 0
```

```
*
*       THE FOLLCWING ITEMS SHOULD BE ADDED AT THE END OF SYSBUF
          ENT         IMPSPD,IMPCT,IMSTR1,IMLAST,IMPTC
        ENT   DSCLU
          ENT         DSCPRI
          ENT         IMCOM
          ENT         IPRI,IMBFSZ,IMTID
          ENT         UNDEF
          ENT         P2629,R2629
UNDEF     ADC         DUMMY
IPRI      ADC         IMPRI
DSCPRI    ADC         PRIDSC
IMBFSZ    ADC         BSIZE
P2629     ADC         P26
R2629     ADC         R26
IMCOM     ADC         IMPCOM
DSCLU   ADC   LUDSC
           ENT    SYNCLI
        ENT   RETLIM
        ENT   JOEMAX
*
*
*       THE FOLLOWING ITEMS ARE DEFAULT CONDITICNS
*       THEY SHCULD BE CHANGED TO SUIT YOUR SITES REQUIREMENTS AND CONFIGURATION
          EQU         IMPCOM(4)             IMPORT COMMENT DEVICE
          EQU         R26(0)                026-029 READ IND.(0=026)
          EQU         P26(0)                026-029 PUNCH IND.(0=026)
        EQU   IMPRI(6)
          EQU         LUDSC(22)
          EQU         PRIDSC(10)
          EQU         BSIZE(165)
IMTID   ALF   1,ME        DEFAULT IMPORT TERMINAL ID
SYNCLI     NUM    -10           SYNCHERR LIMIT
RETLIM NUM   -10
JOEMAX NUM   25
*
        END
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

CLASS ASSEMBLER

The CLASS ASSEMBLER is a CDC Cyber 70 version of the 1700 Macro Assembler. CLASS, written in COMPASS, provides assembly listings and binary object decks from 1700 Assembly Language source.

INSTALLATION

The source of the CLASS Assembler is on the second file of the 1700 MSOS IMPORT source program library. The following deck is an example of a SCOPE 3.4 job to install CLASS and its macro skeleton as permanent files in the SCOPE 3.4 system.

```
CLASS,CM70000,T770,MT1.
REQUEST(MT,E) 1700 MSOS IMPORT RELEASE TAPE
REQUEST,CLASS,*PF.
REQUEST,SMAC17,*PF.
SKIPF(MT,1,15)      SKIP MSOS IMPORT PL
COPYBF(MT,PL)       MOVE CLASS PL
UPDATE(A,P=PL,N=OLDPL)
UNLOAD(MT)
UPDATE(Q)
COMPASS(I,L=0,B=TEXT,S=0) ASSEMBLE KRONTXT
UPDATE(Q)
COMPASS(I,L=0,B=CLASS,G=TEXT) ASSEMBLE CLASS
CATALOG(CLASS,CLASS,ID=CLASS,XR=CLASS)
UPDATE(Q)
CLASS(I,M=SMAC17,L=0) ASSEMBLE MACRO SKELETON
CATALOG(SMAC17,SMAC17,ID=CLASS,XR=CLASS)
7/8/9
*C KRONTXT
7/8/9
*C CLASS
7/8/9
*C SMAC17
7/8/9
6/7/8/9
```

ASSEMBLY OPTIONS

The following CLASS features are additional to those found in the 1700 MACRO Assembler:

1. Banner page at the head of each listing including:

    Address and length of program

    Address of END card

    All entry points with their addresses in alphabetical order

    All external symbols in alphabetical order

    All BZS/BSS blocks and addresses

2. TIDY option which lists the free-form input in specified columns on the output listing.

3. Capability to print list control (LST,SPC,NLS,etc.) cards.

4. Option to omit from the listing comment cards designated by an * in column 1.

5. Option to treat EJT cards as four spaces instead of a page eject. When a page is more than 3/4 full, the listing will be spaced to top-of-form.

6. Option to suppress the listing of conditional code which is not generated.

7. Option to suppress the listing of all but the first word of machine code generated by multiword instructions.

8. Full cross-reference map with page/line number references.

9. Option for a short cross reference map including only the value of the symbol.

CLASS CONTROL CARD PARAMETERS

Input

Not specified	input on file INPUT
I	input on file COMPILE
I=fname	input on file fname

Output

L	full list on file OUTPUT
L=fname	full list on file fname
L=0	no list output

Binary

B or not specified	binary output on file LGO
B=fname	binary output on file fname
B=0	no binary output

Binary output to be punched should be disposed to a file with free-form punch disposition (P8).

TIDY tab columns

Not specified	if list options specify T, tab columns will be 11-18-30
T=nnmmoo	TIDY columns are nn, mm, and oo; must be a six digit number. Example: T=081322 sets tab columns 8,13, and 22.

List options

LO = any combination of up to 7 of the allowable list options shown below. If more than 7 options are required, more than one LO= may appear on the control card.

If no LO options appears, BMR and T are selected. When LO is specified, only the options requested are selected.

LO=B	list BZS/BSS blocks on banner page
C	list program list controls (EMT,SPC,etc.)
D	suppress comment cards
E	process EJT as a page eject
I	list code skipped by an IFA pseudo op
M	list all entries on multiword entries
R	list full reference map
S	list abbreviated reference map
T	tidy the list file into specified tab columns
X	suppress macro expansion

EXAMPLE ASSEMBLIES

To assemble DR1747 from the 1700 MSOS IMPORT source program library:

```
JOB,CM70000,TP1.
REQUEST(OLDPL,E) 1700 MSOS IMPORT SOURCE
UPDATE(Q)
ATTACH(CLASS,CLASS,ID=CLASS)
ATTACH(SMAC17,SMAC17,ID=CLASS)
CLASS(I,G=SMAC17)
DISPOSE(LGO,P8)
7/8/9
  ...Any corrections
*C DR1747
7/8/9
6/7/8/9
```

To assemble a 1700 program with source on cards, TIDY tab columns 8,13,22, and no binary.

```
JOB,CM70000.
ATTACH(CLASS,CLASS,ID=CLASS)
ATTACH(SMAC17,SMAC17,ID=CLASS)
CLASS(B=0,T=081322,LO=BTRM)
7/8/9
  source program cards
7/8/9
6/7/8/9
```

LIMITATIONS

If a name with more than seven characters appears on an EXT card, the symbol will be treated as an entry point.

RELEASE DESCRIPTION

FORTRAN Version 2.3 runs under the SCOPE 3.4 operating system. FORTRAN requires the same minimum hardware configuration as SCOPE.

RELEASE MATERIALS

The released program library for FORTRAN 2.3 is known as PL21.

CORRECTIONS

All eligible PSR code published through PSR Summary 314 has been added to the FORTRAN program library. Additionally, idents FTC0147A (Summary 316) and FT3K125A (Summary 318) have been incorporated.

LIMITATIONS

All known deficiencies or limitations are reported in the PSR Summary as unanswered PSR's or as mentioned in memoranda accompanying this release.

INSTALLATION PARAMETERS

In the standard release of PL21, binary blocking code has been assembled into the I/O routines; the default condition for unformatted files is set to unblocked. To set the default condition for unformatted files to blocked, the identifier RM1567B should be removed by using the *YANK control card. If the identifier RM1567A is removed by *YANK, the code associated with binary blocking will not be assembled. The following two jobs demonstrate these assembly options:

JOB 1 Sets unformatted files default condition to blocked

```
JOB(CM60000,T400,MT1)
REQUEST(PL21,E)                    FORTRAN 2.3 program library
UPDATE(Q,P=PL21)
RETURN(PL21)
COMPASS(I=COMPILE,B=REPLACE,L=0,S=CPCTEXT,S=IPTEXT)
EDITLIB(SYSTEM)
7/8/9

*IDENT,BLOCK
*YANK,RM1567B
*COMPILE,SYSTEM
7/8/9

REWIND(REPLACE)
READY(SYSTEM,OLD)
LIBRARY(RUN2P3,OLD)
REPLACE(*,REPLACE)
FINISH.
COMPLETE.
ENDRUN.
6/7/8/9
```

JOB 2 Prevents assembly of code for binary blocking

```
JOB(CM60000,T400,MT1)
REQUEST(PL21,E)                    RELEASE TAPE
UPDATE(Q,P=PL21)
RETURN(PL21)
COMPASS(I=COMPILE,B=REPLACE,L=0,S=CPCTEXT,S=IPTEXT)
EDITLIB(SYSTEM)
7/8/9

*IDENT,NOBLOCK
*YANK,RM1567A
*COMPILE,IODEFS,SYSTEM,BACKSP,ENDFIL,IFENDF,INPUTB
*COMPILE,OUTPTB,REWINM,SIO$,FTNBIN

7/8/9
REWIND(REPLACE)
READY(SYSTEM,OLD)
LIBRARY(RUN2P3,OLD)
REPLACE(*,REPLACE)
DELETE($SIO$$$)
FINISH.
LIBRARY(SYSIO,OLD)
REPLACE($SIO$$$,REPLACE)
FINISH.
COMPLETE.
ENDRUN.
6/7/8/9
```

A trigger value is used to trigger an access to an I/O device for blocked binary files. On input, it represents the number of buffer words that must be available before a read buffer request is issued; on output, it is the number of words that must be ready for output before a write buffer request is issued.

The trigger value is set to a fixed percentage of the buffer length; however, if that percentage (or the remainder of the buffer) is less than one PRU, the PRU size is taken as the trigger value. The default is set to 80 percent; to change it, the value of the micro TRIGGER in SIO$ should be changed. For example, to change the triggering percentage to 50 percent:

```
*DELETE,SCU316M.94
TRIGGER MICRO 1,0,/50/
*COMPILE SIO$
```

The trigger value must be less than 100.

INSTALLATION PROCEDURES

The FORTRAN 2.3 release tape contains five files: file 1, the program library; file 2, assembled binary; file 3, installation deck V23RUN1; file 4, installation deck V23RUN2; and file 5, a verification program.

V23RUN1 is a maintenance deck which can be used to create a revised program library and binary file. V23RUN2 can be used to enter FORTRAN 2.3 into the running system from either the released tape or a tape created by V23RUN1. Following completion of V23RUN2, job SCOPE3 (see Section 1) can be run to create a deadstart tape of the running system.

To obtain the decks included as files 3-5, perform a job of the type:

```
Job card.
REQUEST(PL21,E)
REWIND(PL21)
SKIPF(PL21,2,17)          SKIP PL AND BINARY
COPYBF(PL21,PUNCH,3)      PUNCH INSTALLATION AND VERIFICATION DECKS
UNLOAD(PL21)
6/7/8/9
```

```
V23RUN1,CM55000,T7000,MT02.                                                       001
COMMENT. THIS JOB UPDATES AND CREATES THE BINARY OF RUN 2.3                        002
COMMENT. THE FIRST FILE OF THE NEW PL21 WILL BE THE NEWPL                          003
COMMENT. THE SECOND FILE WILL BE THE ABSOLUTE BINARY OF THE OVERLAYS AND THE       004
COMMENT. THE FIRST FILE OF THE NEW PL21 WILL BE THE NEWPL                          003
COMMENT. THE SECOND FILE WILL BE THE ABSOLUTE BINARY OF THE OVERLAYS AND THE       004
COMMENT. BINARY OF THE RELOCATABLE ROUTINES                                        005
REQUEST PL2,E,HI.   MOUNT LATEST COMPASS PL                                        006
REWIND PL2.                                                                        007
UPDATE(A,P=PL2,N=COMPCOM)                                                          008
UNLOAD PL2.                                                                        009
LABEL(PL21IN,R,L=RUN2P3*3P4,D=HI) RUN 2.3 OLDPL                                    010
REQUEST,PL21,N,HI.     SCRATCH FOR NEW PL21                                        011
LABEL(PL21,W,L=RUN2P3*3P4,D=HI)                                                    012
REWIND(PL21IN,PL21)                                                                013
UPDATE(F,P=PL21IN,N=PL21,E,X)                                                      014
```

```
UNLOAD(PL21IN)
COMPASS(I=COMPILE,S=CPCTEXT,S=IPTEXT,L=0)    ACGOER-RECOVR$
COMPASS(I=COMFILE,B=RUNN,S=CPCTEXT,S=IPTEXT,L=0)  RUN-Q8QDIAGP
REWIND(RUNN)
REWIND(LGC)
SKIPF(PL21,1,17)
COPYBF(LGO,PL21)
BKSP(PL21,1)
COPYBF(RUNN,PL21)
UNLOAD(PL21)
7/8/9  END OF RECORD
*/  ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
V23RUN2,CM55000,T700,MT1.
COMMENT. THIS JOB EDITLIBS THE RUN V2.3 BINARIES INTO THE RUNNING
COMMENT. SYSTEM FROM THE PL21 TAPE. EITHER THE RELEASED VERSION OR THE
COMMENT. VERSION CREATED BY DECK V23RUN1 MAY BE USED.
LABEL(PL21,R,L=RUN2P3*3P4,D=HI)  MOUNT RUN V2.3 PL
REWIND,PL21.
SKIPF(PL21,1,17)             SKIP OLDPL
COPYBF(PL21,RUNL)            RUN V2.3 BINARIES
REWIND,PL21,RUNL.
UNLOAD(PL21)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(RUN2P3,OLD)
REPLACE(*+WRITMS,RUNL)
DELETE(GETBA)
DELETE($SIO$$$)
FINISH.
LIBRARY(NUCLEUS,OLD)
REPLACE(RUN,RUNL,AL=3,FL=45000,FLO=1)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(RUN1+Q8DIAGP,RUNL)
FINISH.
LIBRARY(SYSIO,OLD)
REWIND(RUNL)
REPLACE(GETBA,RUNL,AL=0)
REPLACE($SIO$$$,RUNL)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAM

Dayfile output of the RUN 2.3 verification program is as follows:

```
            SCOPE 3.4
05.45.28.VRUN04E
05.45.29.VRUN,CM55000,T200.
05.45.29. THIS SIMPLE PROGRAM VERIFIES CORRECT IN
05.45.29.STALLATION OF
05.45.29.RUN(G)
05.45.32.LGO
05.45.38.      4.353 RT SECONDS LOAD TIME
05.45.38.COMPUTATION SUCCESSFUL
05.45.38.STOP
05.45.38.   RUN 2.3
05.45.38.   END OF JOB
05.45.38.CPA       .444 SEC.
05.45.38.PP       4.429 SEC.
```

RELEASE DESCRIPTION

1700 IMPORT HIGH SPEED version 1.0 operates in conjunction with INTERCOM 4.1 and runs under SCOPE 3.4.

The 1700 computer is used as a remote terminal center which accepts jobs for 6000 SCOPE in the same unit record format as the central site and provides identical unit record output within hardware limitations.

Three simultaneous data streams, in addition to operator messages, are provided between the terminal and the central site. The data streams are restricted to one card reader and two printers. A flexible language allows the terminal operator to control his jobs at the central site.

1700 IMPORT HS is written in 1700 Assembly language. It runs under a special operating system constructed from modules of 1700 Mass Storage Operating System (MSOS). No utility functions are provided.

RELEASE MATERIALS

The 1700 IMPORT HS release consists of the following:

IMPORT HS installation deck in binary format

Two decks to establish proper character set orientation

IMPORT HS Verification Program

Source tape of 1700 IMPORT HS (optional)

List tape of 1700 IMPORT HS (optional)

Note: The source and list tapes do not correspond exactly to the binary deck.

HARDWARE CONFIGURATION

The minimum configuration for the 1700 IMPORT HS Terminal is as follows:

1704 4K Computer — A minimum of 8K core is required for 1700 IMPORT HS, its operating system, and buffers.

1708 4K Storage Module

1705 Interrupt Data Channel

1706 Buffered Data Channel

1747 Multiplexer

1711 or 1713 Teletypewriter

1726/405 or 1728/430 Card Reader

1740/501 or 1742/HR300 Printer

Hardware Options

A second printer may be included in the basic configuration.

Communication Link

Each terminal is connected to the central site by a communication link consisting of two DATAPHONE data sets 301B or 303, and one TELPAK A communications line, or their logical and physical equivalent.

Logical Unit Assignments

The following are the logical unit assignments for 1700 IMPORT HS:

Unit Number	Device
2	1728/430 Card Reader
or	
3	1726/405 Card Reader
4	First printer
5	Second printer (if any)

Physical Unit Assignments

The following are the physical unit assignments for 1700 IMPORT HS:

Equipment	Interrupt Line No.	Equipment No. (hexadecimal)
1706 Buffered Data Channel	N/A	(first connection)
1747 Multiplexer	14	8
1711 or 1713 Teletypewriter	1	N/A
1728/430 Card Reader	8 = Data 9 = EOP 10 = Alarm	9
1726/405 Card Reader	11	4
First Printer (1742/HR300 or 1740/501)	5	F
Second Printer (1742/HR300 or 1740/501)	6	E

MODIFICATIONS AND CORRECTIONS

CORRECTIONS

All eligible PSR code published through PSR Summary 320 has been added to the binary deck by addition of hexidecimal patch cards.

Deficiencies

The 1742/HR300 Printer occasionally will lose an interrupt. IMPORT HS detects this and causes the system to signal the operator:

Lnn failed, 00

ACTION

nn is the logical unit number of the printer. The operator should respond with RP to repeat the request.

Limitations

If paper is torn or the supply is depleted, the 1742 rejects all functions including the clear interrupt, causing the same interrupt to be processed repeatedly until the system limits are reached. To recover, reload IMPORT HS.

When paper is running low, the operator should instruct the system to idle the printer by entering ?WAIT,(printer) while the new paper supply is readied. When ready, enter ?GO,(printer) to activate printer.

ADDITIONAL INFORMATION

1. 1726/405 Card Reader Operation

   When a 1726/405 Card Reader is used, IMPORT HS cannot read the last card in the hopper. To ensure that the job is completely read, an extra end-of-file (6/7/8/9 punches in column one) should be added to the end of the deck. IMPORT HS ignores multiple end-of-file cards, so extras will cause no difficulty.

2. Print Complete Messages

   When a job finishes printing, 1700 IMPORT HS will produce the following messages on the teletypewriter:

   JOBNAME LP1 DONE   for the first printer.

   JOBNAME LP2 DONE   for the second printer.

   These messages are the 1700 IMPORT HS variant of the message:

   JOBNAME LP DONE

3. Restarting IMPORT HS

   1700 IMPORT HS should be reloaded from the card reader each time it is executed. The autoload sequence should be checked each time to ascertain that core has not been cleared or used by another program.

   A restart procedure is available if IMPORT HS is terminated by ?END, or if the starting procedure stops before communication is established. To restart IMPORT HS, clear the machine and run. The regular initialization should be executed. If not, reload the deck.

   The above method is not effective if the system was stopped while IMPORT HS processing was active, in such a case, it is best to reload the deck.

4. Job Input Limit Bypass

   The operator may enter ?DVT,,LP to divert all line printer output to the central site. When this statement is entered, the job input limit is ignored until the system is reloaded. Other versions of the DVT command do not cause the job input limit to be bypassed. In these cases, IMPORT HS does not have a true count of jobs in the system.

5. Overflow Light

   When IMPORT HS is entered, the //IMPORT READY message appears, and the operator responds with ?WAIT or one of the options of the ?GO statement, followed by a carriage return. The IMPORT HS system sends an interrupt code to the central site and waits for an answer. During this idle period, before the central site answers the interrupt, a blinking overflow light on the console indicates that IMPORT HS is functioning properly.

6. The IMPORT HS configuration available for checkout included a 501 printer with an ASCII print drum.

GENERAL PROCEDURES

Installation of 1700 IMPORT HIGH SPEED version 1.0 requires the following:

1. Setting installation parameters for the remote terminal configuration, if necessary, and changing any preset parameters as necessary.

2. Loading the IMPORT deck.

3. Modifying core prior to executing IMPORT if necessary.

INSTALLATION PARAMETERS

None

HEXADECIMAL MODIFICATION ROUTINE

If it is necessary to modify core prior to executing IMPORT, hexadecimal change cards must be inserted before the terminating *T card (in Hollerith format) of the released IMPORT deck. The format for these cards follows:

Each modification card may have one or more 9-column change fields starting with an asterisk. The four succeeding characters should contain a hexadecimal address to be changed; the last four should contain the hexadecimal constant to be inserted into this location. Any number of blanks and/or commas may precede or follow each field; and the fields may contain only the characters 0-9 and A-F. If any others are used, results are unpredictable as no error checking is performed by the hexadecimal modification routine. The modification field must be completed by the end of a card, it cannot be continued to the next card. The *T terminator may be placed after all change fields of a modification card.

Examples:

*1234ABCD causes $ABCD to be placed in location 1234. Reading continues.

*0001AAAA, *0002BBBB*0003CCCC*0004DDDD*T causes $AAAA, $BBBB, $CCCC, and $DDDD to be placed into locations 1, 2, 3, and 4, respectively; and IMPORT will be executed.

INSTALLATION OPTIONS

The following changes may be required based on the configuration of the remote terminal. The release deck is constructed for the minimum configuration: one 1728/430 Card Reader and one 1742/HR300 Printer. If another configuration is used, the following hexadecimal corrections should be made to the release deck.

$F9 contains the logical unit number of the card reader. It is set to 2 for the 1728/430. It should be set to 3 to use the 1726/405 Card Reader.

$FB contains the logical unit number of the first printer. It is set to 4 in the release version. This is correct for the first printer, whichever type is used.

$FA contains the logical unit number of the second printer, if available. The released version is set to 0 since it assumes no second printer. If a second printer is to be used, set $FA to 5.

$311 and $240 each contain the constant $2944 in the released version. This assumes 1742/HR300 printers or 1740/501 printers with ASCII type print heads. To print on a 1740/501 printer with a BCD type print head, change contents of $311 to $2844 for the first printer and/or change contents of $240 to $2844 if the second printer has a BCD print head.

The former installation option TERMID is no longer significant. Unique two-character identifiers are now assigned to each port of each HS MUX by the control site.

MJILIM is the maximum number of jobs allowed in the system at any one time. When this number is reached, IMPORT will stop reading jobs from the card reader until a job is printed and leaves the system. The value of MJILIM is arbitrary and can be made significantly larger or smaller to suit the application. The standard value is 25(decimal). To modify this value by assembly, change the EQU value for MJILIM. To modify by using hexadecimal corrections, set cell INIT1 to the instruction ENA- MJILIM where the value may range from 1 to $7F or 127(decimal). For a larger limit, modify cells INIT1 and INIT1+1 to both contain 0B01(hexadecimal). Then set low core cell $DB to contain the number of allowable jobs. This value may range from 1 to 32K. INIT1 is at location $0BE4.

026-029 Keypunch. Core location $1693 in the release contains the command $EA2B which causes IMPORT to convert Hollerith to display, assuming 026 keypunch code. If this location is changed to $EA4B, 029 code will be used. This option must be selected at installation time and cannot be overridden.

PRESET PARAMETERS

The following parameters may be modified as follows:

SYNCCK is set to 10(decimal) in the release version. This setting specifies that status is to be taken ten times upon an interrupt when a transmission occurs before a sync word acknowledge error is recorded. It provides a short time delay for the sync word acknowledge status bit to appear. To change this option by assembly, modify SYNCCK to the desired value. To change it with hexadecimal modifications, place the instruction ENA- SYNCCT in INTT in IMPORT. INTT is at location $0CC5.

TRANCT is set to 120(decimal) in the release version of IMPORT. When a sync word acknowledge error occurs, this count is reduced by one. After 120 unsuccessful retransmissions have been tried, the routine switches to receive mode in an attempt to recover. To change this option, change the EQU value for TRANCT. To change it with hexadecimal corrections, change cell INTTP to contain the instruction ENA- TRANCT. INTTP is at location $0CD2.

RECCYC indicates the number of times status is repeated before recording a cylic code error on input from the data set controller. It provides a short time loop allowing the cyclic code error bit to clear before registering the error. RECCYC is set to 3(decimal) in the release version. To change it by assembly, modify EQU for RECCYC. To change it with hexadecimal corrections, modify INTR to contain the instruction ENA- RECCYC. INTR is at location $0CEF.

Logical Terminal Assignments

A terminal line number is determined by the physical connection.

INSTALLATION PROCEDURES

Additions are required to the released binary deck to select the character set. To select the 63-character set, remove the header card and insert the 63-character set hexadecimal patch cards ahead of the binary deck *T card.

To select the 64-character set, remove the header card and insert the 64-character set hexadecimal patch cards ahead of the binary deck *T card.

Caution: Only one of the above sequences of cards should be added.

LOADING IMPORT DECK

An autoload sequence must be keyed in as follows:

1. Clear the machine, turn on the ENTER switch.

2. Set P to LOC; for this version, use LOC=$1BF0.

3. Set X to the first word of the autoload sequence below.

4. Press the STEP key.

5. Repeat steps 3 and 4 until all the following words are keyed in.

Autoload Word Sequence

Each of the following words must be keyed in sequence according to the above procedure.

```
E80D
C80D
03FE
0DFE
02FE
0FC8
7809
02FE
B807
0107
7806
D8FE
18F7
04A1    (for 430 Reader, 0201 for the 405)
0080
```

6. Load the IMPORT deck into the 430 or 405 Reader and ready the unit. If a 405 is used, a blank card must be placed after the last card in the deck.

7. Put ENTER switch in neutral position.

8. Set P to LOC and run. The deck will be loaded and IMPORT executed.

Punching the IMPORT Deck

If the SKIP switch is on, at the time the IMPORT deck is loaded, a new IMPORT deck will be punched on the 430 before execution of IMPORT. This feature is useful if the number of modification cards is excessive, or if the original deck is in poor condition.

If the original IMPORT deck is loaded from the 1728/430 card reader to punch a new deck, looping will occur after loading is completed, and before punching begins. To punch the new deck, clear the machine and set P to LOC+$DO and run.

Loading the IMPORT Deck

The following errors are detected and signaled by looping with register X set to $18FF at the following locations:

LOC+$56	Deck out of sequence. An 8-bit sequence number is punched into rows 0-5, 11, and 12.
LOC+$61	Incorrect record length. Row 2 of column 2 of each of the two records must contain a punch; this is the sign position of the complemented record field length.
LOC+$70	Checksum error. Deck read or punched incorrectly.

DISK RESIDENT IMPORT

Preparing a Disk Containing the IMPORT System

Load a copy of the MSOS systems initializer and execute it to prepare the new disk. The following statements must be entered from the teletypewriter:

*I,LU	Binary input device definition
*C,LU	Comment device definition
*V	Begin reading input
*T	Final completion after all decks are read

Producing an IMPORT Deck from Disk

If an IMPORT system has been constructed on disk and a deck is to be produced, the following steps are necessary:

1. Turn on the SKIP switch.

2. Autoload the disk; after setting the run switch in run position, the system will loop immediately.

3. Key in the autoload sequence as specified previously in the section Loading IMPORT Deck.

4. Put the first seven cards of the IMPORT deck into the 430 or 405 Reader. If the 405 is used, an extra blank card should be placed at the end of the deck. Then execute the autoload sequence. If the first seven cards are not available, they can be produced by assembling and executing the program BOOTLD. The first seven cards produced by this procedure can be used as specified above. The remainder of the output from the BOOTLD job is of no value and should be discarded.

5. When reading terminates, clear the computer, turn off the SKIP switch, set P to LOC+$D0, and run. A complete deck will be punched and IMPORT reentered. All of core is punched up to the location preceding that into which the autoload sequence is entered.

VERIFICATION PROGRAM

The 1700 IMPORT HS verification program verifies communication with the central site computer. Running time is less than one minute of 6400 computer clock time.

1700/274 INTERACTIVE GRAPHICS SYSTEM (IGS) VERSION 2.0

RELEASE DESCRIPTION

The CONTROL DATA 1700/274 Interactive Graphics System (IGS) Version 2.0 permits real time use of the 6000 Series computers or CDC CYBER 70/Models 72, 73, and 74 by a 274 Graphics Console operator while batch and remote jobs are running under SCOPE 3.4 and INTERCOM 4.1.

The CONTROL DATA 1700 Series Computer is used to control the basic functions of the graphics hardware and allow the software to operate without programmer intervention. Terminals equipped with a remote card reader or card reader/punch and printer may be used to receive output and to submit jobs for execution at the central site.

Programs written in FORTRAN, FORTRAN Extended, COBOL, ALGOL, COMPASS, and BASIC, as well as INTERCOM and IGS jobs may be run under the SCOPE 3.4 Operating System. FORTRAN Extended is used to compile IGS jobs.

IGS can service a maximum of 24 independent graphics consoles simultaneously through four 1700 Series computers.

HARDWARE CONFIGURATION

In addition to the minimum hardware required by 1700 MSOS IMPORT HS and the SCOPE 3.4 Operating System, 1700/274 IGS requires an additional 1706 Data Channel and 1-6 graphics consoles with 1744 Controllers for each 1700 computer.

RELEASE MATERIALS

1700 IGS Version 2.0 source tape (OLDPL) is a labeled binary tape with 3 files.

File 1: UPDATE OLDPL source of the 1700 IGS modules and MSOS 3.0 IGS modification kit

File 2: Binary installation deck

File 3: Verification job deck

LIMITATIONS AND DEFICIENCIES

Limitations

1. Recovery Deadstart does not recover graphics jobs attached to consoles; each job is aborted or rerun.

2. NC=ALL aborts all jobs attached to consoles, but it does not clear the NCON queue.

3. A maximum of 2 IGS jobs may be queued for each console.

4. If an error fatal to the graphics data stream occurs on one 1700, graphics processing is suspended and the data stream must be redefined. All jobs using consoles attached to the 1700 are aborted.

5. Response time is degraded when several consoles use one disk.

6. Buffered magnetic tapes and the IGS/274 console cannot be run on the same 1706 Buffered Data Channel.

Deficiencies

1. Hard copy capabilities are not provided; however routines are available to create hard copies of data on a device chosen by the installation.

1700 INTERACTIVE GRAPHICS INSTALLATION PROCEDURES

The release package, 1700 MSOS 3.0 IGS, is designed for installation in an MSOS 3.0 Operating System in which 1700 MSOS IMPORT HS has been installed.

CUSTOMIZING MSOS 3.0

Modifications to LOCORE and SYSBUF are discussed in separate sections. No changes to MSOS 3.0 other than LOCORE and SYSBUF customization are required for 1700 IGS.

All PSR corrections to MSOS 3.0 and MSOS 3.0 IMPORT HS should be reviewed.

CUSTOMIZING IMPORT HS FOR IGS

MASS MEMORY ORDINAL OF IMPORT

IGS increases the IMPORT mass memory module by adding three routines. As shown in the example, it may be necessary to add IMPORT as mass memory ordinal one or two before the loading of core resident *L DRCORE and *L DRIVERS is processed. This procedure gives the system initializer more room for loader tables while loading IMPORT to avoid overflow.

The sample system memory map demonstrates the *YM control card changes and the deck sequence to load IMPORT at ordinal 2 in the system library. CRIMPT must be included in the *L LOCORE core resident load to avoid unsatisfied externals.

TRANSMISSION BUFFER SIZE

Either transmission buffer size, 165 or 325, can be used for IMPORT when IGS is active. The larger size is more efficient and should be used unless 1700 memory size limitations necessitate the smaller buffers.

DIVERT BY TERMINAL ID

The modified MESIMP provided with the IGS 1700 installation package recognizes a third parameter on the ?DVT command. This parameter allows remote files to be diverted to another remote terminal as well as to central site. To implement this terminal identifier parameter, the following change is required to TRAFIC.

Reassemble module TRAFIC from the 1700 MSCS IMPORT source tape including the following corrective code:

```
*D  M30.4   NEAR TRAFIC.2
      EQU  IGS(1)   ASSEMBLE DVT TID CODE

*C  TRAFIC
```

LOCORE MODIFICATIONS

The standard 1700 MSOS 3.0 LOCORE has been modified for IGS. DECK name IMODKIT on the OLDPL contains the source of the LOCORE modification kit. The LOCORE modification package supplies source card images to define the GRAPHICS communications region in LOCORE. After customization, the cards should replace LOCORE card image 00048 (BZS $B2-$47+1) which defines the length of the free process control area in LOCORE.

Example of 6000 or CDC Cyber 70 job to punch the IMPORT source cards from source OLDPL:

```
JOB,CM40000,TP1.
REQUEST(OLDPL,E) 1700 IGS OLDPL
UPDATE(Q,8)
DISPOSE(COMPILE,PU)
7/8/9
*C IMODKIT
7/8/9
6/7/8/9
```

Most examples in the following sections use specific numerical values for clarity. The actual values must be determined individually for each site.

In LOCORE, an interrupt trap area should be added for each interrupt line connected to a 1744/274. Example for NCON=1:

```
           EXT  DIGI1
LINEn      NUM  0
           RTJ- ($FE)
                         $B is a sample priority level related
           NUM  $B       to the interrupt mask table.
           ADC  DIGI1    DIGI1 is a sample interrupt processor
                         entry in the PHYSTAB for a 1744/274.
```

For more than one 1744/274, additional interrupt trap entries are required. A second interrupt trap might appear:

```
            EXT   DIGI2
LINEm       NUM   0
            RTJ-  ($FE)
            NUM   $B
            ADC   DIGI2      DIGI2 is an entry point in the 1744/274 PHYSTAB
                             for the second console. The interrupt trap source
                             cards are not in IMODKIT.
```

CUSTOMIZING LOCORE IGS MODIFICATIONS

Two changes may be required to customize an installation:

1. 274 consoles on the system should be represented by an entry in the TABLE OF LOGICAL UNITS of 274 CRTS. Each entry is the logical unit of the console and the table should have six entries. The first entry in the table will be console 1; the second console 2, etc. Console numbers should correspond to entries in word 23 of the PHYSTAB tables for the consoles in SYSBUF (SYSBUF modifications). An ADC 0 card should occupy the entry for every NCON not physically present on the system.

2. The mask used to clear graphics interrupts should be set to a 16-bit hexadecimal number such that each bit representing a 1744/274 interrupt line is clear and all others are set.

   Example:

   A 1700 terminal is configured with:

   1744/274 interrupt line 9 logical unit 15

   1744/274 interrupt line 10 logical unit 16

   Required changes to the IGS LOCORE modification package:

   To establish the console numbers:

```
*D  IMODKIT.27,28
     NUM  15          LU NCON1
     NUM  16          LU NCON2
*C  IMODKIT
```

   To establish the INTERRUPT CLEAR MASK:

```
*D  IMODKIT.50
     NUM  $F9FF       So that bits corresponding to
                      interrupt lines 9 and 10 are clear.
*C  IMODKIT
```

SOURCE OF IGS LOCORE MODIFICATIONS

The source is available on the source tape as UPDATE DECK name IMODKIT. A Modification listing is included here for information only.

SYSBUF MODIFICATIONS

An example of the PHYSTAB entry for a 1744/274 follows: Additional SYSBUF modifications include customizing the LOG1, LOG1A, and LOG2 tables. The interrupt mask table should be modified to reflect the addition of the 1744/274 interrupt lines.

```
*                                                                               IMODK139
***** SAMPLE SYSBUF PHYSTAB ENTRIES FOR 1744/274                                IMODK140
*                                                                               IMODK141
* NOTE THAT WORD SEVEN SHOULD BE CHANGED TO REFLECT THE EQUIPMENT NUMBERIMODK142
* AT YOUR SITE                                                                  IMODK143
********** 1744 I74 CONSOLE 1 ******************************************IMODK144
*                                                                               IMODK145
*                                                                               IMODK146
       ENT   DIGI1,DIGI2                                                        IMODK147
       EXT   DIGINI,DIGDRV,DIGERR                                               IMODK148
*                                                                               IMODK149
DIGI1  LDQ   =XPH2741                                                           IMODK150
       JMP*  (PH2741+2)                                                         IMODK151
*                                                                               IMODK152
PH2741 NUM   $12BB     0 SCHEDULER REQUEST/PRIORITY LEVEL                       IMODK153
       ADC   DIGINI    1                                                        IMODK154
       ADC   DIGDRV    2                                                        IMODK155
       ADC   DIGERR    3                                                        IMODK156
       NUM   -1        4                                                        IMODK157
       NUM   0         5                                                        IMODK158
       NUM   0         6                                                        IMODK159
       NUM   $3E00     7 EQUIP C BDC 2 - Q REGISTER                             IMODK160
       NUM   $A6       8                                                        IMODK161
       NUM   0         9                                                        IMODK162
       NUM   0         10                                                       IMODK163
       NUM   0         11                                                       IMODK164
       NUM   0         12                                                       IMODK165
       ADC   0         13 MASS MEM LENGTH 0, DRIVER CORE RESIDENT               IMODK166
       ADC   0         14 MASS MEM SECTOR NAME                                  IMODK167
       ADC   0         15 FNR RETURN INFORMATION                                IMODK168
       ADC   0         16                                                       IMODK169
*   THIS ADC POINTS TO THE NEXT AVAILABLE CONSOLE.                              IMODK170
*   IF THERE IS NO OTHER CONSOLE PRESENT IT POINTS TO ITSELF.                   IMODK171
       ADC† PH2742     17                                                       IMODK172
       ADC   0         18                                                       IMODK173
       ADC   0         19                                                       IMODK174
       NUM   0         20                                                       IMODK175
       NUM   0         21                                                       IMODK176
       NUM   0         22                                                       IMODK177
       NUM   0         23 CONSOLE NUMBER MINUS 1                                IMODK178
       EJT                                                                      IMODK179
```

†If multiple 1744/274 units are on the system, this address constant points to the PHYSTAB of the next console. If only one is present, it points to itself. In this example, if PH2741 is the only 274 PHYSTAB, the instruction will be ADC PH2741.

Each 1744/274 in the configuration should be represented by an interrupt processor (DIGI1 above) and a PHYSTAB (PH2741 above).

For the second 1744/274 PHYSTAB, the last word will be NUM 1, as it always contains the quantity (NCON-1). For the third PHYSTAB, the last word should be replaced by NUM 2, etc.

Sample PHYSTAB entries are in DECK name IMODKIT in the 1700 IGS OLDPL.

Step by Step Procedures for SYSBUF modification:

1. Customize the LOG1A table by placing

   ```
   ADC (PHYSTAB address)
   ```

   in the entry corresponding to the logical unit selected for the 1744/274 when the LOCORE modifications were customized. The entry should follow the EQU Lnn(*) card in LOG1A for the interrupt line of the 1744/274.

   For the example of PHYSTAB used in this section, the LOG1A entry would be

   ```
   ADC   PH2741
   ```

2. Add to LOG1 table

   ```
   ADC   0
   ```

   at the ordinal corresponding to the logical unit of the 1744/274.

3. Add to LOG 2 table

   ```
   NUM   $FFFF
   ```

   at the ordinal corresponding to the logical unit selected for the 1744/274.

4. The interrupt mask table, MASKT, should be customized so that the interrupt bit corresponding to the interrupt line of the 2744/274 is set in every priority up to, but not including the priority for the 1744/274 in the interrupt trap region in SYSBUF. For example, if the 1744/274 is running at priority $B, as in the examples, the MASKT entries for priorities zero through ten ($A) must be changed. If the 1744/274 is on interrupt line 8, this change requires adding $0100 (setting bit 8) to all the MASKT entries which must be changed.

5. The 1744/274 entries should not be added to DGNTAB, the diagnostic timer table.

6. The PHYSTAB, physical device table, should be added to SYSBUF. If the priority of the 1744/274 is other than $B, word zero of the PHYSTAB should be changed from

   ```
   TAG  NUM  $12BB      to      TAG  NUM  $12xx
   ```

   where x is the priority level chosen for the 1744/274. Word seven of the PHYSTAB should be customized to reflect the data channel and equipment code of the 1744/274.

   The priority also appears in the interrupt trap region. That priority and the one in PHYSTAB must be the same.

CUSTOMIZING 1744 CONTROLLER SIZE

The 1700 IGS system, as released, is configured for the following controller sizes:

Console one	8K
Console two	8K
Console three	4K
Console four	4K
Console five	4K
Console six	4K

They may be changed by modifying the IGSPRO module before building the system or by entering a hexadecimal patch at initialization.

MODIFYING IGSPRO

The MAXBUF table in IGSPRO contains six entries, each corresponding to a possible 1744/274 in the configuration. Each entry in the table represents the last word address of the 1744 controller corresponding to the console. For a 4K controller, this entry is $FFF; for an 8K controller, $1FFF.

In the release source and binary, the table appears:

```
MAXBUF  NUM  $1FFF,$1FFF,$FFF,$FFF,$FFF.$FFF       IGSPRO 1377
```

The following UPDATE correction will modify IGSPRO to indicate that console 1 and 2 are connected with 4K 1744 controllers.

```
*IDENT,FOURK
*DELETE,IGSPRO.1377
MAXBUF  NUM  $FFF,$FFF,$FFF,$FFF,$FFF,$FFF
*C IGSPRO
```

MAXBUF must have six entries. Values in the table that correspond to consoles not present in the configuration will not be used.

HEXADECIMAL PATCH

The following procedure, using the CDEBUG module, allows initialization patching of the table indicating the 1744 controller size. All type-in's are followed by a carriage return.

1. Initialize IMPORT, and establish communications with INTERCOM.

2. Define a GRAPHICS data stream using the standard procedure; <u>do</u> <u>not</u> type ?GO to initialize data transfer.

3. Press the manual interrupt switch and enter DB to load the DEBUG package. DEBUG responds DEBUG IN.

4. Dump the contents of LOCORE cell $5D by typing DPC,5D
   DEBUG responds 005D hhhh
                  NEXT?

   The value dumped, hhhh, is the address of a table in IGSPRO.

5. Add 5 plus console number (one to six) to the hhhh hexadecimal value just dumped. The result is the address of the MAXBUF entry for the console to be changed. aaaa = hhhh + 5 + console number.

6. Store the last word address of the 1744 controller into aaaa (the result from step 5) by typing

   LHX,aaaa,size

   size is FFF for a four K controller and 1FFF for an 8K controller.

   DEBUG responds NEXT?

7. Enter OFF to exit the DEBUG package.

In the following example, the MAXBUF table is patched to indicate that console two is present on the configuration with a 4K 1744 controller.

```
Operator Action

Press manual interrupt
                            MI
type-in                     ?IM,,325
                            //1700 MSOS IMPORT
                            //CENTRAL READY
Press manual interrupt      //YOUR TERMINAL ID IS AC
                            MI
type-in                     ?DEF,GR,15
                            //01.00.00 OK
                            //01.00.00 OK
Press manual interrupt
                            MI
type-in                     DB
                            DEBUG IN
type-in                     DPC,5D
                            005D  3600
                            NEXT?
type-in                     LHX,3607,FFF
                            NEXT?
type-in                     OFF
                            DEBUG OUT
Press manual interrupt
                            MI
type-in                     ?GO
                            .
                            .
                            .
```

The patch is stored in address 3607 which is the contents of address 5D plus 5 plus console number, 2 for the example. All numbers and calculations used with DEBUG in the example are hexadecimal.

INTERACTIVE GRAPHICS 1700 BINARY INSTALLATICN PACKAGE

The binary installation and verification decks are on files 2 and 3 of the 1700 MSOS 3.0 IGS tape. Example program to obtain the decks on cards:

```
PUN,CM20000,TP1.
REQUEST(O,E) 1700 IGS TAPE
SKIPF(O,1,15)
COPYBF(O,P80C)
COPYBF(O,PUNCH)
6/7/8/9
```

The binary installation package contains binary object decks and binary system initialization control cards. For installation, it should be separated into five sections.

Separating the binary object decks by card count may lead to errors. The Mass Storage Operating System Version 3.0 Reference Manual contains a discussion of binary card format and relocatable binary format which may be referenced when separating the binary installation package into the proper set of object decks.

1. System Initialization Control Cards

   The first 14 cards in the deck are *YM control cards for establishing the system library ordinals of 14 GRAPHICS overlays. It is assumed that the MSOS 3.0 operating system with MSOS IMPORT HS installed has 17 system library ordinals established. The control cards are:

```
*YM,GIOV1,18
*YM,GIOV2,19
*YM,GIOV3,20
*YM,GIOV4,21
*YM,GIOV5,22
*YM,GIOV6,23
*YM,GIOV7,24
*YM,GIOV8,25
*YM,GIOV9,26
*YM,GIOV10,27
*YM,GIOV11,28
*YM,GIOV12,29
*YM,QUEUE,30
*YM,GIOV13,31
```

   These control cards should be placed in the system initialization deck following the *YM control card which defines system library ordinal 17. If the number of system library ordinals is not 17 before IGS is installed, the *YM cards should be changed so that the IGS overlays will be the next 14 system library modules.

   The next 14 cards in the package are control cards of the form *S,ordinal,priority,M. They should be placed at the end of the system initialization deck with the other *S control cards, which establish the core priorities of existing system library modules (after the *T control card). If the ordinals of the IGS overlays have been changed on the *YM control cards, the ordinal field of each *S card must be changed accordingly.

2. DIGDRV (2744/274 driver) Object Deck

   The next 30 cards in the package are a binary object deck of DIGDRV, the 1744/274 driver. This deck should be placed in the system initialization deck so that it will be loaded in core resident. In the example on the following pages, DIGDRV is placed in the *L DRIVERS core resident load. As always, DIGDRV must precede SPACE, as the SPACE program must be the last core resident program loaded.

3. MESIMP Object Deck (modified for IGS)

   The next 71 cards in the installation package are a binary object deck of the MESIMP module (Message Processor for IMPORT) as modified for IGS. This object deck should replace the binary of MESIMP installed with MSOS IMPORT HS. It must be part of the *M IMPORT load.

4. IGSPRO, IDSGR, GRAPH, Modules to be Added to IMPORT

   The next 88 cards include three object decks, IGSPRO, IDSGR, and GRAPH, to be added to the IMPORT module. They may be added as the last three decks following the *M IMPORT control card and preceding the next system initialization control card. Routine IMPORT must be the first routine loaded in that module.

5. IGS Overlays and Control Cards

   The remainder of the deck (557 cards) includes an *M control card and the binary object decks for each of 14 IGS overlays. These should be placed in the system initialization deck so that the order of the *M control card matches the ordinals on the *YM control cards previously inserted in the deck.

   In the example, the overlays are the last 14 mass memory resident system library routines in the installation deck.

   The order of the overlays in the IGS installation package matches the order of the routines on the *YM control cards included in the package.

Summary of Deck Structure 1700 IGS Installation Package

Binary *YM control cards	14 cards
Binary *S control cards	14 cards
Binary object deck DIGDRV	30 cards
Binary object deck MESIMP	71 cards
Binary object decks to be added to IMPORT module (IGSPRO, IDSGR, GRAPH)	88 cards
IGS overlays, control cards and object decks	557 cards

The *M card to load GIOV1 specifies sector number 140 (base 16) to ensure that all primary IGS overlays are resident on the same disk cylinder, reducing disk repositioning time during graphics activity. If the system library loaded before GIOV1 occupies this sector, the *M card must be changed.

1700 SOURCE

The source code of the IGS Version 2.0 routines which run in the 1700 system is provided on a magnetic tape PL17, together with MESIMP (a portion of the 1700 Installation Package). The tape is in 6000 UPDATE OLDPL format compatible with UPDATE 1.2 under the SCOPE 3.4 Operating System.

The source of the IGS modules and MESIMP is followed by *CWEOR,0. Following the conditional end of record is the DECK IMODKIT containing 1700 IGS LOCORE and SYSBUF modifications.

The first file on the tape is the OLDPL source. The sample job below demonstrates how the tape may be used to punch a source deck with sequence numbers from the COMPILE file of an UPDATE run. It also is possible to obtain card images with an UPDATE control card including the parameter C=PUNCH.

Program PUN copies the first three characters of the UPDATE correction identifier to columns 73-75 of the card and places the UPDATE sequence number in columns 76-80.

Sample Run to Create Card Source with Sequence Numbers

```
    UP1700,P4,T2000,CM45000,TP1.
    REQUEST, OLDPL,E.
    UPDATE(Q)
    RUN(G)
    DISPOSE(TAPE2,PU)
    7/8/9
    *COMPILE,DIGDRV          include any corrections to the module
    7/8/9
          PROGRAM PUN(PUNCH,COMPILE,TAPE1=COMPILE,TAPE2)
          DIMENSION IC(80)
        1 READ(1,100) IC,IN
          IF(EOF,1) 3,2
        2 WRITE(2,101) (IC(I),I=1,72),(IC(I),I=74,76),IN
          GO TO 1
        3 STOP
      100 FORMAT(80R1,I6)
      101 FORMAT(75R1,I5)
          END
      7/8/9
      6/7/8/9
```

If magnetic tape equipment is available on the 1700, PROGRAM PUN may be used to write a tape suitable for input to the 1700 Macro Assembler. The output tape should be an S tape at density suitable for the 1700 magnetic tape equipment. Example REQUEST card, placed before the RUN(G) control card:

```
    REQUEST(TAPE2,MT,S,HI)
```

If TAPE2 is defined as magnetic tape, the DISPOSE control card must be removed.

Procedures for the use of CLASS, a CDC CYBER 70 Version of the 1700 Macro Assembler, are described at the end of section II-22. Either CLASS or the 1700 Macro Assembler Version 2.0 may be used to maintain 1700 IGS.

Organization of routines in the IGS 1700 overlays and the corresponding source by UPDATE DECK name, as it appears on the release source tape:

Overlay	Routines Required	Corresponding Source UPDATE DECK
GIOV1	*GIOV1	GIOV1
	RW	+RW
GIOV2	*GIOV2	GIOV2
	RW	+RW
	RWAAT	+RWAAT
GIOV3	*GIOV3	GIOV3
	GICLR	GICLRA
GIOV4	GIOV4	GIOV4
GIOV5	*GIOV5	GIOV5
	RW	+RW
	RWAAT	+RWAAT
GIOV6	*GIOV6	GIOV6
	RWAAT	+RWAAT
GIOV7	*GIOV7	GIOV7
	GIDISP	GIDISPA
	RW	+RW
	RWAAT	+RWAAT
GIOV8	*GIOV8	GIOV8
	GIMASK	GIMASK
	GICLR	GICLR
	GIABRT	GIABRT
	GUAN	GUAN
	PUT	PUT
	RW	+RW
	RWAAT	+RWAAT
GIOV9	*GIOV9	GIOV9
	QUEUE2	QUEUE2
	QUEUE3	QUEUE3
	GIANS	GIANS
	RW	+RW
GIOV10	*GIOV10	GIOV10
	GITCON	GITCON
	GITMMV	GITMMV
	GITIMV	GITIMV
	GIBRWT	GIBRWT
	RW	+RW
	RWAAT	+RWAAT
GIMAC	*GIMAC	GIMAC
	RW	+RW
GIOV12	*GIMACE	GIMACE
	RW	+RW
QUEUE	*QUEUE	QUEUE
	TRACD	TRACD
	GIANC	GIANC
GIOV13	*GIOV13	
	GIABRT	
	GURSET	GIOV13
	GUAN	
	PUT	
	RW	

*The object deck for this routine must be the first binary in the overlay.
+Although multiple copies of the object deck for these routines occur in the system, they correspond to one copy of source on the source tape.

SUMMARY OF REMOTE ACTIVATION SEQUENCE

The operator loads MSOS IMPORT HS from the system disk to start operation. Jobs are input through the remote terminal's card reader. MSOS IMPORT HS is loaded at the terminal according to the following sequence.

1. Set all console switches to neutral.

2. Verify that:

   Previously prepared disk pack on the disk pack on the disk drive contains the operating system, MSOS IMPORT HS, and MSCS IMPORT HS IGS

   Disk and controller are on and ready

   DSC is on, all test switches are off, and the data set is plugged in

   Card reader, printers, and teletypewriter power is on

   1713 teletypewriter right-hand selector switch is set in the ON LINE position and that it is in K mode.

3. Press the CLEAR switch on the computer console.

4. Momemtarily press the AUTO LOAD button on the 1738 Disk Pack Controller.

5. Momentarily set the RUN-STEP switch to RUN. At the teletypewriter, the typeout PP appears.

6. Set the PROTECT switch to the PROTECT position.

7. Press the BREAK RELEASE button and type an asterisk, followed by a carriage return.

8. Press MANUAL INTERRUPT key each time a command is entered.

9. MI is typed out each time in response.

10. Clear breaklight.

11. Enter command:

    ?IM,,325 to initialize MSOS IMPORT HS.

12. Press carriage return key.

13. The system responds as follows;

//1700 MSOS IMPORT (MSOS IMPORT HS has been loaded)

//NO COMM (Printed out every 30 seconds until central site responds)

//CENTRAL READY (Communication has been established with INTERCOM)

//YOUR TERMINAL ID IS AC

14. To drop MSOS IMPORT HS at any point, enter the following command: (All commands must be followed by a carriage return and preceded by a manual interrupt).

?END

This command may be entered also after the NO COMM message is received.

15. Otherwise, press the MANUAL INTERRUPT key.

16. MI is typed out.

17. Typically, the next step is to assign a data stream as follows:

?DEF,GR,lu (lu is the logical unit number of the data stream assigned.)

18. Command is acknowledged:

//hh.mm.ss OK

//hh.mm.ss OK

19. Additional data streams, such as one for the card reader, may be defined and acknowledged. If core is not available, the following message appears:

//NO CORE

If desired, another stream can be released to make core available.

20. Enter command to activate data streams:

?GO or ?GO,lu

21. Command is acknowledged:

//OK

22. The system is ready to receive and process graphics and batch data. The teletypewriter output produced should appear as follows:

```
PP
*
MI
?IM,,325
//1700,MSOS IMPORT
//CENTRAL READY
//YOUR TERMINAL ID IS AC
MI
?DEF,GR,14
///0.20.15.OK.
///0.20.15.OK.
MI
?GO,14
//OK
```

IGS VERSION 2.0 VALIDATION DECK

The deck may be input from central or remote site.

IMPORT must be active and communicating with INTERCOM, and the graphics data streams must be defined.

When the job attaches the console, it displays:

```
                   BUTTON

    ONE                          FOUR

    TWO                          FIVE

    THREE                        SIX
```

BUTTON is a light button. The others are single picks.

Pick one of the six single pick alphanumeric; it will be marked (blink). Pick BUTTON; it also will be marked. When the button request from the 6000 has been completed, both display items will be unmarked and the cycle may begin again. The validation deck prints out the data returned.

The following output should result from lightpen selection of: one, button, two, button, three, button, four, button, five, button, six, button.

```
BUTTON SELECTED

STRING PICK 1 SELECTED


BUTTON SELECTED

STRING PICK 2 SELECTED


BUTTON SELECTED

STRING PICK 3 SELECTED


BUTTON SELECTED

STRING PICK 4 SELECTED


BUTTON SELECTED

STRING PICK 5 SELECTED


BUTTON SELECTED

STRING PICK 6 SELECTED
```

RELEASE DESCRIPTION

8-Bit Subroutines Version 1.0 run under 6000 SCOPE 3.4 and 6000 Record Manager 1.0.

RELEASE MATERIALS

The 8-Bit package is released on program library tape PL3 along with 6RM (6000 Record Manager), IS (SCOPE Indexed Sequential) and DA (SCOPE Direct Access).

A complete catalog of PL3 contents may be found in section I-3.

HARDWARE CONFIGURATION

The 8-Bit package requires the same minimum hardware configuration as SCOPE 3.4. An extended print train is required to print ASCII 96-character graphic files, if used.

GENERAL DESCRIPTION

The relocatable routines from the 8-Bit package 1.0 run under SCOPE 3.4, 6RM 1.0 with COBOL 4.0 or FORTRAN Extended 4.0 or COMPASS 3.0. COPY8P, a stand-alone routine used to print 360/370 print files, is control card callable and runs under SCOPE 3.4.

INSTALLATION PROCEDURES

PL3 contains 26 files, files 10-13 and 23-26 apply to the 8-Bit package: A complete PL description appears in section 3 of this document.

Files 23-26 may be obtained as follows:

```
Job Card

REQUEST(PL3,E)            mount tape PL3
SKIPF(PL3,22,17)          skip 22 files
COPYBF(PL3,PUNCH,2)       2 installation decks
COPYBF(PL3,PUNCH,2)       2 verification decks
UNLOAD(PL3)
6/7/8/9
```

The installation decks are listed below:

Deck BIT81 is a maintenance deck that allows updates of the 8-Bit package routines on the PL3 tape. This deck updates the 8-Bit program library, assembles the 8-Bit relocatable object routines, assembles COPY8P and creates a new COPY8P absolute overlay. The job essentially allows creation of a revised PL3 release tape.

Deck BIT82 adds the 8-Bit package to the running system. Relocatable object routines are put in the SYSIO library. COPY8P becomes part of the NUCLEUS library. Deck SCOPE3, described in the SCOPE section of this document, then can be run to create a deadstart tape of the running system.

The T600 parameter on the BIT81 job card has been found to be insufficient on a 6200 or 6400. Please increase the value to T1400.

```
BIT81,CM65000,T600,MT2.
COMMENT. THIS JOB UPDATES AND CREATES BINARIES OF 8 BIT SUBROUTINES
COMMENT. AND COPIES THE OLDPL AND BINARY OF 6RM, IS AND DA TO THE NEW
COMMENT. PL3C. THE NEW PL3C WILL CONSIST OF 13 FILES AND 8-BITS BEGINS
COMMENT. WITH FILE 10.
COMMENT. FILE 10 WILL BE THE 8-BIT NEWPL WITH A *WEOR BETWEEN
COMMENT.       OBJECT-TIME ROUTINES AND CONTROL CARD CALLABLE COPY8P.
COMMENT. FILE 11 WILL BE RELOCATABLE BINARIES OF OBJECT-TIME ROUTINES.
COMMENT. FILE 12 WILL BE RELOCATABLE BINARY OF COPY8P.
COMMENT. FILE 13 WILL BE CONTROL CARD CALLABLE COPY8P IN OVERLAY.
LABEL(PL3CIN,R,L=RM*3P4,D=HI)
REQUEST,PL3C,N,HI.     SCTATCH FOR NEW PL3C
LABEL(PL3C,W,L=RM*3P4,D=HI)
REWIND(PL3CIN,PL3C)
COPYBF(PL3CIN,PL3C,9)           POSITION TO 8-BIT OLDPL
UPDATE(F,P=PL3CIN,N=PL3C,R,X)
COPYBF(EOF,PL3C)
UNLOAD(PL3CIN)
REWIND(COMFILE)
COMPASS(I,S=IOTEXT,S=IPTEXT,L=0,B=PL3C)       OBJECT ROUTINES
COPYBF(EOF,PL3C)
COMPASS(I,S=IOTEXT,S=IPTEXT,L=0,B=CPY8)
REWIND(CPY8)
COPYBF(CPY8,PL3C)
LOAD(CPY8)
NOGO.
REWIND(COPY8P)
COPYBF(COPY8P,PL3C)
UNLOAD(PL3C)
7/8/9  END OF RECORD
*/ ADD CORRECTIONS HERE
*C PARSE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
BIT82,CM45000,T50,MT1.
COMMENT.  THIS JOB EDITLIBS THE 8-BIT BINARIES FROM
COMMENT.  THE RELEASED 8-BIT PL OR THE PL CREATED BY DECK BIT81.
LABEL(PL3C,R,L=RM*3P4,D=HI)     MOUNT PL3C
SKIPF(PL3C,10,17)     SKIP 6RM, IS, DA AND 8-BIT PL
COPYBF(PL3C,BIT8)       RELOCATABLES
SKIPF(PL3C,1,17)       SKIP COPY8P RELOCATABLES
COPYBF(PL3C,CPY8)       COPY8P OVERLAY
REWIND(BIT8,CPY8)
EDITLIB(SYSTEM)
UNLOAD(PL3C)
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(*,CPY8,AL=1,FL=13300,FLO=1)
FINISH.
LIBRARY(SYSIO,OLD)
REPLACE(*,BIT8)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

VERIFICATION PROGRAMS

Dayfile output as produced by running the two verification program jobs is listed below.

```
              SCOPE 3.4
21.15.07.VAL8BIT
21.15.08.VAL8BIT,CM65000,T100.
21.15.08.COBOL(LRM)
21.16.08.      1.875 RT SECONDS LOAD TIME
21.16.12.COMPILING NESTED
21.16.23. 000 E AND     T/U  DIAGNOSTICS ISSUED
21.16.23. FIELD LENGTH NEEDED FOR COBOL 053100
21.16.25.      1.093 CP SECONDS COMPILATION TIME
21.16.25.END COBOL
21.16.25.FILE(TAPE01,BT=K,RT=U,RB=1,MBL=270,MRL=2
21.16.25.70,CM=NO)
21.16.26.REDUCE.
21.16.26.LDSET(FILES=TAPE01)
21.16.27.LGO.
21.16.53.      NON-FATAL LOADER ERRORS - SEE MAP
21.16.57.     30.852 RT SECONDS LOAD TIME
21.17.05.  8 BITS ROUTINES VALIDATE
21.17.06.CPA       5.597 SEC.
21.17.06.PP       23.840 SEC.
```

```
              SCOPE 3.4
21.15.04.VLCPY8P
21.15.04.VLCPY8P,CM15000,T40.
21.15.04.COPY8P(IN1,IN2,CODE=A)
21.15.05. COPY8P VALIDATES
21.15.05.CPA        .004 SEC.
21.15.05.PP         .639 SEC.
```

RELEASE DESCRIPTION

MARS VI, Multi-Access Retrieval System for 6000 Series computers, is a data management system which allows a user to organize data in the computer, store it, and retrieve and/or modify all or any specified part of the data. The language follows simple English syntax, facilitating use of the system by non-programmers. MARS VI provides extremely rapid retrievals; it operates from remote terminals as well as from the central site. It allows batch mode processing under SCOPE 3.4 as well as Teletype or 200 User Terminal interactive processing under INTERCOM, Version 4.1.

Version 2.1 includes relocatable subroutines that can be called from a COBOL program to access the data base. MARS VI is a central memory application program operating under the SCOPE 3.4 Operating Sysstem in conjunction with RUN FORTRAN Version 2.3, COMPASS 2.0 assembler and FORTRAN Extended Version 3.0. The MARS VI system has no peripheral processor routines.

HARDWARE REQUIREMENTS

Central memory requirement is 49,152 decimal words; otherwise the minimum configuration for MARS VI is the same as for SCOPE 3.4.

RELEASE MATERIALS

MARS VI is released on an Update program library.

LIMITATIONS

1. Minimum octal field length requirements for the MARS VI modules:

LOADER	62K
RETRIEVAL	55K
REPORTER	55K
UPDATE	65K (non-key updates)

2. When absolute overlays are being prepared, at least 105000 (octal) field length is required.

3. When index tables of a partially inverted file are on permanent files, the user must make certain he has EXTEND and MODIFY permissions before using the UPDATE module, or he may destroy his data base.

4. The validation deck will not execute properly unless COBOL interface has been installed.

5. No embedded blanks are allowed in arithmetic expressions.

INSTALLATION PROCEDURES

The release tape for MARS contains eight files. File 1 contains the program libraries; file 2 contains binary of MARS in absolute overlay format; file 3 contains the COBOL interface in a form which may be cataloged as a permanent file; file 4 contains the same information as file 2 but in relocatable format. Files 5-8, which contain the necessary installation and verification decks, may be obtained by running a job of the type:

```
job card
REQUEST,PL23,E.
SKIPF(PL23,4,17)
COPYBF(PL23,PUNCH,4)
UNLOAD,PL23.
6/7/8/9
```

Job MARS1 performs program library maintenance regenerating files 1 through 4 of the release tape. Job MARS2 use EDITLIB to enter file 2 of the release tape, or a tape created by deck MARS1, into the running system. Job MARS3 catalogs the COBOL interface as a permanent file.

NOTE: The T7000 parameter on the MARS1 job card must be changed to T20000 if the deck is to be run on a 6200.

```
MARS1,CM105000,T7000,MT02.
RFL(65000)
COMMENT.  THIS JOB UPDATES AND CREATES THE BINARY OF MARS
COMMENT.  THE NEW PL23 WILL CONSIST OF FOUR FILES
COMMENT.  THE FIRST FILE WILL BE THE MARS NEWPL
COMMENT.  THE SECOND FILE WILL BE THE ABSOLUTE OVERLAYS
COMMENT.  THE THIRD FILE WILL BE MARS COBOL INTERFACE
COMMENT.  THE FOURTH FILE WILL BE THE RELOCATABLE BINARY
LABEL(PL23IN,R,L=MARS2P1*3P4,D=HI) MARS OLDPL
REQUEST(PL23,N,HI)   SCRATCH FOR NEW PL23
LABEL(PL23,W,L=MARS2P1*3P4,D=HI)
REWIND(PL23IN,PL23)
UPDATE(F,P=PL23IN,N=PL23)
UNLOAD(PL23IN)
RUN(S,,,COMPILE,XX)
RUN(S,,,COMPILE,XX,MARSCOB)
FTN(I=COMPILE,B=MARSCOB,L=XX,OPT=2)
REWIND(MARSCOB)
RFL(105000)
LOAD(LGO)
NOGO.
RFL(25000)
REWIND(MARS,PL23,LGO)
SKIPF(PL23,1,17)
COPYBF(MARS,PL23)
COPYBF(MARSCOB,PL23)
COPYBF(LGO,PL23)
UNLOAD(PL23)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
*/   ADD CORRECTIONS HERE
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
MARS2,CM55000,T7000,MT01.
COMMENT.  THIS JOB EDITLIBS THE MARS V2.1 BINARIES FROM THE
COMMENT.  MARS PL OR THE PL CREATED BY DECK MARS1.
LABEL(PL23,R,L=MARS2P1*3P4,D=HI)    MOUNT MARS 2.1 PL
REWIND(PL23)
SKIPF(PL23,1,17)    SKIP OLDPL
COPYER(PL23,MAR)    MARS MAIN OVERLAY
COPYBF(PL23,MARSOVL)    SECONDARY OVERLAYS
REWIND(PL23,MAR,MARSOVL)
UNLOAD(PL23)
EDITLIB(SYSTEM)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
READY(SYSTEM)
LIBRARY(NUCLEUS,OLD)
REPLACE(MARS,MAR,AL=1,FL=65000,FLC=1)
FINISH.
LIBRARY(SYSOVL,OLD)
REPLACE(*,MARSOVL)
FINISH.
COMPLETE.
ENDRUN.
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

```
MARS3,CM35000,T100,MT01.
COMMENT.  THIS JOB CATALOGS THE COBOL INTERFACE
COMMENT.  AS A PERMANENT FILE CALLED MARSCOB.
LABEL(PL23,R,L=MARS2P1*3P4,D=HI)     MARS PL
REQUEST(MARSCOB,*PF)
REWIND(PL23)
SKIPF(PL23,2,17)    SKIP OLDPL AND OVERLAYS
COPYBF(PL23,MARSCOB)
REWIND(PL23,MARSCOB)
CATALOG(MARSCOB,MARSCOB,ID=MARS,MR=1)
UNLOAD(PL23)
COMMENT.  *** END OF JOB ***
7/8/9  END OF RECORD
6/7/8/9  END OF FILE
```

The header information should be changed when PSR's are installed to reflect the current PSR or PSR summary.

The following cards can be used for this purpose:

```
*I LZ.1       HEADR (1MARS VI V2.1 PSR 1)
```

USER LIBRARIES

The following paragraphs apply only when MARS is executed from a user library.

To install MARS as a user library, it is necessary to assemble the name of the library into MARS.

```
    *DELETE USERLIB.1
    LIBNAME   VFD      60/6LMARSVI
```

The default value for LIBNAME is SYSOVL. The EDITLIB directives to create a user library are:

```
REQUEST(MARSVI,*PF)
EDITLIB(USER)
CATALOG(MARSVI,MARSVI,ID=MARS)
7/8/9
LIBRARY(MARSVI,NEW)
REPLACE(*,MARS,AL=1,FL=65000,FLO=1)
FINISH.
ENDRUN.
6/7/8/9
```

The above example assumes the overlays to be on a file called MARS.

If LIBNAME is not changed while MARS is assembled, the MARS command

```
USERLIB IS MARSVI;
```

can be issued while in MARS. This command has the same effect of assembling in the library name.

EXECUTION FROM USER LIBRARIES

When batch, is running the following control cards are required

```
ATTACH(MARSVI, MARSVI,ID=MARS,MR=1)
LIBRARY(MARSVI)
MARS.
```

For execution under INTERCOM, the following commands should be entered.

```
ATTACH(MARSVI, MARSVI,ID=MARS,MR=1)
XEQ,LIBLOAD=MARSVI,MARS,EXECUTE
```

VERIFICATION PROGRAM

Dayfile output from running the verification program should be similar to the following. The COBOL Interface must be available as a permanent file for the verification program to run.

```
MARS VI V2.1
$                                                                   ;
$        THE FIELD LENGTH REQUIRED FOR THE MODULES ARE              ;
$                                                                   ;
$                      LOADER    63000  OCTAL               ;
$                 RETRIEVAL    56000  OCTAL          ;
$                 UPDATE      65000  OCTAL        ;
$                                                                   ;
DEFINE:
NEW DATA BASE AIRFIELD;
 1) ID NUMBER(INTEGER NUMBER):
MAP;
DEFINE SUBITEMS;
RECORD ID IS AFIELD;
COL 1-10, C1, IDENTIFICATION NUMBER.NUM,KEY;
COL 11-37,NAME,AIRFIRLD NAME.ALPHA;
COL 38,CODE,ENTRY CODE;
COL 39-53,COORDS,GEOGRAPHIC COORDINATES;
COL 39-40,DEGLAT,DEGREES LATATITIDE.NUM;
COL 45,NSDIRECT,NORTH OR SOUTH DIRECTION.ALPHA;
COL 53,EWDIRCT,EAST OR WEST DIRECTION.ALPHA;
COL 57,OPSTATUS,OPERATIONAL STATUS CODE;
COL 59-63, ELEV,ELEVATION.INTEGER;
COL 68,MAINCODE,MAINTENANCE FACILITIES CODE;
COL 72,LIGHT,LIGHTING CODE:
COL 86,RUNCOND,RUNWAY CONDITION;
COL 96-115/2,RG1:
COL 1-3, RG2.ALPHA;
COL 4-8, RG3.ALPHA:
COL 9-10,RG4.ALPHA;
END RG1:
COL 99-103,RNWYELEV,RUNWAY ENDDELEVATION.NUM;
COL 106-108,RENDID,HIGH RUNWAY IDENTIFICATION.NUM;
COL 114-115,ROVERUN,RUNWAY OVERRUN2;
COL 118-120,RUNOBS,RUNWAY OBSTRUCTION;
COL 126-129,MAGVAR,MAGNETIC VARIATION;
COL 130,RECMARK,RECORD MARK:
END DEFINITION;
LOADER;
DATA FILE IS TEST,CR;
RECORD FILE IS TESTA;
SCAN AFIELD;
NUMBER OF RECORDS EXAMINED WAS          20
NUMBER OF RECORDS ACCEPTED WAS          20
NUMBER OF RECORDS STARRED WAS           0
NUMBER OF RECORDS ON RECORD FILE IS          20

AIRFIELD              DEFINITION VERSION     1   DATA VERSION     1
UNLOAD TEST:
$        ;
DEFINE MACRO MYMACRO;
EXPAND ON;
$                                              ;
$          THIS IS A MACRO EXPANDING           ;
$                                              ;
ATTACH:  AFIELD; ID=CDC;CY=40;  FINIS; GETPF  TESTA,TESTAPFN;
RETRIEVAL:
EXPAND OFF;
END MACRO;
DEFINE MACRO REPORT1;
```

```
MESSAGE FILE IS NULL;
RETRIEVAL;
IF   CCL(38) EQ ≠P(01) AND (ELEV-3)/10 EQ ≠P(02) QUALIFY C1,KK=≠P(03),
CODE,BK=C1*≠P(04),BY LOW BK;
REPORTER;
FORMAT;
TITLE IS A I R F I E L D  R E P O R T;
TITLE IS THE PARAMETERS USED IN THE RETRIEVAL ARE ≠P(01),≠P(02),≠P(03),
AND ≠P(04);
HEADING IS C1,KK,CODE,BK;
DETAIL IS C1,KK,CODE,BK;
FOOT IS END OF PAGE;
SPACE TO 16 IN T1 BEFORE I1;
SPACE TO 15,47,63 IN H1,H1,H1 BEFORE I2,I3,I4;
SPACE TO 15,47,53 IN D1,D1,D1 BEFORE I2,I3,I4;
SPACE TO 18 IN F1 BEFORE I1;
GENERATE;
RETRIEVAL;
MESSAGE FILE IS OUTPUT;
END MACRO;
$       ;
RETRIEVAL;
PRINT C1,CODE,MAGVAR WHERE C1 GT 0000500140 AND C1 LT 0000500210;
C1=0000500150
CODE=N
MAGVAR=012E

C1=0000500160
CODE=N
MAGVAR=015E

C1=0000500170
CODE=N
MAGVAR=018E

C1=0000500180
CODE=N
MAGVAR=020E

C1=0000500190
CODE=N
MAGVAR=024E

C1=0000500200
CODE=N
MAGVAR=000E

--- END OF RETRIEVAL ---
IF C1 RANGE(0000500140,0000500210) MASS UPDATE;
CODE=E;
READY
MAGVAR=AECD;
READY
FINIS;
UPDATE COMPLETE, ENTER COMMAND
RETRIEVAL;
PRINT C1,CODE,MAGVAR WHERE C1 EQ 0000500170;
C1=0000500170
CODE=E
MAGVAR=AECD
```

```
--- END OF RETRIEVAL ---
PRINT C1,NAME,OPSTATUS WHERE C1 EQ RANGE[500190/500260];
C1=0000500190
NAME=GRAND HAVEN MEM            24
OPSTATUS=A

C1=0000500200
NAME=GRAND HAVEN MEM            25
OPSTATUS=A

C1=0000500210
NAME=GRAND HAVEN MEM            26
OPSTATUS=A

C1=0000500220
NAME=GRAND HAVEN MEM            27
OPSTATUS=A

C1=0000500230
NAME=GRAND HAVEN MEM            28
OPSTATUS=A

C1=0000500240
NAME=GRAND HAVEN MEM            29
OPSTATUS=A

C1=0000500250
NAME=GRAND HAVEN MEM            30
OPSTATUS=A

C1=0000500260
NAME=GRAND HAVEN MEM            31
OPSTATUS=A

--- END OF RETRIEVAL ---
IF (RG4(ANY) EQ N) AND (C1 LT 0000500160) PRINT C1,RG2(EACH);
C1=0000500100
RG2( 1)=090
RG2( 2)=270

C1=0000500110
RG2( 1)=090
RG2( 2)=270

C1=0000500120
RG2( 1)=090
RG2( 2)=270

C1=0000500130
RG2( 1)=090
RG2( 2)=270

C1=0000500140
RG2( 1)=090
RG2( 2)=270

C1=0000500150
RG2( 1)=090
RG2( 2)=270

--- END OF RETRIEVAL ---
IF C1+50 GT 500250 REPORT C1,NAME,CODE,IX=C1/100;
```

```
C1              NAME                                CODE     IX

0000500210     GRAND HAVEN MEM           26     N                   5002
0000500220     GRAND HAVEN MEM           27     N                   5002
0000500230     GRAND HAVEN MEM           28     N                   5002
0000500240     GRAND HAVEN MEM           29     N                   5002
0000500250     GRAND HAVEN MEM           30     N                   5002
0000500260     GRAND HAVEN MEM           31     N                   5002
0000500270     GRAND HAVEN MEM           32     N                   5002
0000500280     GRAND HAVEN MEM           33     N                   5002
0000500290     GRAND HAVEN MEM           34     N                   5002

--- END OF RETRIEVAL ---
PRINT RECORD(DETAIL) WHERE C1 EQ 500170:
C1=0000500170
NAME=GRAND HAVEN MEM            22
CODE=B
COORDS=430200N0861200W
DEGLAT=43
NSDIRECT=N
EWDIRCT=W
OPSTATUS=A
ELEV=00603
MAINCODE=0
LIGHT=7
RUNCOND=G
RG2(1)=090
RG3(1)=U
RG4(1)=N
RG2(2)=270
RG3(2)=U
RG4(2)=N
RNWYELEV=U
RENDID=270
ROVERUN=N
RUNOBS=270
MAGVAR=ABCD
RECMARK=]

--- END OF RETRIEVAL ---
CATALOG:
PFN=
  AFIELD;
  ID=CDC;
CY=40;
PERMANENT FILE FUNCTION COMPLETE
  FINIS:
SAVEPF TESTA,TESTAPFN;
PERMANENT FILE FUNCTION COMPLETE
EXIT:
```

```
IDENTIFICATION DIVISION.
PROGRAM-ID. LP25.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT FILE-1 ASSIGN TO OUTPUT.
DATA DIVISION.
FILE SECTION.
FD  FILE-1
    LABEL RECORDS ARE OMITTED
    DATA RECORD IS REC-1.
01  REC-1.
    02  C-C1 PICTURE X(10).
    02  C-NAME PICTURE X(27).
    02  C-CODE PICTURE X(1).
    02  C-C9 PICTURE X(10).
WORKING-STORAGE SECTION.
77  NAME VALUE ≠NAME≠ PICTURE X(4).
77  CODES VALUE ≠CODE≠ PICTURE X(4).
77  C1 VALUE ≠C1≠ PICTURE X(2).
77  C9 VALUE ≠C9999≠ PICTURE X(5).
01  TEMPS.
    02  C1VALUE VALUE ≠0000500220≠ PICTURE X(10).
    02  C2VALUE VALUE ≠0000500210≠ PICTURE X(10).
    02  RECC USAGE IS COMP-1  PICTURE 9(10).
    02 ERFLG USAGE IS COMP-1  PICTURE 9(10).
    02  NEWCODE PICTURE X(1).
01  FILESTUF.
    02  DATABASE VALUE ≠AFIELD≠ PICTURE X(6).
    02  DBP VALUE ≠ID=CDC:≠ PICTURE X(7).
    02  REF VALUE ≠TESTA≠ PICTURE X(5).
```

```
           PROCEDURE DIVISION.
           START.
               OPEN OUTPUT FILE-1.
               ENTER MARSOPN USING DATABASE, DBP.
               ENTER MARSDEF USING REF, NAME, C-NAME, CODES, O-CODE, C1,
                          O-C1, C9, O-C9.
               ENTER MARSGET USING REF, C1, C1VALUE, RECC, ERFLG, ERPARA.
               WRITE REC-1.
               MOVE ≠Z≠ TO C-CODE.
               ENTER MARSTOR USING REF.
               ENTER MARSREP.
               ENTER MARSGET USING REF, C1, C2VALUE, RECC, ERFLG, ERPARA.
               WRITE REC-1.
               ENTER MARSGET USING REF, C1, C1VALUE, RECC, ERFLG, ERPARA.
               WRITE REC-1.
               ENTER MARSCLO.
               CLOSE FILE-1.
               STOP RUN.
           ERPARA.
               DISPLAY ≠FAIL≠.
               STOP RUN.
  LP25    LENGTH IS  000204
FIELD LENGTH NEEDED FOR COBOL 012500

000500220GRAND HAVEN MEM          27N          13

000500210GRAND HAVEN MEM          26N          12

000500220GRAND HAVEN MEM          27Z          13
```

```
MARS VI V2.1  05/01/72
$                                          :
$          THIS IS A MACRO EXPANDING       :
$                                          :
ATTACH;
PFN=
AFIELD:
ID=CDC;
CY=40;
PERMANENT FILE FUNCTION COMPLETE
FINIS;
GETPF  TESTA,TESTAPFN:
PERMANENT FILE FUNCTION COMPLETE
RETRIEVAL;
EXPAND OFF:
RETRIEVAL;
REPORT FILE IS X:
IF C1 LT 0000500200 SUBSET C1,NAME,CODE,LL=ELEV-3:
--- END OF RETRIEVAL ---
SUBSET COMPLETE-     10 RECORDS-  6 WORDS EACH- 9 CHARS OF FILL
REPORT FILE IS OUTPUT:
REWIND X;
TRANS FILE IS UF,CP:
MASS UPDATE BY C1;
DEFINITION:
 DATA BASE:
 ENDEF:
INVALID TRANSACTION RECORD, KEY VALUE NOT IN FILE SPECIFIED OR FILE INACTIVE
RTESTA  0000500350ST. LOUIS MISSOURI   USA  S

INVALID TRANSACTION RECORD, KEY VALUE ALREADY IN FILE SPECIFIED
ITESTA  0000500200ST. LOUIS  MISSOURI   USA
        E
INVALID TRANSACTION RECORD, FILE NAME NOT RECOGNIZED
SXXXXXD 0000500250ST. LOUIS MISSOURI   USA

INVALID TRANSACTION RECORD, FILE NAME NOT RECOGNIZED
PXFIELD  0000500270

INVALID TRANSACTION RECORD, KEY VALUE NOT IN FILE SPECIFIED OR FILE INACTIVE
RTESTA  0050019 0ST. LOUIS  MISSOURI   USA    S

    10 TRANSACTIONS PROCESSED
UPDATE COMPLETE, ENTER COMMAND
RETRIEVAL:
PRINT C1,NAME,OPSTATUS WHERE C1 GT 0000500190 AND C1 LT 0000500260;
C1=0000500210
NAME=GRAND HAVEN MEM          26
OPSTATUS=A

C1=0000500220
NAME=GRAND HAVEN MEM          27
OPSTATUS=A

C1=0000500230
NAME=GRAND HAVEN MEM          28
OPSTATUS=A
```

```
C1=0000500240
NAME=GRAND HAVEN MEM           29
OPSTATUS=A

C1=0000500250
NAME=GRAND HAVEN MEM           30
OPSTATUS=A

--- END OF RETRIEVAL ---
PRINT RG2(1),RG3(2) WHERE SAME;
RG2(1)=090
RG3(2)=U

RG2(1)=090
RG3(2)=U

RG2(1)=090
RG3(2)=U

RG2(1)=090
RG3(2)=U

RG2(1)=090
RG3(2)=U

--- END OF RETRIEVAL ---
TRANS FILE IS INPUT:
MASS UPDATE BY C1;
NODEF;
READY
C1=0000500150,CODE=Z:
READY
C1=0000500200,CODE=Z;
KEY VALUE DOES NOT EXIST OR FILE INACTIVE
READY
C1=0000500250,ELEV=DEGLAT+10;
READY
FINIS:
UPDATE COMPLETE, ENTER COMMAND
RETRIEVAL;
PRINT C1,CODE,MAGVAR WHERE C1 EQ 0000500150 OR C1 EQ 0000500200:
C1=0000500150
CODE=Z
MAGVAR=ABCD

--- END OF RETRIEVAL ---
↑REPORT1,N,EQ,COL(11 TO 37),2↑

                  A I R F I E L D   R E P O R T
THE PARAMETERS USED IN THE RETRIEVAL ARE N     EQ     COL(11 TO 37)     AND 2
C1                KK                                   CODE  BK
0000500290        GRAND HAVEN MEM              34      N        1000580.00
0000500280        GRAND HAVEN MEM              33      N        1000560.00
0000500270        GRAND HAVEN MEM              32      N        1000540.00
0000500260        GRAND HAVEN MEM              31      N        1000520.00
0000500240        GRAND HAVEN MEM              29      N        1000480.00
0000500230        GRAND HAVEN MEM              28      N        1000460.00
0000500210        GRAND HAVEN MEM              26      N        1000420.00
0000500140        GRAND HAVEN MEM              19      N        1000280.00
0000500130        GRAND HAVEN MEM              18      N        1000260.00
0000500100        GRAND HAVEN MEM              15      N        1000200.00
```

```
              :
IF SAME COUNT RECORD;
COUNT =     10
--- END OF RETRIEVAL ---
DESCRIBE MACRO SKELETONS:

MACRO NAME = MYMACRO
EXPAND ON;
$                                         :
$          THIS IS A MACRO EXPANDING      ;
$                                         :
ATTACH:  AFIELD: ID=CCC;CY=40;  FINIS; GETPF  TESTA,TESTAPFN;
RETRIEVAL:
EXPAND OFF;
END MACRO:

MACRO NAME = REPORT1
MESSAGE FILE IS NULL;
RETRIEVAL;
IF   COL(38) EQ ≠P(01) 'AND (ELEV-3)/10 EQ ≠P(02) QUALIFY C1,KK=≠P(03),
CODE,BK=C1*≠P(04),BY LOW BK:
REPORTER;
FORMAT;
TITLE IS A I R F I E L D  R E P O R T;
TITLE IS THE PARAMETERS USED IN THE RETRIEVAL ARE ≠P(01),≠P(02),≠P(03),
AND ≠P(04):
HEADING IS C1,KK,CODE,BK;
DETAIL IS C1,KK,CODE,BK;
FOOT IS END OF PAGE;
SPACE TO 16 IN T1 BEFORE I1:
SPACE TO 15,47,63 IN H1,H1,H1 BEFORE I2,I3,I4:
SPACE TO 15,47,63 IN D1,D1,D1 BEFORE I2,I3,I4;
SPACE TO 18 IN F1 BEFORE I1;
GENERATE;
RETRIEVAL;
MESSAGE FILE IS OUTPUT:
END MACRO;
END DESCRIPTION
PURGE;
PFN=
AFIELD;
CY=40;
PERMANENT FILE FUNCTION COMPLETE
EXIT;

21.08.51.VMARS1C
21.08.51.VMARS,CM65000,T250.
21.08.51.COPYBR(INPUT,UP,1)
21.08.52.COPYBR(INPUT,TEST,1)
21.08.52.REWIND(UP,TEST,ADDLOAD)
21.08.53.COPYSBF(UP,OUTPUT)
21.08.54.COPYSBF(TEST,OUTPUT)
21.08.54.REWIND(ADDLOAD,TEST,UP)
21.08.55.REQUEST(TESTA,*PF)
21.08.55.MARS(PF)
21.08.57.      2.185 RT SECONDS LOAD TIME
21.08.57.REQUEST(TAPE8,*,PF)
21.08.57.REQUEST(TAPE12,*,PF)
21.08.57.REQUEST(TAPE11,*,PF)
21.08.57.REQUEST(TAPE10,*,PF)
```

```
21.08.58.REQUEST(TAPE9,*,PF)
21.08.58.REQUEST(TAPE8,*,PF)
21.08.58.REQUEST(TAPE7,*,PF)
21.08.58.REQUEST(TAPE6,*,PF)
21.08.59.( TAPE6   ASSIGNED TO EST 10 )
21.08.59.( TAPE6   ASSIGNED TO EST 10 )
21.08.59.( TAPE7   ASSIGNED TO EST 10 )
21.08.59.( TAPE8   ASSIGNED TO EST 10 )
21.08.59.( TAPE9   ASSIGNED TO EST 10 )
21.08.59.( TAPE10  ASSIGNED TO EST 10 )
21.08.59.( TAPE11  ASSIGNED TO EST 10 )
21.08.59.( TAPE12  ASSIGNED TO EST 10 )
21.09.04.( TESTA   ASSIGNED TO EST 10 )
21.09.05.SRTMRG
21.09.05.         41 RECORDS SORTED
21.09.06.          1 INTERNAL MERGE PHASES
21.09.09.SRTMRG
21.09.09.         41 RECORDS SORTED
21.09.09.          1 INTERNAL MERGE PHASES
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.09.59.INDEX BUFFER NOT SPECIFIED IN FET
21.10.06.RETURN(TESTA)
21.10.07.ATTACH(MARSCOB,MARSCOB,ID=MARS,MR=1)
21.10.07.PF CYCLE NO. = 001
21.10.07.ATTACH(TESTA,TESTAPFN,ID=CDC)
21.10.07.PF CYCLE NO. = 040
21.10.07.COBOL(L)
21.10.10.      1.857 RT SECONDS LOAD TIME
21.10.11.COMPILING LP25
21.10.17. 000 E AND      T/U  DIAGNOSTICS ISSUED
21.10.17. FIELD LENGTH NEEDED FOR COBOL 052500
21.10.18.       .415 CP SECONDS COMPILATION TIME
21.10.18.END COBOL
21.10.18.LOAD(MARSCOB)
21.10.18.LGO.
21.10.30.    12.496 RT SECONDS LOAD TIME
21.10.37.RETURN(TESTA)
21.10.38.ATTACH(TAPE6,AFIELD2,ID=CDC,CY=40)
21.10.38.MARS(M=MYMACRO)
21.10.41.     2.260 RT SECONDS LOAD TIME
21.11.01.SRTMRG
21.11.01.          3 RECORDS SORTED
21.11.02.          1 INTERNAL MERGE PHASES
21.11.05.SRTMRG
21.11.06.          3 RECORDS SORTED
21.11.06.          1 INTERNAL MERGE PHASES
21.11.08.INDEX BUFFER NOT SPECIFIED IN FET
21.11.08.INDEX BUFFER NOT SPECIFIED IN FET
21.11.08.INDEX BUFFER NOT SPECIFIED IN FET
21.11.09.INDEX BUFFER NOT SPECIFIED IN FET
21.11.09.INDEX BUFFER NOT SPECIFIED IN FET
21.11.09.INDEX BUFFER NOT SPECIFIED IN FET
21.11.09.INDEX BUFFER NOT SPECIFIED IN FET
```

```
21.11.58.INDEX BUFFER NOT SPECIFIED IN FET
21.11.58.REWIND(TESTA)
21.11.58.COPYSBF(TESTA,OUTPUT)
21.11.59.REWIND(X)
21.11.59.COPYSBF(X,OUTPUT)
21.12.00.PURGE(TESTA)
21.12.00.CPA     10.173 SEC.
21.12.00.PP     166.305 SEC.
```

RELEASE DESCRIPTION

6000/241/IGS Version 2.1 runs under INTERCOM 4.1 in conjunction with SCOPE 3.4 to provide a remote interactive graphics capability in IGS mode as well as the capability to converse with INTERCOM in COMMAND mode.

The 6000/241/IGS package allows the user to write a central memory application in FORTRAN Extended and interact with that program at the console through the keyboard, function keys, or lightpen. The IGS routines are callable only under FORTRAN Extended. The user, in COMMAND mode, has all the normal INTERCOM commands and capabilities available to a CRT terminal user under INTERCOM 4.1.

The installation of 6000/241/IGS Version 2.1 requires the previous installation of INTERCOM 4.1 and those products required by INTERCOM 4.1 under SCOPE 3.4.

6000/241/IGS Version 2.1 will be made available under SCOPE 3.3 at a later date.

HARDWARE CONFIGURATION

In addition to the minimum hardware required by SCOPE 3.4 and INTERCOM 4.1, 6000/241/IGS requires the following equipment for communication and operations:

1. A 241-1 Graphics Subsystem with a 248-2 Memory Expansion for each terminal.

2. The communication between the remote terminal and the central site requires a 201A or 201B Data Set or a 358-2 Transceiver. This communication equipment must be connected to the 6671 multiplexer as required for INTERCOM 4.1.

Hardware Options

No other hardware options are supported by this software.

RELEASE MATERIALS

6000/241/IGS Version 2.1 release material consists of 2 magnetic tapes which contain the 6000/241/IGS 2.1 OLDPLs in UPDATE format.

PL24 Central Memory Routines
PL25 Remote Routines

PL24 consists of the following files:

OLDPL File

File 1: 6000/241/IGS Version 2.1 CM OLDPL in UPDATE format. (This contains the central memory routines required for IGS operation.)

Binary File

File 2: Central memory routines. This file is part of the LIBRARY IGS241.

Installation Decks

File 3:	CMFINST	Performs new installation of central memory routines.
File 4:	CMFMANT	Updates OLDPL (File 1) and performs system EDITLIB functions.
File 5:	CMFBIN	Installs from Binary File 2
File 6:	JOB3	Verification deck

PL25 consists of the following files:

OLDPL File

File 1: 6000/241/IGS Version 2.1 RM OLDPL in UPDATE format. (This contains the remote terminal resident software and utilities to assemble and format the code.)

Binary Files

File 2:	BNG	GRASS Assembler	Part of LIBRARY IGS241
File 3 :	BNB	BINTAP routine	
File 4:	GIN	Central memory transfer routine for GFINT	
File 5:	GFINT	Grid resident routine	

Installation Decks

File 6:	RMFINST	Performs new installation of RM routines
File 7:	RMFMANT	Updates OLDPL (File 1) and performs system EDITLIB functions
File 8:	RMFBIN	Install from Binary Files (2-5)

NEW FEATURES

1. 6000/241/IGS Version 2.1 runs under INTERCOM 4.1 with no modifications made to SCOPE 3.4 or INTERCOM 4.1 routines.

2. The GRIDRES routine has been rewritten to conform to the changes in the central memory Q8GEXEC routine. The internal table structure in GRIDRES has also been changed to reduce the amount of storage required for processing and the time required to process the graphics data.

3. GUPNTS routine has been added which provides point plot capability.

4. The GUARC and GULINE routines have been added. These routines provide line and arc scissoring capabilities in the central processor.

5. A new error processing feature has been added which will display error messages at the remote terminal as they are generated. To continue, the user is required to pick the error message with the lightpen.

LIMITATIONS

1. All limitations as they exist for INTERCOM 4.1 apply to 6000/241/IGS Version 2.1 when operating in COMMAND mode.

2. IGS application programs must use FORTRAN Extended to call the IGS subroutines.

3. The 6000/241/IGS Version 2.1 communication with INTERCOM assumes the place of a 217-13 or 217-14 CRT remote terminal, (an ANSI terminal). Please note the restriction in INTERCOM 4.1 concerning the prohibition of mixing BCD and ANSI 200 User Terminals.

4. The simultaneous use of interactive graphics data and non-IGS data at a remote terminal is not allowed. During IGS mode, INPUT and OUTPUT files in the central memory program should be disconnected to eliminate the possibility of mixing IGS and non-IGS data.

GENERAL PROCEDURES

The installation of the complete 6000/241/IGS system requires installing from the 6000/241/IGS program libraries. The jobs required to install the system are released as installation decks. The installation of INTERCOM 4.1 is required before attempting to use 6000/241/IGS Version 2.1 but not necessary for the proper installation of IGS.

Central Memory Requirements

The standard installation of 6000/241/IGS Version 2.1 does not require any CMR space beyond that required by INTERCOM 4.1 for multiplexer tables and buffers.

Installation Parameters

See INTERCOM 4.1 installation instructions.

Port Definition Entry

For each 6671 multiplexer port which is to service a 241 GRID terminal, a port definition entry macro-CRTA must be used in the CMR MULTIPLEXER SUBTABLE.

Password

A password entry is required to initialize the 241 memory using the Deadstart Initialization Procedure:

```
ADD U=G41,P=G41,F=60000,T=777
```

INSTALLATION PROCEDURES

The 6000/241/IGS Version 2.1 system is to be installed under SCOPE 3.4. The following set of jobs will produce a new system with IGS installed. The jobs are included with the release program libraries. (See files 3, 4, 5, 6 of PL24 and 6, 7, 8 of PL25 under Release Materials.)

To create a deadstart tape from the running system, modify job SCOPE3 from the SCOPE 3.4 PL by inserting this card in front of card 22 (the COMPLETE card): INCLUDE(IGS241,SYSTEM,DS). A similar card must be inserted in front of card 038 of deck SCOPE2 (see section 1, MODEL JOBS).

Deck RMFMANT contains two PURGE cards which must be removed if no previous level of RMFMANT exists in the system.

Files that contain the necessary installation decks may be obtained by running a job of the type:

```
jobcard                           jobcard
REQUEST,PL24,E.                   REQUEST,PL25,E.
SKIPF(PL24,2,17)          or      SKIPF(PL25,5,17)
COPYBF(PL24,PUNCH,3)              COPYBF(PL25,PUNCH,3)
UNLOAD,PL24.                      UNLOAD,PL25.
6/7/8/9                           6/7/8/9
```

```
CMFINST,CM60000,T2000,MT2.
COMMENT.  THIS DECK MAY BE USED TO INSTALL 241IGS VERSION 2.1
COMMENT.     CENTRAL MEMORY ROUTINES
COMMENT.
COMMENT.  THIS JOB WILL REQUIRE PL24 AS INPUT AND WILL
COMMENT.     EDITLIB -IGS241- INTO SYSTEM.
COMMENT.
COMMENT.                 PL24 FOLLOWS
COMMENT.
COMMENT.                    FILE1  -- UPDATED OLDPL OF CENTRAL MEMORY IGS
COMMENT.                              ROUTINES WITH MODIFICATIONS.
COMMENT.
COMMENT.                    FILE2  -- RELOCATABLE OBJECT CODE FOR 241IGS
COMMENT.                              CENTRAL MEMORY ROUTINES.
COMMENT.
COMMENT.                    FILE3  -- A COPY OF THIS DECK.
COMMENT.
COMMENT.                    FILE4  -- A COPY OF THE MAINTENANCE DECK.
COMMENT.
COMMENT.                    FILE5  -- A COPY OF A DECK TO INSTALL FROM THE
COMMENT.                              BINARIES ON THIS TAPE OR THE BINARIES
COMMENT.                              ON A MAINTENANCE TAPE.
COMMENT.
COMMENT.                    FILE6  -- A COPY OF THE VERIFICATION PROGRAM.
COMMENT.
LABEL(PL24,R,L=241CMF2P1*3P4,D=HI)
COPYBF(PL24,CMO)
COPYBF(PL24,NIL,4)
COPYBF(PL24,PUNCH)
REWIND(CMO)
UNLOAD(NIL)
UNLOAD(PL24)
COMMENT.  THIS UPDATE MAKES NEWPL WITH  INSTALLATION MODIFICATIONS.
UPDATE(P=CMO,F,C=CMN)
UNLOAD(CMO)
FTN(I=CMN,B=BN1,S=PFMTEXT,S=CPCTEXT,OPT=1)
COMMENT.  MAKES LIBRARY OF ABOVE ROUTINES = IGS241.
EDITLIB(SYSTEM)
7/8/9
*/ MODIFICATIONS TO CENTRAL MEMORY ROUTINES SHOULD FOLLOW THIS CARD.
*/
7/8/9
REWIND(BN1)
READY(SYSTEM,OLD)
LIBRARY(IGS241,OLD)
ADD(*,BN1)
FINISH.
COMPLETE.
ENDRUN.
6/7/8/9
```

```
RMFINST,CM60000,T2000,MT2.
COMMENT.  THIS DECK MAY BE USED TO INSTALL 241IGS VERSION 2.1
COMMENT.       REMOTE ROUTINES.
COMMENT.
COMMENT.  THIS JOB WILL REQUIRE PL25 AS INPUT AND WILL
COMMENT.    ADD TO LIBRARY (IGS241) IN THE SYSTEM.
COMMENT.
COMMENT.            PL25 FOLLOWS
COMMENT.
COMMENT.              FILE1  -- UPDATED OLDPL OF GRIDRES AND UTILITIES
COMMENT.                        WITH MODIFICATIONS.
COMMENT.
COMMENT.              FILE2  -- RELOCATABLE OBJECT CODE FOR
COMMENT.                        GRASS ASSEMBLER.
COMMENT.
COMMENT.              FILE3  -- RELOCATABLE OBJECT CODE FOR
COMMENT.                        BINTAP ROUTINE.
COMMENT.
COMMENT.              FILE4  -- RELOCATABLE OBJECT CODE FOR
COMMENT.                        GINIT ROUTINE.
COMMENT.
COMMENT.              FILE5  -- RELOCATABLE OBJECT CODE FOR
COMMENT.                        GRID RESIDENT FORMATED BY BINTAP.
COMMENT.
COMMENT.              FILE6  -- A COPY OF THIS DECK.
COMMENT.
COMMENT.              FILE7  -- A COPY OF THE MAINTENANCE DECK.
COMMENT.
COMMENT.              FILE8  -- A COPY OF A DECK TO INSTALL FROM THE
COMMENT.                        BINARIES ON THIS TAPE OR THE BINARIES
COMMENT.                        ON A MAINTENANCE TAPE.
COMMENT.
LABEL(PL25,R,L=241RMF2P1*3P4,D=HI)
COPYBF(PL25,CMO)
REWIND(CMO)
UNLOAD(PL25)
COMMENT.  THIS UPDATE MAKES NEWPL WITH  INSTALLATION MODIFICATIONS.
UPDATE(P=CMO,F,N=CMN)
UNLOAD(CMO)
COMMENT.  THIS UPDATE PUTS THE GRASS ASSEMBLER SOURCE ON A FILE NAMED
COMMENT.  GRAS FOR INPUT TO THE RUN COMPILER.
UPDATE(P=CMN,Q,C=GRAS)
COMMENT.  THIS UPDATE PUTS THE GRID RESIDENT SOURCE ON A FILE NAMED
COMMENT.  GRES FOR INPUT TO THE GRASS ASSEMBLER.
UPDATE(P=CMN,Q,C=GRES)
COMMENT.  THIS UPDATE PUTS BINTAP SOURCE ON A FILE NAMED BINT
COMMENT.  FOR INPUT TO COMPASS.
UPDATE(P=CMN,Q,C=BINT)
COMMENT.  THIS UPDATE PUTS GINIT SOURCE ON A FILE NAMED GINT
COMMENT.  FOR INPUT TO COMPASS.
UPDATE(P=CMN,Q,C=GINT)
```

```
RUN(S,,,GRAS,,BN2,100000)
COMMENT.  THIS LOAD CREATES A FILE NAMED BNG WHICH CONTAINS THE GRASS
COMMENT.  ASSEMBLER TO BE EDITLIBED INTO THE SYSTEM.
LIBRARY(RUN2P3)
LOAD(BN2)
NOGO.
COMPASS(I=BINT,B=BNB,S=PFMTEXT,S=CPCTEXT)
COMMENT.  MAKES LIBRARY OF ABOVE ROUTINES = IGS241.
EDITLIB(SYSTEM)
REQUEST(GIN,*PF)
COMMENT.  GINIT  IS A PROGRAM USED TO TRANSMIT GRID RESIDENT.
COMPASS(I=GINT,B=GIN,S=PFMTEXT,S=CPCTEXT)
CATALOG(GIN,GIN,ID=G41,RP=999)
LIBRARY(IGS241)
GRASS(GRES,BNT)
REWIND(BNT)
REQUEST(GFINT,*PF)
COMMENT.  GFINT IS A FILE CONTAINING FORMATED GRID RESIDENT.
BINTAP(GFINT)
CATALOG(GFINT,GFINT,ID=IGS241,RP=999)
7/8/9
*/  MODIFICATIONS TO GRASS SHOULD FOLLOW THIS CARD.
*/
*/  MODIFICATIONS TO GRID RESIDENT SHOULD FOLLOW THIS CARD.
*/
*/  MODIFICATIONS TO BINTAP SHOULD FOLLOW THIS CARD.
*/
*/  MODIFICATIONS TO GINIT SHOULD FOLLOW THIS CARD.
*/
7/8/9
*COMPILE GRASS
7/8/9
*COMPILE GRIDRES
7/8/9
*COMPILE BINTAP
7/8/9
*COMPILE GINIT
7/8/9
REWIND(BNG)
REWIND(BNB)
READY(SYSTEM,OLD)
LIBRARY(IGS241,OLD)
ADD(*,BNB,AL=1)
ADD(*,BNG,AL=1)
FINISH.
COMPLETE.
ENDRUN.
6/7/8/9
```

```
CMFMANT,CM60000,T2000,MT2.                                              001
COMMENT.   CENTRAL MEMORY ROUTINES.                                     002
COMMENT.  THIS DECK IS ISSUED TO FACILITATE THE MAINTENANCE OF          003
COMMENT.  241IGS VERSION 2.1.  THIS JOB REQUIRES A MAINTENANCE          004
COMMENT.  FORM TAPE (EITHER THE PL24 TAPE FROM THE INSTALLATION         005
COMMENT.  RUN OR THE OUTPUT TAPE FROM A PREVIOUS MAINTENANCE RUN)       006
COMMENT.  AS INPUT.  THE INPUT TAPE WILL BE REQUESTED AS  PL24,         007
COMMENT.   AN OUTPUT TAPE WILL BE CREATED AS FOLLOWS-                   008
COMMENT.                                                                009
COMMENT.       NPL24  -- A NEW MAINTENANCE TAPE WITH MODIFICATIONS.     010
COMMENT.                                                                011
COMMENT.                                                                012
COMMENT.                                                                013
COMMENT.  IF AN EDITLIB IS NOT DESIRED, DELETE                          014
COMMENT.    EDITLIB(SYSTEM)    -  CARD 37
COMMENT.    7/8/9 THRU ENDRUN.   -CARDS 53 THRU 60
COMMENT.                                                                017
COMMENT.                                                                018
LABEL(PL24,R,L=241CMF2P1*3P4,D=HI)                                      019
COPYBF(PL24,CMO)                                                        020
COPYBF(PL24,NIL)                                                        021
COPYBF(PL24,FILE3)                  INSTALLATION DECK                   022
COPYBF(PL24,FILE4)                  MAINTENANCE DECK                    023
COPYBF(PL24,FILE5)                  INSTALL FROM BINARIES               024
COPYBF(PL24,FILE6)                  VERIFICATION DECK                   025
REWIND(CMO,FILE3,FILE4,FILE5,FILE6)                                     026
UNLOAD(NIL)                                                             028
UNLOAD(PL24)                                                            029
COMMENT.  THIS UPDATE MAKES NEWPL WITH  INSTALLATION MODIFICATIONS.     030
UPDATE(P=CMO,F,N=CMN)                                                   031
UNLOAD(CMO)                                                             032
COMMENT.  THIS UPDATE PUTS SOURCE ON  FILE CMU FOR FTNX USE.            033
UPDATE(P=CMN,F,C=CMU)                                                   034
FTN(I=CMU,B=BN1,S=PFMTEXT,S=CPCTEXT,OPT=1)                              035
COMMENT.  MAKES LIBRARY OF ABOVE ROUTINES = IGS241.                     036
EDITLIB(SYSTEM)                                                         037
REWIND(BN1)                                                             038
LABEL(NPL24,W,L=241CMF2P1*3P4,T=999,D=HI)                               039
UPDATE(P=CMN,F,N=NPL24,8)                                               040
REWIND(NPL24)                                                           041
COPYBF(NPL24,Z)                                                         042
COPYBF(BN1,NPL24)                                                       043
COPYBF(FILE3,NPL24)                                                     044
COPYBF(FILE4,NPL24)                                                     045
COPYBF(FILE5,NPL24)                                                     046
COPYBF(FILE6,NPL24)                                                     047
UNLOAD(NPL24)                                                           048
7/8/9                                                                   049
*/  MODIFICATIONS TO CENTRAL MEMORY ROUTINES SHOULD FOLLOW THIS CARD.   050
*/                                                                      051
7/8/9                                                                   052
7/8/9                                                                   053
REWIND(BN1)                                                             054
READY(SYSTEM,OLD)                                                       055
LIBRARY(IGS241,OLD)                                                     056
REPLACE(BYTSUBG+Q8GEXEC,BN1)                                            057
FINISH.                                                                 058
COMPLETE.                                                               059
ENDRUN.                                                                 060
7/8/9                                                                   061
6/7/8/9                                                                 062
```

```
001 RMFMANT,CM60000,T2000,MT2.
002 COMMENT.  THIS DECK IS ISSUED TO FACILITATE THE MAINTENANCE OF
003 COMMENT.  241IGS VERSION 2.1.  THIS JOB REQUIRES A MAINTENANCE
004 COMMENT.  FORM TAPE (EITHER THE PL25 TAPE FROM THE INSTALLATION
005 COMMENT.  RUN OR THE OUTPUT TAPE FROM A PREVIOUS MAINTENANCE RUN)
006 COMMENT.  AS INPUT.  THE INPUT TAPE WILL BE REQUESTED AS  PL25,
007 COMMENT.   AN OUTPUT TAPE WILL BE CREATED AS FOLLOWS-
008 COMMENT.
009 COMMENT.        NPL25  -- A NEW MAINTENANCE TAPE WITH MODIFICATIONS.
010 COMMENT.
011 COMMENT.  IF AN EDITLIB IS NOT DESIRED, DELETE
012 COMMENT.  EDITLIB(SYSTEM) THRU REQUEST(GIN,*PF) -CARDS 48 TO 50
013 COMMENT.  CATALOG(GIN,GIN,...) AND PURGE(JUNC,...) -CARDS 53 AND 54
014 COMMENT.  REQUEST(GFINT,*PF) AND CATALOG(GFINT,...) CARDS 58 AND 60
015 COMMENT.  7/8/9 THRU ENDRUN -CARDS 91 TO 100
016 LABEL(PL25,R,L=241RMF2P1*3P4,D=HI)
017 COPYBF(PL25,CMO)
018 COPYBF(PL25,NIL,4)
019 COPYBF(PL25,FILE6)
020 COPYBF(PL25,FILE7)
021 COPYBF(PL25,FILE8)
022 REWIND(CMO,FILE6,FILE7,FILE8)
023 UNLOAD(NIL)
024 UNLOAD(PL25)
025 COMMENT.  THIS UPDATE MAKES NEWPL WITH  INSTALLATION MODIFICATIONS.
026 UPDATE(P=CMO,F,N=CMN)
027 UNLOAD(CMO)
028 COMMENT.  THIS UPDATE PUTS THE GRASS ASSEMBLER SOURCE ON A FILE NAMED
029 COMMENT.  GRAS FOR INPUT TO THE RUN COMPILER.
030 UPDATE(P=CMN,Q,C=GRAS)
031 COMMENT.  THIS UPDATE PUTS THE GRID RESIDENT SOURCE ON A FILE NAMED
032 COMMENT.  GRES FOR INPUT TO THE GRASS ASSEMBLER.
033 UPDATE(P=CMN,Q,C=GRES)
034 COMMENT.  THIS UPDATE PUTS BINTAP SOURCE ON A FILE NAMED BINT
035 COMMENT.  FOR INPUT TO COMPASS.
036 UPDATE(P=CMN,Q,C=BINT)
037 COMMENT.  THIS UPDATE PUTS GINIT SOURCE ON A FILE NAMED GINT
038 COMMENT.  FOR INPUT TO COMPASS.
039 UPDATE(P=CMN,Q,C=GINT)
040 RUN(S,,,GRAS,,BN2,100000)
041 COMMENT.  THIS LOAD CREATES A FILE NAMED BNG WHICH CONTAINS THE GRASS
042 COMMENT.  ASSEMBLER TO BE EDITLIBED INTO THE SYSTEM.
043 LIBRARY(RUN2P3)
044 LOAD(BN2)
045 NOGO.
046 COMPASS(I=BINT,B=BNB,S=PFMTEXT,S=CPCTEXT)
047 COMMENT.  MAKES LIBRARY OF ABOVE ROUTINES = IGS241.
048 EDITLIB(SYSTEM)
049 PURGE(JUNK,GIN,ID=G41)
050 REQUEST(GIN,*PF)
051 COMMENT.  GINIT  IS A PROGRAM USED TO TRANSMIT GRID RESIDENT.
052 COMPASS(I=GINT,B=GIN,S=PFMTEXT,S=CPCTEXT)
053 CATALOG(GIN,GIN,ID=G41,RP=999)
054 PURGE(JUNC,GFINT,ID=IGS241)
055 LIBRARY(IGS241)
056 GRASS(GRES,BNT)
057 REWIND(BNT)
058 REQUEST(GFINT,*PF)
```

Pull these PURGE cards if no previous level of RMFMANT exists in the system.

```
BINTAP(GFINT)
CATALOG(GFINT,GFINT,ID=IGS241,RP=999)
REWIND(BNG,BNB,GIN,GFINT)
LABEL(NPL25,W,L=241RMF2P1*3P4,I=999,D=HI)
UPDATE(P=CMN,F,N=NPL25,B)
REWIND(NPL25)
COPYBF(NPL25,Z)
COPYBF(BNG,NPL25)
COPYBF(BNB,NPL25)
COPYBF(GIN,NPL25)
COPYBF(GFINT,NPL25)
COPYBF(FILE6,NPL25)
COPYBF(FILE7,NPL25)
COPYBF(FILE8,NPL25)
UNLOAD(NPL25)
7/8/9
*/  MODIFICATIONS TO GRASS SHOULD FOLLOW THIS CARD.
*/
*/  MODIFICATIONS TO GRID RESIDENT SHOULD FOLLOW THIS CARD.
*/
*/  MODIFICATIONS TO BINTAP SHOULD FOLLOW THIS CARD.
*/
*/  MODIFICATIONS TO GINIT SHOULD FOLLOW THIS CARD.
*/
7/8/9
*COMPILE GRASS
7/8/9
*COMPILE GRIDRES
7/8/9
*COMPILE BINTAP
7/8/9
*COMPILE GINIT
7/8/9
REWIND(BNG)
REWIND(BNB)
READY(SYSTEM,OLD)
LIBRARY(IGS241,OLD)
REPLACE(GRASS,BNG,AL=1)
REPLACE(BINTAP,BNB,AL=1)
FINISH.
COMPLETE.
ENDRUN.
7/8/9
6/7/8/9
```

```
CMFBIN,CM60000,T2000,MT1.
COMMENT. INSTALL 241/IGS  FROM BINARIES)
LABEL(PL24,R,L=241CMF2P1*3P4,D=HI)
COPYBF(PL24,X)
UNLOAD(X)
COPYBF(PL24,BN1)
EDITLIB(SYSTEM)
COPYBF(PL24,Z,3)
UNLOAD(Z)
COMMENT.  PUNCH VERIFICATION DECK.
COPYBF(PL24,PUNCH)
7/8/9
READY(SYSTEM,OLD)
LIBRARY(IGS241,OLD)
REWIND(BN1)
ADD(*,BN1)
FINISH.
COMPLETE.
ENDRUN.
6/7/8/9
```

```
RMFBIN,CM60000,T2000,MT1.
COMMENT. INSTALL 241/IGS REMOTE ROUTINES.
COMMENT. FROM BINARIES.
COMMENT.  ASSUMES IGS241 LIBRARY EXISTS
COMMENT.  FROM INSTALLATION OF CM ROUTINES.
LABEL(PL25,R,L=241RMF2P1*3P4,D=HI)
COPYBF(PL25,X)
UNLOAD(X)
COPYBF(PL25,BNG)
COPYBF(PL25,BNB)
EDITLIB(SYSTEM)
REQUEST(GIN,*PF)
COPYBF(PL25,GIN)
CATALOG(GIN,GIN,ID=G41,RP=999)
REQUEST,GFINT,*PF.
COPYBF(PL25,GFINT)
CATALOG(GFINT,GFINT,ID=IGS241)
7/8/9
READY(SYSTEM,OLD)
LIBRARY(IGS241,OLD)
REWIND(BNG)
REWIND(BNB)
ADD(*,BNB,AL=1)
ADD(*,BNG,AL=1)
FINISH.
COMPLETE.
ENDRUN.
6/7/8/9
```

```
JOB3,CM50000,T500.
COMMENT.  MAKES A PERM FILE OF VRFY.
REQUEST(VRFY,*PF)
FTN(B=VRFY)
CATALOG(VRFY,VRFY,ID=CDC)
7/8/9
      PROGRAM VRFY
      DIMENSION IBUF(40),MESS(24),IPRT(4)
      DATA IALF/4LOKOK/
      DATA JALF/4LNONO/
      DATA(MESS(J),J=1,4)
     */10H* THIS IS  ,10HA VERIFICA,10HTION TEST ,10H*          /
      DATA(MESS(J),J=5,8)
     * /10H PICK ANY ,10HPART OF SQ ,10HUARE WITH ,10HLIGHT PEN   /
      DATA(MESS(J),J=9,12)
     * /10H THE PICK ,10HWILL BE QU,10HEUED AS A ,10HBUTTON      /
      DATA(MESS(J),J=13,16)
     * /10H IF PICK I,10HS GOOD THE ,10H SQUARE WI,10HLL VANISH   /
      DATA(MESS(J),J=17,20)
     *  /10H AND MESG.  ,10H OKOK WILL,10H BE DISPLA ,10HYED.       /
      DATA(MESS(J),J=21,24)
     *  /10H IF PICK I,10HS BAD, NON,10HO WILL APP,10HEAR.         /
      IDC=2$IDT=8$IDDT=5$IDDC=6$NCON=1$NBYTE=0
      CALL GICNJB(NCON)
      CALL GICLR(NCON)
      CALL GIMASK(NCON,33,8,8)
      JH=-400 $ JV=400 $ I=0 $ IEND=10H
       I=0
      DO 82 J=1,6
       DO 81 K=1,4
 81    IPRT(K)=MESS(K+I)
      CALL GURSET(JH,JV,0,IBUF,NBYTE,200)
       CALL GUAN(IPRT,40,IBUF,NBYTE,200)
      CALL GIDISP(1,IBUF,NBYTE,IDX)
      JV=JV-40
       I=I+4
82    CONTINUE
C   DRAW OUTSIDE SQUARE.
      CALL GURSET(-500,-500,1008,IBUF,NBYTE,200)
      CALL GUSEGS(-500,-500,-500,+500,1,0,IBUF,NBYTE,200)
      CALL GUSEG(+500,+500,1)
      CALL GUSEG(500,-500,1)
      CALL GUSEG(-500,-500,1)
      CALL GIDISP(1,IBUF,NBYTE,IDDAD,IDT,IDC)
   11 CONTINUE
      CALL GIBUT(0,1,LDT,LDC)
      IF(LDC .EQ. 2) GO TO 10
      CALL GURSET(-100,-100,0,IBUF,NBYTE,200)
      CALL GUAN(JALF,4,IBUF,NBYTE,200)
      CALL GIDISP(1,IBUF,NBYTE,JDDAD,IDDT,IDDC)
      GO TO 11
   10 IER=IDDAD
      CALL GIERAS(IER)
      CALL GURSET(-100,-100,0,IBUF,NBYTE,200)
      CALL GUAN(IALF,4,IBUF,NBYTE,200)
      CALL GIDISP(1,IBUF,NBYTE,JDDAD,IDDT,IDDC)
           CALL GIMOVE(0,0,4,JDDAD)
      CALL GICNRL(NCON)
      END
7/8/9
7/8/9
6/7/8/9
```

VERIFICATION PROCEDURE

When the installation of 6000/241/IGS Version 2.1 is complete, the EDITLIB of the IGS central routines will be complete, the files GIN and GFINT will be made permanent, the PASSWORD files entered, and INTERCOM enabled in the system.

Execution of job deck 4, file 6 of the release tape, PL24, will produce the file, VRFY, which will be used to verify the installation of 241 IGS.

Enter the BOOTSTRAP into the remote terminal memory using the Deadstart Initialization Procedure. After valid user name and password have been entered and accepted by INTERCOM, the word COMMAND will be displayed:

1. Type: FETCH,VRFY,CDC, press SEND key

   Response: COMMAND

2. Type: XEQ,LOAD=VRFY,SATISFY=IGS241,EXECUTE, press SEND key

   Response: Program VRFY will send:

```
"THIS IS A VERIFICATION TEST"

PICK ANY PART OF SQUARE WITH LIGHTPEN
THE PICK WILL BE QUEUED AS A BUTTON IF
PICK IS GOOD, THE SQUARE WILL VANISH
AND MESS.  OKOK WILL BE DISPLAYED.
IF PICK IS BAD, NONO WILL APPEAR.
```

3. Using lightpen, with LPCAPT light on, pick a portion of the square.

4. The sequence, as stated, should proceed. The display will then be cleared and control returned to Command Mode.

5. Type: LOGOUT., press SEND key

TERMINAL OPERATING PROCEDURES

Procedures for using the 241 Remote Terminal depend on the following circumstances:

1. Deadstart Initialization Procedure

   This procedure is required when it is not certain whether the resident program is intact. Partial destruction might be caused by preventive maintenance, emergency maintenance, a Power OFF condition, or a manual change in the resident program.

2. Normal Operating Procedure

   This procedure assumes that the resident program is intact. The resident program should be assumed to remain unchanged even though the previous user may have aborted his job.

DEADSTART INITIALIZATION PROCEDURE

1. Press MC (Master Clear) button.

2. Press LOAD ADDR button.

3. Press KBD button (to see what is entered via the keyboard-buttons 0 through 7).

4. Enter 7500 via keyboard (7500 should be visible in keyboard display).

5. Press STEP button <u>once</u>, press Bank 2 button.

6. Press WRITE MEM button.

7. Enter code as defined in the GRIDRES listing, beginning at Bank 2, location 7500. For example:

   The listing contains---

   ```
   2  7500   0062                 BOOT
   2  7501   2200     7101
   2  7503   4021
   ```

   Enter 0062 via keyboard, press STEP button (<u>once</u>)
   Enter 2200 via keyboard, press STEP button (<u>once</u>)
   Enter 7101 via keyboard, press STEP button (<u>once</u>)
   Enter 4021 via keyboard, press STEP button (<u>once</u>)

   The address to be entered can be verified by pressing P - button.

8. Press LOAD ADDR button.

9. Enter 7500 via keyboard.

10. Press STEP button, press Bank 2 button.

11. Press OPR MEM button.

12. Press PROC MODE button.

13. Press RUN button.

14. Press ON LINE button.

15. Dial 6000 computer via dataphone.

16. When connection is made, the numbers 4, 3, 2, 1 appear on the CRT. After the initialization program is established, the words LOADING GRID appear on the CRT.

17. When the total resident routine is loaded, the bootstrap will LOGOUT automatically. A cursor will appear on the upper left corner of the CRT indicating the user can LOGIN.

TABLE FORMATS

The table formats are intended to serve only as reference material for those who are familiar with SCOPE 3.4 and its product set; more detailed information is available in the various reference manuals and internal maintenance specifications.

Section 1 contains formats of tables that are part of CMR. Generally, they are of interest only to SCOPE and INTERCOM system programmers.

Section 2 contains formats of tables that can be used by central processor programs running at a control point. Section 2 is generally of interest to all users of the product set.

Section 3 contains formats of tables residing on mass storage devices. Generally, they are of interest only to system programmers.

Section 4 contains formats of tables and areas that reside in ECS. ECS formats are generally of interest only to SCOPE system programmers.

Unless reserved for a specific purpose or group, all currently unused fields, names, codes, etc. are reserved for future standard development.

3.4 TABLE FORMATS CONTENTS

SECTION 1

CENTRAL MEMORY RESIDENT TABLES

CENTRAL MEMORY RESIDENT

0		Pointers
100		Channel Status Table
154		PP Status Words
200	$T.CPA_n$	Control Point Areas
	T.XPIDLA	System Exchange Packages
	$T.PPC_n$	PP Communication Areas
*	T.EST	Equipment Status Table
*	T.FNT	File Name Table
		CIO-CPCIO Special FNTs
		Permanent File FNTs
*	T.ITABL	INTERCOM Table
*	T.DAT	Device Activity Table
*	T.STG	Tape Staging Table
*	T.APF	Attached Permanent File Table
‡	T.RQS	Request Stack
	T.RBR	Record Block Reservation Table (Headers)
	T.RBRBIT	RBR Bit Table
	T.DST	Device Status Table
	T.DPT	Device Pool Table
	T.SEQ	Sequencer Table
	T.RMS	Rotating Mass Storage Diagnostic Table
	T.INS	Installation Area
	T.VSNBUF	VSN Buffer
	T.TAPES	Tapes Table
	T.RPT	Removable Pack Table
	T.MAIL	Scheduler Mailbox Buffer
	T.DFB	Dayfile Buffers
	T.PJT	Parameter Storage for Delayed PP Jobs
	T.SCHPT	(Optional) Scheduler Statistics
	T.SCHJCA	Scheduler Job Control Area
	T.SCHJDT	Scheduler Job Descriptor Table
	T.BCFAP	CPMTR CEFAP Buffer
	T.EPAGE	Empty Page Stack
	T.ECSPRM	ECS Parameters
	T.SUBPG	Subpage Buffer
	T.ECTL	Description of T.EBUF Area
	T.EBUF	ECS Buffer for RMS–ECS Transfer
		CM Resident Programs
	T.LIB	Library Directory
		INTERCOM Pointer Area
		INTERCOM Small Buffers and User Tables
		INTERCOM Large Buffers

*Table Must Begin Before 10000_8

‡Table Must Begin Before 20000_8

CMR POINTER AREA

	59	47	35	23	11	0
P.ZERO	Zeros					0
P.LIB	a C.DIRFWA FWA of Library Directory			LWA+1 Library Directory	C.DSFLAG Deadstart Load Flag	1
P.RBR P.RBT	C.RBRAD FWA of RBR Area		RBT Ordinal of Empty Chain	Length/100B of RBT Area	C.CMLWA (LWA+1)/100B of CM	2
P.NPP P.NCP P.DFB	FWA/8 of Dayfile Buffer	(Reserved for 250 Graphics Package)		C.NPP No. of PPs	C.NCP No. of CPs	3
P.SEQ P.FNT P.HEC	FWA of FNT	LWA+1 of FNT	C.SEQ T.SEQ/8	C.SEQL L.SEQ	C.HEC Hardware Error Count	4
P.CST P.PCOM P.EST	FWA of EST	LWA+1 of EST	C.CST FWA of CST	C.CSTL LWA+1 of CST	C.PCOM Address of Comm Area PP1	5
P.PFM1	C.SDTL N.SD	C.APFL N.EAPF	C.PFACT Activity Count	C.APF T.APF	C.PFMCH Toggle Byte	6
P.PFM2	C.SDL C.ESD					7
P.INS	(Reserved for Installations)					10
P.EIRPR	C.LEPAGE L.ECSTK+1		C.ECSPRM T.ECSPRM	ICC Area Address		11
P.ELBST	Max. Length/1000B of ECS Library File	ECS Flaw Table Address		ECS Page Stack Address		12
P.RQS	T.DAT	L.DAT	C.RQSFS FWA/2 of Request Stack	No. of DST Entries	FWA/8 of DST	13
P.DPT P.TAPES P.RMS	T.TAPES/8	L.TAPES	C.RMS T.RMS/8	C.RMSL L.RMS	C.DPT T.DPT/8	14
P.STG					C.STG T.STG	15
P.INT	C.INT/C.IFL Control Point 0 FL	C.ITABL	C.IBUFF Start of INTERCOM pointer area		C.ILTABL	16
P.PFM3	C.RBTC1	C.RBTC2 RBTC Pointer	C.RBTC3	C.RBTCL N.RBTC	C.PFFNT FWA of PF FNTs	17

P.LIB(1)

	a		Library change bit
	C.DSFLAG		
Bit	0	S.SYSEDT	1=bypass EDITLIB "GO/DROP" message (internal to deadstart)
	1-2	S.EDTRUN	01=EDITLIB running
	3-5	S.ECSLVL	ECS level
			000=no ECS
			001=ECS up
	6	S.ACTION	0=load
			1=recovery
	7-8	S.SYSLVL	System level
			00=A
			01=B
			10=C
			11=D
	9	S.PFLVL	PF level
			0=initiallize
			1=check
	10-11	S.LBLLVL	RMS label level
			00=initiallize
			01=check

P.PFM1(6) **C.PFMCH**

Bit	0	S.RBTCW	RBTC wraparound
	1	S.APFIL	APF interlock
	2	S.PFDIL	PFD interlock
	3	S.RBTCIL	RBTC interlock
	4	S.PFUTIL	Utility interlock
	5	S.MDIL	MD interlock
	6		TRANSPF lockout

CMR POINTER AREA

Bit positions: 59, 47, 41, 35, 23, 11, 0

Symbol	Contents (bits 59 → 0)	Word
T.JDATE	(Leading Zeros) Y Y D D D	20
P.NRBR	Number of Request Stack Entries (N.RQS) | Number of RBR Headers (N.RBR) | (blank) | Size of Total RBR Area (L.RBR)	21
T.BJDT	Julian Date in Binary (YYYDDD) | (blank) | Time in Binary (HHMMSS)	22
P.EVICT	(blank) | Trace Buffer T.TRB/8	23
P.CMFL	(blank) | Machine FL/100B	24
	∧ S Y S T E M ∧ ∧ ∧	25
T.CPJOBN, P.PJT, P.SPDROP	Job Sequence Number | C.SPDROP DST Ordinal for 1SP Drop | Job Count | C.PJTFWA T.PJT/8 | C.PJTLWA T.PJT/8+L.PJT/8	26
T.EPBL, P.ECSFL	C.ECSPL ECS Page Length | C.ECSBL ECS Buffer Length | C.CPECFL Direct Access ECS FL/1000B	27
T.CLK	H H M M S S	30
T.SLAB1, T.DATE	M M / D D / Y Y	31
T.SLAB2	System Label SCOPE Version 3.4	32
		33
		34
		35
T.SLAB6		36
T.MSP	(blank) | Debugger | Step Flag	37

CMR POINTER AREA

	59	47	35	23	11 … 0	
T.MSC	Count of PP Job Queue Entries	Number of Idle PPs	Number of Seconds*4096			40
P.CHRQ				C.CHRQ First 10 Channels	C.CHRQ2 Second 10 Channels	41
P.PPLIB	Position of CIO	0 0 0 0	Number of Programs		Address of First Entry	42
P.VRNBUF	C.VRNFWA T.VRNBUF/8	C.VRNFIN Pointer to First VSN	C.STGFLG Stage ON/OFF	C.VRNINT Buffer Interlock	C.VRNFUL Buffer Full Flag	43
T.CPSTA	Idle Exchange Package Address	* * / * * * * * *	Next Slice Time	2 0 / 0 3 0 3 * * * * / 0 0 0 L * * * *	Active XP Address	44
T.CPSTB						45
T.MXNCTL	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 P / STL Code		20	Active XP Address / * * * * * *	2 6 1 P / 2 6 2 P	46
T.PPID	CP-MTR Requests				PP Input Register Address	47
T.PPIP	PP-MTR Requests				PP Input Register Address	50
	(Reserved)					51
	(Reserved)					52
T.SPF	Control Point Number				EST Ordinal	53
	(Reserved)					54
T.RCHN	SPM-1RN Communications Word				First RBT Word Pair to Release	55
T.CPT1 / T.UAS	Unassigned CM/100B	Unassigned ECS/1000B	ECS Size		Initial CMTR P Address	56
T.ECSPAR / P.EPAGE		C.EPAGE T.EPAGE	ECS Flaw Table Flag	ECS Parity Flag	ECS Parity Address/1000B	57

Rescheduling, OFF (word 44); EXN, MXN, MAN (word 46)

L = { 0 Turned Off / 1 Locked Off }   P = { 0 CPUA / 1 CPUB }

CMR POINTER AREA

	59	47	35	23	11 0	
P.SCH	C.SRSL C.LEJDT LE.JDT	C.SRS T.XPSCH/10B	C.JCA T.SCHJCA/8	C.LJDT L.SCHJDT	C.JDT T.SCHJDT/8	60
P.STR	C.NFL Needed FL/100B	C.JQP Queue Priority of Job in Counter	C.RFL Reserved FL/100B	C.STMF SCH Recall	C.AFL Available FL/100B	61
T.SCHCP	Interlock Word (Scheduler)					62
T.SCHPP	Interlock Word (PP Routines)					63
P.RPT	T.RPT/8	L.RPT				64
P.MAIL P.SWPECS P.SCHPT	C.MAILF T.MAIL/8	C.MAILL L.MAIL	C.SWPECS L.ECSSWP	C.SCHPT T.SCHPT/8		65
	(Reserved)					66
	...					
	(Reserved)					76
P.ILR P.PPOVL			C.ILR IP.ILR	C.PPOVL T.PPOVL/10B	T.ELIBD	77

IP.ILR:
0 = No ILR
1 = 64-bit ILR
2 = 128-bit ILR

P.SWPECS(65) C.SWPECS

The bits in C.SWPECS determine which jobs are swapped to ECS. If no bits are set, only INTERCOM and graphics jobs swapped to the central memory queue and MUJ jobs are swapped to ECS. Additional job types are swapped to ECS as follows:

Bit 0	1 = INTERCOM and graphics jobs at end of command (EOJ bit set)
1	1 = Batch jobs in the central memory queue
2	1 = All INTERCOM and graphics jobs
3	1 = All batch jobs

CHANNEL STATUS TABLE

48	C.CSTCN 36	C.CSTCB 26	24	12	0
0	Channel Number		C	X	Y

C MMTC Conversion Table
00 — No Table
01 — EBCDIC
10 — ASCII
11 — Reserved

X Address of this word

Y Same as X when channel not reserved
PPIR address when reserved

CHANNELS

00-13	Hardware channels
14	CH.FST
15	CH.FNT/CH.ILR
16	CH.LIB (GRAPHICS software only)
17	CH.RBT
20-33	Hardware channels
34	CH.CPA
35	CH.PFM
36	CH.INS
37	CH.DMP
40	CH.EST = CH.TAPE
41	CH.ICOM INTERCOM/SCOPE communication interlock
42	CH.IEMBF INTERCOM empty buffer channel
43	CH.IUSER INTERCOM user table channel
44	CH.SCH scheduler channel
45	CH.IHUSR High speed user table channel
46	CH.IHSMT High speed empty buffer channel

PP STATUS WORDS

	59	47	35	23	0	
T.PPS1	Control Point Address	PP Status	Field Access Flag	Channel I/O Time		154_8
			. . .			177_8

EXCHANGE PACKAGE

0		P	A0	B0	0
1		CMRA	A1	B1	1
2		CMFL	A2	B2	2
3		EM	A3	B3	3
4	ECS RA		A4	B4	4
5	ECS FL		A5	B5	5
6	MA		A6	B6	6
7			A7	B7	7
10	X0				8
	. . .				
17	X7				15

CONTROL POINT AREA

Bit positions: 59, 47, 44, 41, 35, 29, 23, 17, 14, 11, 5, 0

Word Symbol						Word
	Exchange Package					0–17
W.CPUST W.CPLINK	C.CPSTAT Status Byte	C.CPSLIC M.RCLCP Time or Slice Period	* *	C.CPLINK Previous Active Control Point	Next Active Control Point	20
W.CPTIME	C.CPUQS CPU-A	C.CPUQMS Seconds*4096 This Quantum	C.CPUAS Total CPU-A Time as Number of Seconds*4096			21
W.CPTIMB	C.CPUQS CPU-B	C.CPUQMS Seconds*4096 This Quantum	C.CPUBS Total CPU-B Time as Number of Seconds*4096			22
W.PPTIME W.CPPTM	C.CPPQS PP	C.CPPQMS Seconds*4096 This Quantum	C.CPPTS Total PP Time as Number of Seconds*4096			23
W.CPSTAT W.CPFL W.CPEF		C.CPEF Error Flag	C.CPSM Storage Move	C.CPRA RA/100B	C.CPFL FL/100B	24
W.CPJNAM	C.CPJNAM Job Name				JDT Ordinal	25
W.CPCC	C.CPRPV Reprieve CKSM Value	C.CPRPA Reprieve Address		C.CPNFL Nominal $FL/100_8$	C.CPNCSP Next Control Card Pointer	26
W.CPECS				C.CPECRA ECS RA/1000B	C.CPECFL ECS FL/1000B	27
W.CPDFM	Last Dayfile Message					30–37
W.CPPRI W.CPJCP W.CPTIML	C.CPTIML Current Time Limit (15 Bits)	C.CPTLI Initial Time Limit (15 Bits)	C.CPRPRI Job Class	C.CPECSI Initial ECS FL/1000B	C.CPFLI Initial FL/100B	40
W.CPSWP W.CPINT	C.CPQNT Quantum	C.CPUTA User Table Address			C.CPORG C.CPEVNT Flags / Job Origin	41
W.CPSCH W.CPRO	C.CPFLG Swap Flags	C.CPJQP Job Queue Priority	C.CPRFL Reserved FL		C.CPJDA JDT Address (Absolute)	42
W.SSW W.CPSSW				C.CPSSW Sense Switches		43
	(Reserved)					44
	(Reserved)					45
	(Reserved)					46
	(Reserved)					47

CONTROL POINT AREA

Bit positions: 59, 47, 42, 35, 23, 17, 11, 5

Symbol	Contents	Word
W.CPFACT	Account Parameter for Permanent Files	50
W.CPFST	FST Entry for Next Control Card PRU	51
W.CKP, W.CPCKP	(bits 59–18: blank) ; C.CPCON Console Checkpoint Flag ; C.CPCKP Number of Checkpoints	52
W.CPOAE	C.CPREQ Req Flag ; (1 = Extended Label Format) ; Relative Address of Tape Label Info ; (blank) ; C.CPOAE Equipment Assigned	53
W.CPVRNO	Family Pack VID Assignment	54
W.CPLDR1	C.CPLW C.CPLT Loader Flags ; (blank) ; Global Library	55
W.CPLDR2, W.CPLS	Set	56
W.CPLDR3	Indicators	57
W.CPAR	RA+1 Contents (and Control Point Number) of Last Auto-recall Request ; C.CPAR Reply Word Address	60
W.CPSTG	C.CPTMT MT ; C.CPTNT NT ; (blank) ; C.CPMNT MT Max / NT Max	61
W.CPDFMC, W.CPDPV, W.CPIRB	C.CPDFMC Dayfile Msg Count ; (blank) ; C.CPRBID INTERCOM Batch Routing ID ; (blank) ; C.CPDPV Job Dep. ID	62
W.CPFP, W.CPOUT, W.CPFLAG, W.CPERT	C.CPFLAG Flags ; (blank) ; C.CPFST FST Address ; (blank) ; C.CPFP C.CPOUT Flags	63
W.CPMSLM	C.CPMSLO MS Limit Saved During Swap ; C.CPMSLM MS Limit in PRU's ; C.CPMSRC Running PRU Count	64
W.CHTIM	(blank) ; Channel Time as Number of Seconds*4096	65
W.CPMSI	C.CPSITM Time of Swap-In ; (blank)	66
W.CPSR	(blank) ; C.CPSR ; (blank) ; C.CPESR (Non-zero During ECS-Disc Transfer)	67
W.CPCAF	Control Card Buffer	70 – 167
W.CPCAL	Reserved for Installations	170 – 177

NOTES: CONTROL POINT AREA

W.CPLINK(20) **C.CPSTAT**

Bit	0	S.CPUSTM	Move flag - move in progress
	1	S.CPUSTY	Auto recall
	2	S.CPUSTA	CPUA assigned only
	3	S.CPUSTB	CPUB assigned only
	4	S.CPUSTX	Recall status
	5	S.CPUSTW	Wait status
	6	S.CPUSTR	Real time job
	7	S.CPUSTC	Active CPUA
	8	S.CPUSTD	Active CPUB
	9	S.CPUSTS	Control point activity suspended
	10	S.CPUSTP	Suspended by check point

W.CPEF(24) C.CPEF Values:

0001	F.ERTL	Time limit exceeded
0002	F.ERAR	Arithmetic error
0003	F.ERPP	PPU abort (M.ABORT)
0004	F.ERCP	CPU abort (ABT in RA+1)
0005	F.ERPCE	PP call error (garbage in RA+1)
0006	F.EROD	Operator drop
	F.IUABT	INTERCOM user abort
0007	F.ERK	Operator kill (batch job only)
0010	F.ERRN	Rerun (batch job only)
0011	F.EREX	Control card error
	F.ERCC	
0012	F.ERECP	ECS parity error
0013	F.ERJC	Job card error
	F.ICCD	
	F.IJBCRD	
0014	F.ERPA	Pre-abort (batch job only)
0015	F.ERRCL	Auto-recall error
0016	F.ERHANG	Job hung in auto-recall
0017	F.ERMSL	Mass storage limit exceeded (batch job only)
0020	F.EROVL	PP overlay not in PP LIB

W.CPSWP(41) C.CPORG Values:

Value	4	Real time
	10	Graphics
	20	Multi-user
	40	INTERCOM
Bit	6	Swap out event bit

NOTES: CONTROL POINT AREA (CONT'D)

W.CPSCH(42) C.CPFLG Values:

Bit	3	S.CP1IB	1IB bit
	4	S.CPFFL	FNTs in positive FL
	5	S.CPEOJ	End of job
	6	S.CPCLR	Control point area clear request
	7	S.CPRFL	Storage request
	8	S.CPROP	Roll out
	9	S.CPSIP	Swap in
	10	S.CPSOP	Swap out
	11	S.CPSWC	Swap out complete

W.CPCKP(52) **C.CPCON**

Bit	0	Console checkpoint request

W.CPLDR1(55) **C.CPLW**

Bit	1	S.CPLP	Program loaded from non-system library
	2-3	L	Library set indicator
	4	S.CPLT	Debugging aid flag
	5	R	Reduce flag
	6-9	M	Map Options
	10-11	W	Indicator for loader to be used

W.CPLDR1(55)
W.CPLDR2(56)
W.CPLDR3(57) Global Library Set Indicators:

00	End of global library set
01-76	LNT ordinal of system library
77	User library; lfn of first user library in W.CPLDR3; lfn of second user library

W.CPFLAG(63) **C.CPFLAG**

Bit	0	S.CPLDAF	MDI interlock
	2		Private pack overflow
	3	S.CPNFNT	If on, do not search FNT

W.CPFP(63) **C.CPFP**

Bit	0	S.CPL	Reprocess
	1	S.CPG	Abort
	2	S.CPA	No return
	3	S.CPS	Sequencer
	4	S.CPN	Checkpoint taken
	5	S.CPX	EXIT card encountered
	6	S.CPDP	Private disk pack
	7	S.CPEOR	Control card EOR
	8	S.CPJFL	Job card field length assigned
	9	S.CPJ	JANUS
	10	S.CPR	Remote batch
	11	S.CPE	INTERCOM

SYSTEM JOB EXCHANGE PACKAGE AREA

	59	47	35	29 23	14 11 0
T.XPSCH	Exchange Package for Scheduler and Storage Move				
W.CPUST/W.CPSLIC	C.CPSTAT Status Byte	C.CPSLIC Total CPU Time Slice Alotted		C.CPLINK Previous Active Control Point	Next Active Control Point
W.CPTIME	CPU-A Scheduler/Storage Move Time				
W.CPTIMB	CPU-B Scheduler/Storage Move Time				
W.PPTIME	PPU Idle Time				
W.XPIR	Scheduler Input Register				
W.XPOR	Scheduler Output Register				
Scheduler Only T.SCHRS	Scheduler Request Stack Table				L.SCHRS=4
T.MTRRS	CPMTR MTR Request Stack				L.MTRRS=4
T.XPIDLA	CPU-A Idle Program Exchange Package P=CP.MTR (If IP.ECSB≠0, FL=400000B ECS FL=17777777B B2=1 B5=L.ECSTK B7=0 A6=T.EPAGE)				
	Status of Idle Ex. Pkg.	Accumulated Slice Time		Exchange Package Links	
W.CPTIME	CPU-A Idle Time				
W.XPIR	I D L E 1				
T.XPIDLB	(If IP.MCPU=2) CPU-B Idle Program Exchange Package P=CP.MTR (If IP.ECSB≠0, FL=400000B ECS FL=17777777B B2=1 B5=L.ECSTK B7=1 A6=T.EPAGE)				
	Status of Idle Ex. Pkg.	Accumulated Slice Time		Exchange Package Links	
W.CPTIME	CPU-B Idle Time				
W.XPIR	I D L E 2				

PP COMMUNICATION AREA

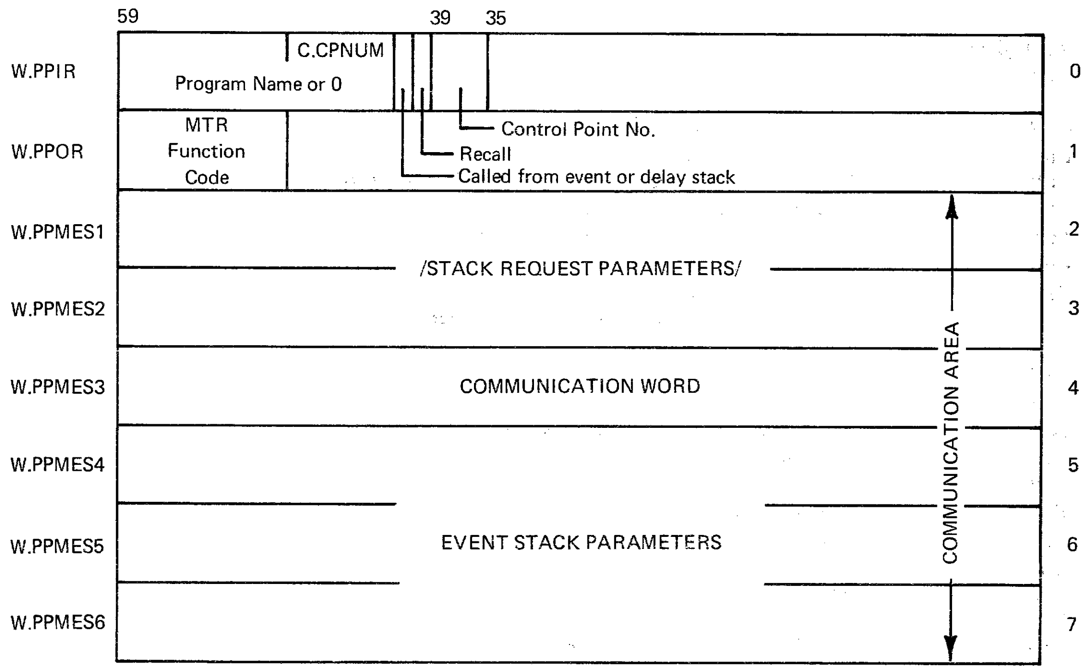
FOR PP0

Word	59	17	0	
W.PPIR	CPCIO Output Register			
W.PPOR	MTR Output Register			
W.PPMES1	Stack Request Parameters			
W.PPMES2				
W.PPMES3	7HDAYFILE	000030		Extend for System Dayfile
W.PPMES4	0			
W.PPMES5	7HCERFILE	000030		Extend for CERFILE
W.PPMES6	0			

FOR PP1

Word		
W.PPIR	DSD Input Register	
W.PPOR	DSD Output Register	
W.PPMES1	Stack Request Parameters	COMMUNICATION AREA
W.PPMES2		
W.PPMES3	Control Point 0	
W.PPMES4	Message Buffer	
W.PPMES5		
W.PPMES6	DSD – 1DL Communication Word	

PP COMMUNICATION AREA
FOR PP2 THROUGH PPn



COMMUNICATION WORD

59	47	35	23	17	11	0
C.RWPPCF Control Point	C.RWPPWT Cumulative Byte Count	C.RWPPLW PP Buffer Length	C.RWPPCC C.RWPPST Code and Status	C.RWPPWC Current PRU Byte Count

PP PROGRAM NAME RESERVATIONS

Routine Name	Description
AOO (010000)	Stack Processor segment
ACE	Advance control card
CCP	6000 station routine
CEM	Central error manager for ECS
CIO	Preliminary I/O request processor
CKP	Saves information necessary to restart a checkpoint job
CLO	Dummy program used to call CIO
CON	INTERCOM-connect file to remote terminal
CP1	C.E.-415 card punch test
CR1	C.E.-405 card reader test
CY1	Resets FNT of file being processed by restart
DF4	C.E.-3234 test
DF7	C.E.-3553 test
DF8	C.E.-808 test
DIS	Console display program for a control point
DLE	C.E. Diagnostics
DMP	Dump CM
DPF	Dump permanent files to tape
DSP	Dispose function processor
EKG	Private pack closing-1EJ
EPF	Send audit information to CM
FAD	INTERCOM
FNT	INTERCOM-FNT alter routine
GBJ	INTERCOM-274 Graphics begin job
GEJ	INTERCOM-274 Graphics end job
IAP	INTERCOM-initiate another program
IEF	Routine for CEFAP
IUP	INTERCOM-initiate user program
JDP	Job dependency count decrementor
LDL	Loader utility program
LDV	Loads CPU absolute overlays
LDW	Loads CPU absolute overlays in conjunction with LDV
LOC	Load octal corrections
LPF	In conjunction with LOADPF, reloads permanent files
LPT	C.E.-501 line printer test
LP1	C.E.-512 line printer test
MAC	INTERCOM
MDI	Used by EDITLIB to handle I/O involved in changing and moving directory
MEM	Process memory function
MES	INTERCOM-writes messages to remote terminal
MSD	INTERCOM
MSG	Issues dayfile messages
MTT	C.E.-60X tape test
MVJ	INTERCOM-Multi-User Job
OPE	Dummy program used to call CIO
PFA	Permanent file manager attach function
PFC	Permanent file manager catalog function

PP PROGRAM NAME RESERVATIONS (CONT'D)

Routine Name	Description
PFD	Attaches permanent file directory to control point
PFE	Permanent file manager extend function
PFP	Permanent file manager purge function
PFR	Permanent file manager rename function
PFS	Permanent file manager position function
PPI	Reserved
PRM	Permission checking function
QAJ	Reserved
REQ	Makes non-allocatable device assignment and formats FNT entries for allocatable devices in response to request control card or a request macro call.
RMS	Routine for CERMS
RPV	Reprieve central program
RST	Restores control point area of restart job
RWE	INTERCOM-checks for INTERCOM job
SLT	Reserved
SRB	Used by EDITLIB to complete the disk address of a record
STS	Used by CP program to obtain certain status
TBL	INTERCOM-Get table
TDS	Terminate deadstart
TPF	Transfer permanent files and permanent file table
TPT	Transfer permanent file tables
T76	INTERCOM
Uxx	Reserved for installations
VSM	STIMULATOR routine
VSN	Volume serial number card processor
XDQ	PP portion of dump queue
XRO	PP portion of restore queue
nUx	Reserved for installations
0ZA-0Z9	PL10 drivers
1AJ	Advance job
1BR	INTERCOM-buffer manager
1BT	Blank label tape routine
1CI	INTERCOM-Queue Manager
1CL	Close function for all non-tape or non-permanent files
1CR	Tape read recovery – write CM for 9-track tapes
1CS	Tape read recovery – write CM for S tapes
1CT	Tape read recovery – write CM for SCOPE tapes
1C9	Write CM for tape read recovery
1DA	Process private packs
1DF	Dump dayfile
1DL	Overlay loader and dayfile message processor for DSD
1DM	Device queue manager
1DS	INTERCOM – H-display
1DU	Used in conjunction with DPF to clear dump flags and directory entries
1EJ	End of job processor
1EP	Twin stack processor for processing system double buffer stack requests for ECS buffering

PP PROGRAM NAME RESERVATIONS (CONT'D)

Routine Name	Description
1FC	Creates an RBTC entry for PF catalog
1GJ	INTERCOM
1GM	Issues GOOD MORNING when time changes from 23.59 to 00.00
1GR	INTERCOM
1IB	Initiate batch job from input queue
1IS	INTERCOM–Send dayfile message to terminal
1IM	INTERCOM–Send message to terminal
1IQ	Initiate JANUS control point
1IR	Main JANUS routine; drives readers, punches, printers, etc.
1lS	Initialize overlay setup
1IU	Called by JANUS to backspace print file
1I1	INTERCOM–Initialization
1LT	Loads jobs from tapes
1LX	INTERCOM
1MF	Multifile positioning routine
1MH	Tape scheduling/prescheduling routine
1MT	Long record stranger tape driver
1NO	Tape read recovery noise record verifier
1NR	9 track tape read driver
1NW	9 track tape write driver
1N2	Tape read recovery noise record read forward 1
1N3	Tape read recovery noise record read forward 2
1OP	File open routine for non-tape files
1PC	Drop permanent file mass storage
1PD	Called by PFA to either enter event stack, call another PP routine or swap out
1PF	Permanent file queue manager
1PJ	INTERCOM – Process job card
1PK	Sequential disk pack close
1PL	Dummy plot program
1PS	6000 Station routine
1PT	INTERCOM–Low speed remote batch processor
1P1	Tape recovery to LGR positioning driver
1P2	Tape recovery write driver
1P3	Tape recovery verification driver
1P4	Tape recovery to LGNR positioning driver
1QM	INTERCOM–Check for MUJ swap-out completion
1QP	INTERCOM-Quantum calculator and MUJ servicer
1RC	Restores field length of a checkpointed job
1RN	Ages queues, manages RBT chains and statuses tape drives
1RP	End of reel processor
1RQ	REQ overlay
1RS	Read stranger tape driver
1RT	Read SCOPE tape driver
1RV	Tape I/O read recovery driver initializer and terminator
1R2	Tape read recovery – tape parity error recovery 1

Routine Name	Description
1R3	Tape read recovery – tape parity error recovery 2
1R9	SCOPE tape 9 track (659) read tape driver
1SI	Routine to swap-in or roll-in a job
1SO	Swap-out or roll-out a job
1SP	Mass storage I/O processor (stack processor)
1SX	Error message and abort function for stack processors
1S5	Load and execute 1SP or 3DO at second entry
1TD	Dump output files to tape
1TF	Tape forward motion routine
1TO	Tape open routine
1TS	Tape sampler
1VG	STIMULATOR routine
1WB	INTERCOM–Wideband driver
1WI	SCOPE internal tape write driver
1WS	Stranger tape write driver
1W9	SCOPE tape 9 track (659) write tape driver
1XG	INTERCOM–1XP overlay used for graphics
1XP-6XP	INTERCOM–High speed EXPORT processor
1ZA-1Z9	INTERCOM drivers
2GJ	INTERCOM
21S	Reservoir of routines for 1IS
2LP	3256/3659 driver for an on-line print file
2ME	INTERCOM–Message sending routine
2PC	3446 card punch driver
2RC	3447 card reader driver
2RP	Overlay to 1RP-End-of-reel processor
2TA	Tape assignment overlay
2TB	All backward tape motion
2TC	Extended trailer label group processor
2TJ	Translate job card
3CF	INTERCOM – Overlay to lCI
3CI	INTERCOM – Overlay to 1CI
3CT	INTERCOM – Overlay to 1CI
3CU	INTERCOM – Overlay to 1CI
3CX	INTERCOM – Overlay to 1CI
3DO	Initialize allocatable device file
3EP	ECS version of 6603-I disk driver
3EQ	ECS version of 6638 disk driver
3ER	ECS version of 865 drum driver
3ES	ECS version of 854 disk packs driver
3ET	ECS version of 6603-II driver
3EU	ECS version of 814 disk driver
3EV	ECS version of 821 disk driver
3EW	ECS version of 841 MDD driver
3LX	INTERCOM–Overlay to 1LX
3ME	INTERCOM–Overlay to 2ME
3PK	User-pack initialization

PP PROGRAM NAME RESERVATIONS (CONT'D)

Routine Name	Description
3PM	Segment of 1P1 used for holding code for future use
3PO	Segment of 1P3 that processes uncorrectable parity error GO or RECHECK code
3PS	Segment of 1P4 used for holding code for future use
3RQ	REQ overlay containing 2TACOM
3SP	Driver for 6603-I disk
3SQ	Driver for 6638 disk
3SR	Driver for 865 drum
3SS	Driver for 854 disk packs
3ST	Driver for 6603-II disk
3SU	Driver for 814 disk
3SV	Driver for 821 disk
3SW	Driver for 841 MDD
3SY	844 driver
3TT	INTERCOM–Transmit data from CPU to terminal
3T1-3T2	INTERCOM–Overlays to 3TT
4ES	Enter stack request
4LB	ANSI standard label processor
4LC	3000 label processor
4LX	INTERCOM–Overlay to 1LX
5DA	Initiate or destroy file on private pack
6BR	ANSI label processor read function code overlay
6BW	4LB overlay
6CR	3000 label processor read function code overlay
6CW	3000 label processor write function code overlay
6LC	Segment of 4LB or 4LC to load conversion table into MMTC
6LM	Segment of 4LB used to construct tape label messages
6L1	4LB inlay to convert PRU count
6L2	BCD conversion table inlay for 4LB
6L3	4LB inlay to check that proper conversion table is in the MMTC
6L4	4LB inlay for debug message writer
6L5	4LB inlay to format the label information
6L7	4LB inlay to pack and write label to tapes table
6MD	Dummy EDITLIB overlay
6NO	Tape error recovery debug segment assembled to give more detail about segment being read by 1NO
6PA	Prints system bulletin before header
6SI	Process Swap-in parity errors
6WM	Outputs dayfile error messages for I/O requests
7EC	Generate ECS buffers
7T1	ASCII/Display code conversion table
7T2	EBCDIC/Display code conversion table
7W1-7W2	Overlay for 6WM
8AA-8A9	Reserved
8BA-8B9	Reserved
8CA-8C9	C.E.–Reserved names
8DA	A display overlay for DSD (dayfile buffers)
8DB	B display overlay for DSD (control point status)

PP PROGRAM NAME RESERVATIONS (CONT'D)

Routine Name	Description
8DC	C display overlay for DSD (central memory)
8DE	E display overlay for DSD (equipment status table)
8DF	F display overlay for DSD (file name table)
8DH	H display overlay for DSD (I/O queues)
8DK	K display overlay for DSD (pointers and control point area)
8DL	L display overlay for DSD (central programmable)
8DM	M display overlay for DSD (PP communications area)
8DO	O display overlay for DSD (operator message)
8DP	P display overlay for DSD (tapes table and VSN previewing)
8DQ	Q display overlay for DSD (INTERCOM status)
8DR	R display overlay for DSD (JDT tables and queues)
8DS	S display overlay for DSD (job control area)
8DX	X display overlay for DSD (ECS memory)
8DY	Y display overlay for DSD (command format dictionary)
8DZ	Z display overlay for DSD (display dictionary)
8D1-8D9	DSD
8EA-8E9	DSD (7000 Station Display)
8FA-8PS	Reserved
8GO	Loaded by 1R3 when GO or DROP operator decision necessary during tape processing
8NO	Segment to 1N3 that writes debug messages to dayfile if IP.DBUG=1
8PT	INTERCOM – Overlay to 1PT
8PU-8W9	Reserved
8T3	Overlay to load MMTC memory
8XA	Channel commands overlay for DSD
8XB	Debugging commands overlay for DSD
8XC	PPU calling control points requests commands overlay for DSD
8XD	Equipment status commands overlay for DSD
8XE	Control point commands overlay for DSD
8XF	Deadstart commands overlay for DSD
8XG	Priority and tape staging job control commands overlay for DSD
8XH	INTERCOM commands for DSD
8XI	Miscellaneous commands overlay for DSD
8XJ	Miscellaneous commands overlay for DSD
8XK	Tape scheduling commands overlay for DSD
8XL	Operator action manager commands overlay for DSD
8XM	Error flag commands overlay for DSD
8XN	CP-PP interlock commands overlay for DSD
8XO	Initiate system jobs command overlay for DSD
8XP	Tape assignment command overlay for DSD
8XQ	Bring up displays command overlay for DSD
8XR	Divert a file command overlay for DSD
8X1-8X9	DSD
8YA-8Y9	DSD (7000 Station Commands)
8ZA-8Z9	INTERCOM PP drivers
9AA-9PS	Customer Engineering
9PT	INTERCOM
9PU-9Y9	Customer Engineering
9ZA-9Z9	INTERCOM

MONITOR FUNCTIONS

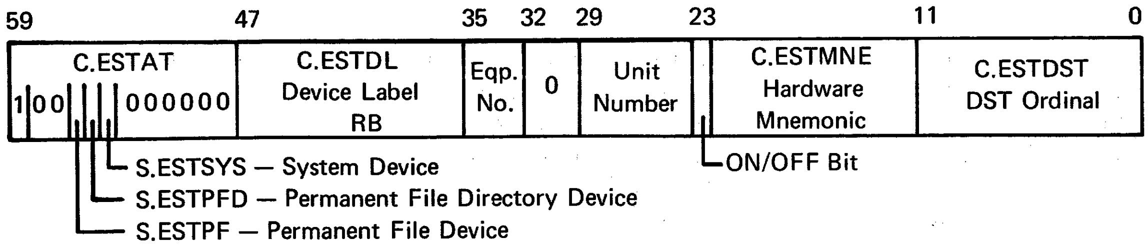
01	M.SETST		Set CPU status bits
02	M.CLRST		Clear CPU status bits
03	M.RCP		Request central processor
04	M.DCP		Drop central processor
05	M.RCLCP		Recall central processor
06	M.ICE		Initiate central executive
	00	EX.CMSM	CM storage move
	01	EX.ECSM	ECS storage move
	02	EX.ECOVL	ECS overlay load
	03	EX.SPM	Call stack processor manager
	05	EX.SCH	Call scheduler
	06	EX.SCH1	Call scheduler (storage request entry)
	07	EX.REQEB	Request ECS buffer
	10	EX.RELEB	Release ECS buffer
	11	EX.REQSB	Request system buffer
	12	EX.RELSB	Release system buffer
	13	EX.MVIN	Move data to ECS from system buffer
	14	EX.MVOUT	Move data from ECS to system buffer
	15	EX.FLHB	Flush buffer
	16	EX.CSWAP	Clean ECS after ECS RPE in swap file
	17	EX.AUTEB	Terminate automatic allocation
	20	EX.ECD	Display ECS
	21	EX.ECR	Release display
	22	EX.ECW	Modify ECS
	23	EX.CEM	Clear CEM-working flag
	24	EX.DDPER	Process DDP overlay loading error
	25	EX.ECLDV	Make successive partial reads of ECS record
07	M.CPUST		Change CPU status (IP.MCPU $\neq$ 1)
10	M.SLICE		MTR interrupts CPMTR at end of time slice for job
12	M.RCH		Reserve channel
13	M.DFM		Process dayfile message
15	M.STEP		Enter step mode
16	M.RBTSTO		Request RBT storage
17	M.RSTOR		Request storage
20	M.TSR		Terminate storage request (IP.RTMTR $\neq$ 0)
21	M.DPP		Drop PP
22	M.ABORT		Abort control point and drop PP
25	M.SEQ		Assign job sequence number
26	M.SEF		Set error flag
27	M.ISP		Initiate stack processor
30	M.SPRCL		Stack processor recall
31	M.CCPA		Change control point assignment
32	M.RPJ		Request peripheral job
33	M.EES		Enter event stack
34	M.CPJ		Capture peripheral job
35	M.SCH		Initiate integrated scheduler
36	M.PASS		MTR ignores it -- to be cleared by another routine
37	M.RACT		Request control point activity
41	M.NTIME		Enter new time limit

MONITOR FUNCTIONS (CONT'D)

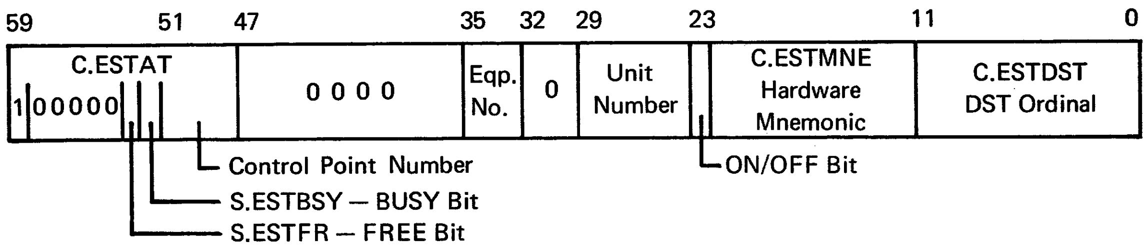
42	M.NOTE	Null function, cleared immediately. Used as break point
43	M.PPCH	Request channel surveillance
44	M.BUFPTR	Buffer pointer address
45	M.PATCH	Enter a patch into MTR
46	M.TRACE	Turn on MTR trace
47	M.SLPER	XJ to other CPU
77	M.KILL	Bad monitor request made

EQUIPMENT STATUS TABLE

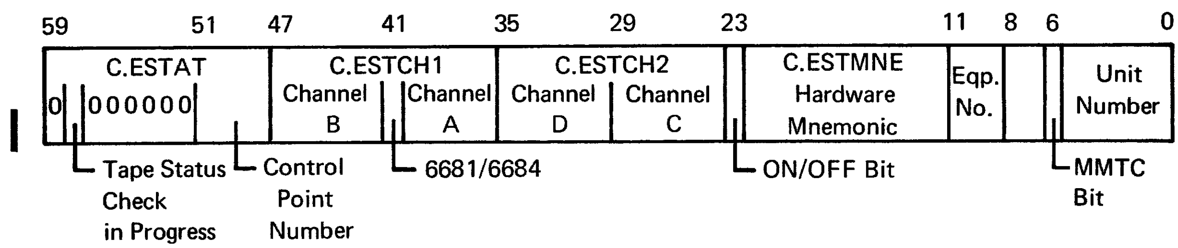
PUBLIC RMS DEVICE ENTRY



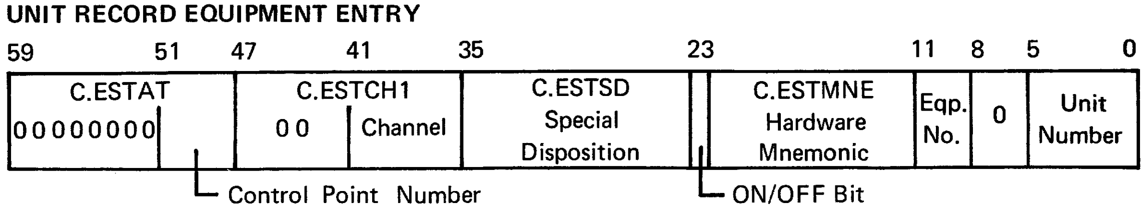
PRIVATE RMS DEVICE ENTRY



MAGNETIC TAPE ENTRY



UNIT RECORD EQUIPMENT ENTRY



EQUIPMENT STATUS TABLE

6612 DISPLAY CONSOLE ENTRY

59	51	47	41	35	32	29	23	11	0
C.ESTAT 0		C.ESTCH1	Channel	Eqp. No.	0	Unit Number	C.ESTMNE DS	0 0 0 0	

Control Point Number

ON/OFF Bit

Available Bit

MULTIPLEXOR ENTRY

59	47	41	35	32	23	11	0
C.ESTAT 0	C.ESTCH1 0 0	Channel	Eqp. No.	0 0 0	C.ESTMNE Hardware Mnemonic	Index into MUX Subtable	

Available Bit

ON/OFF Bit

6000/7000 CHANNEL COUPLER

59	47	35	23	11	0
C.ESTAT	C.ESTCH1 0 0 | Channel	Eqp. No. | 0 0 0	C.ESTMNE C C	0 0 0 0	

Control Point Number

ON/OFF Bit

DDP ENTRY

59	47	41	35	29	23	11	8	0
	C.ESTCH1 Channel 2	Channel 1	C.ESTCH2 Channel 4	Channel 3	C.ESTMNE D P	Eqp. No.		

ON/OFF Bit

DEVICE CODES

Mnemonic	Device Type	Description
AA	01	6603-I disk
AB	02	6638 disk
--	03	Data cell
AC	04	6603-II disk
AL	05	821 data file
AM	06	841 multiple disk drive
AP	07	3234/854 disk pack drive
AF	10	814 disk file
--	11	CDC reserved
AD	12	3637/865 drum
AY	13	Reserved for 844 disk pack
--	14	CDC reserved
--	15	
--	16	
--	17	
--	20	ECS resident file
--	21	CDC reserved
--	.	
--	.	
--	.	
--	27	
--	30	Reserved for installations, RMS devices only
--	.	
--	.	
--	37	
MT	40 xx	7-track magnetic tape†
NT	41 xx	9-track magnetic tape†
--	42 xx	Member file 7-track tape†
--	43 xx	Member file 9-track tape†
TR	44	Paper tape reader
TP	45	Paper tape punch
--	46	Reserved for installations
--	47	Reserved for installations
LP	50	501, 512, 505 line printer
L1	51	501, 505 line printer
L2	52	512 line printer
--	53	CDC reserved
--	54	
--	55	
--	56	Reserved for installations
--	57	Reserved for installations
CR	60	405 card reader
--	61	Remote terminal keyboard
--	62xx	7-track multi-file set tape†
--	63xx	9-track multi-file set tape†

†See following page

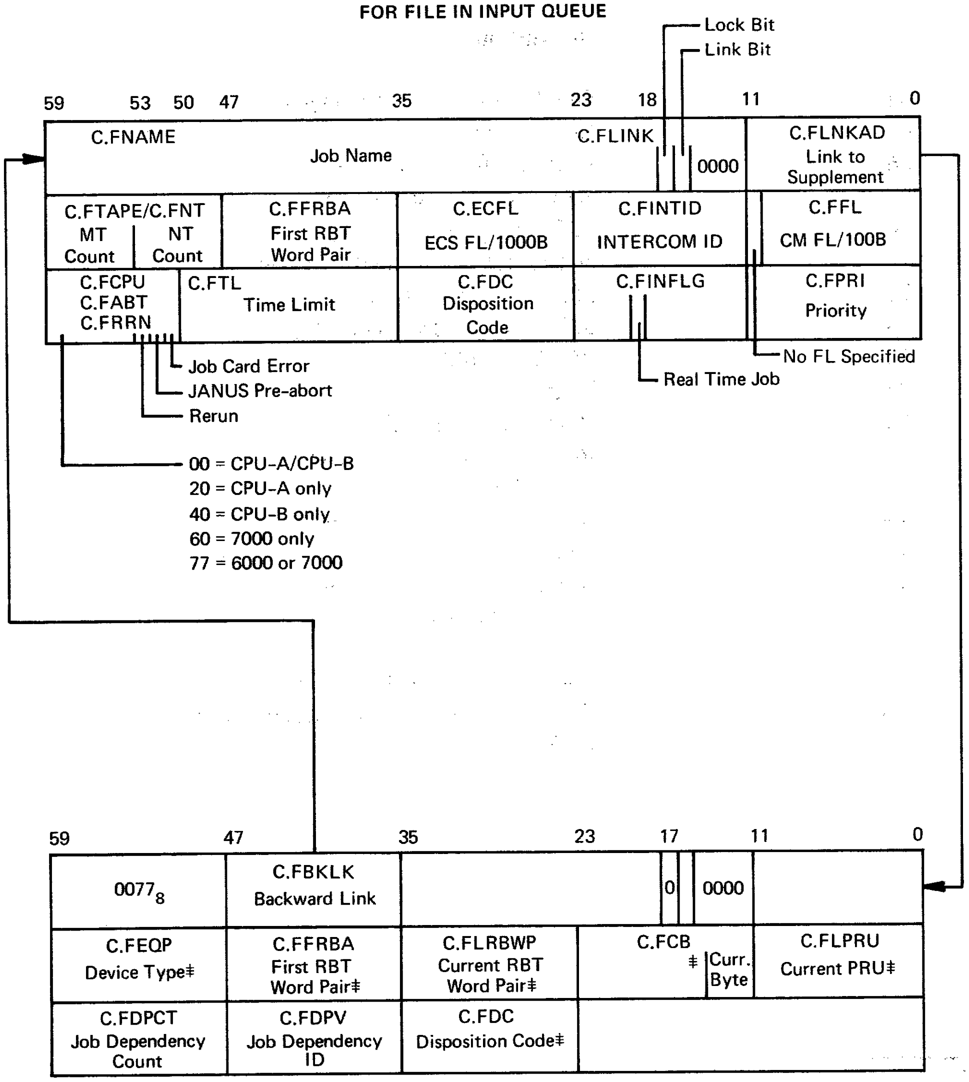
DEVICE CODES (CONT'D)

Mnemonic	Device Type	Description
--	64	Pseudo code for tape staging
--	65	CDC reserved
--	66	Reserved for installations
--	67	Reserved for installations
CP	70	415 card punch
DS	71	6612 keyboard/display console
GC	72	252-2 graphic console
HC	73	523-2 hard copy recorder
FM	74	254-2 microfilm recorder
PL	75	Plotter
--	76	Reserved for installations
--	77	Reserved for installations
DC	--	6671 DSC
IX	--	Reserved for installations
Wx	--	Reserved for installations
Xx	--	Reserved for installations
SC	--	6673/6674 DSC
YC	--	6676 DSC
CC	--	6000/7000 channel coupler
CS	--	7077-1 communications station (LCC)
DP	--	DDP

† Low order 6 bits (xx)	7-track	9-track
----00	HI-density (556 bpi)	CDC-reserved
----01	LO density (200 bpi)	CDC-reserved
----10	HY density (800 bpi)	HD density (800 cpi)
----11	CDC-reserved	PE density (1600 cpi)
--00--	Unlabeled	Unlabeled
--01--	U- or Z-labeled	U- or Z-labeled
--10--	Y-labeled	Y-labeled
--11--	CDC-reserved	CDC-reserved
00----	SCOPE data format	SCOPE data format
01----	CDC-reserved	CDC-reserved
10----	S tape	S tape
11----	L tape	L tape

FILE NAME TABLE

ENTRY AND OPTIONAL SUPPLEMENT FOR FILE IN INPUT QUEUE

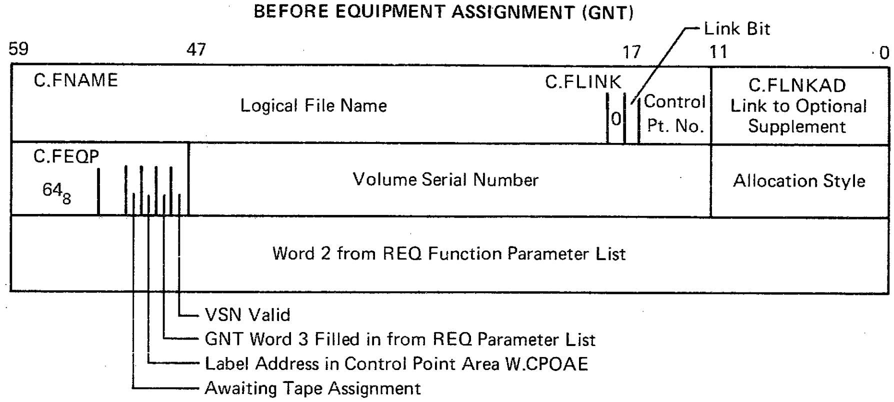


‡If nonzero, fields apply to pre-OUTPUT file

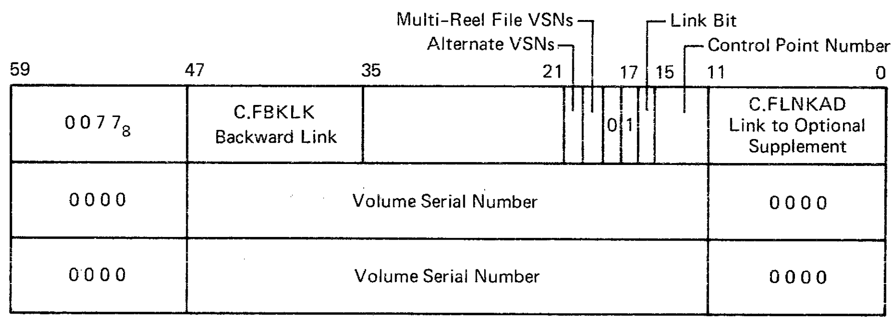
FILE NAME TABLE

TAPE FILE ENTRIES

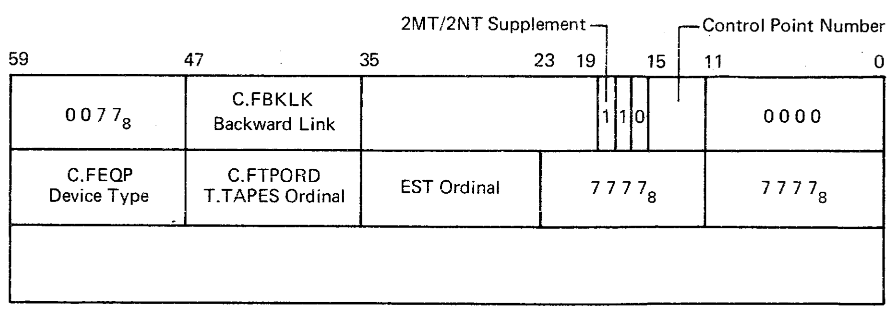
BEFORE EQUIPMENT ASSIGNMENT (GNT)



SUPPLEMENT(S) IF MORE THAN ONE VSN GIVEN



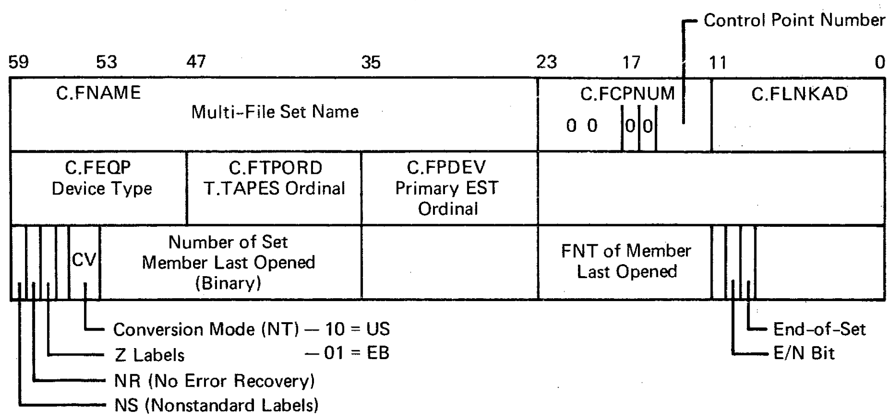
SUPPLEMENT IF 2MT/2NT DECLARED



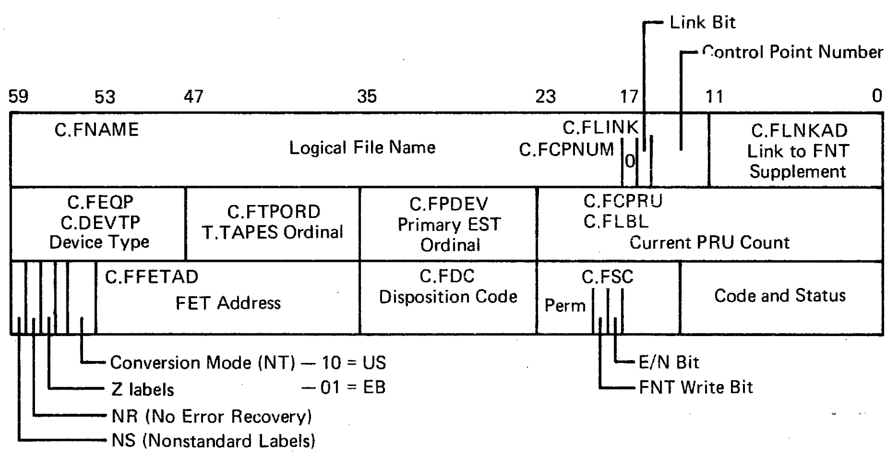
FILE NAME TABLE

TAPE FILE ENTRIES AFTER EQUIPMENT ASSIGNMENT

MULTI-FILE SET MASTER ENTRY

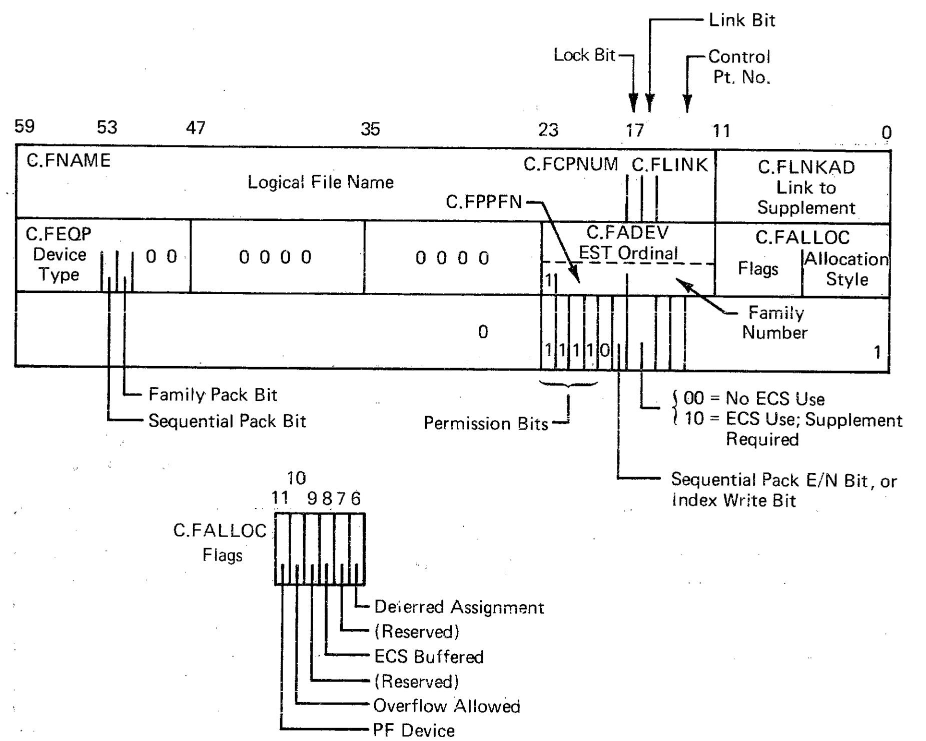


TAPE FILE ENTRY DURING PROCESSING

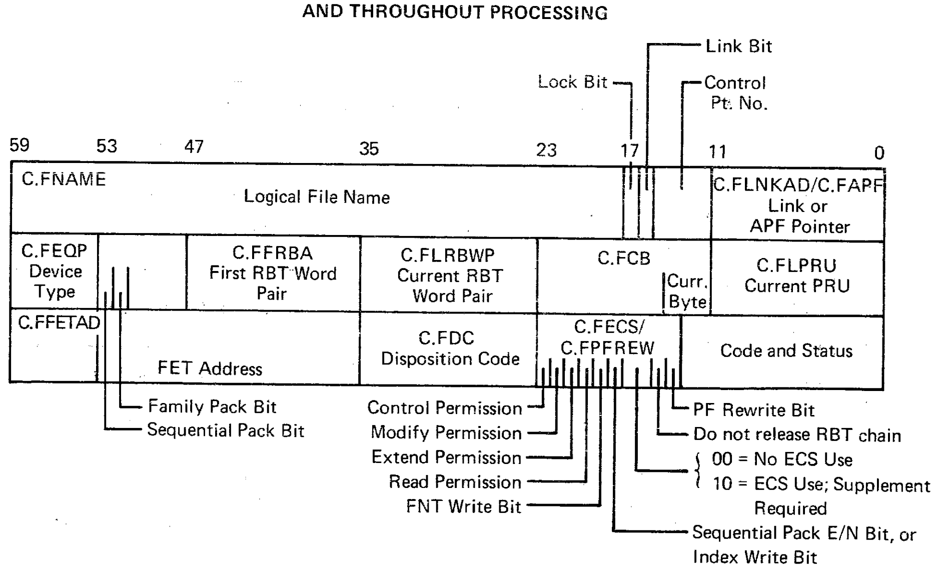


FILE NAME TABLE

ENTRY FOR LOCAL RMS FILE BEFORE ASSIGNMENT TO A DEVICE



AFTER ASSIGNMENT TO A DEVICE AND THROUGHOUT PROCESSING



FILE NAME TABLE

OPTIONAL SUPPLEMENTS FOR LOCAL RMS FILES

SEQUENTIAL PACK FILE SUPPLEMENT

59	47	35	23	17	11 (Control Pt. No.)	0
0 0 7 7_8	C.FBKLK Backward Link			0 0		0 0 0 0
0 0 0 0	T.RPT Ordinal	Primary EST	Secondary EST			0 0 0 0
0 0 0 0	Visual Identifier of First Pack					EST Ordinal of First Pack Assigned

INTERCOM-USER FILE SUPPLEMENT

(Required only when file to be attached to swapped-out job)

59	47 41	35	23 (C.FMUJC MUJ Code)	17	11 (Control Pt. No.)	0
0 0 7 7_8	C.FBKLK Backward Link	(Reserved for Deferred Special DISPOSE)		0 0		C.FAPF APF Pointer
	User Table Address		C.FLOCID INTERCOM ID			

ECS FILE SUPPLEMENT (ECS resident files or IO buffers)

59	53	47	35	23	17	11 (Control Pt. No.)	0	
0 0 7 7_8		C.FBKLK Backward Link	(Reserved for Deferred Special DISPOSE)		0 0		C.FAPF APF Pointer	0
(bits 59–54)	Current Subpage Index	Address of First Subpage		Address of Current Subpage				1
(bits 59–55)				Backspace Count (ECS Overflow While Transferring from RMS to ECS)				2

Word 1
- Bit 54 Outstanding PPCIO Request
- 55 ECS Preallocation Flag
- 56 1 = Release Bit
- 57 0 = Output Buffer
  1 = Input Buffer
- 58 Buffer Overflow
- 59 1 = ECS-Buffered File

Word 2
- Bit 55 Release ECS Buffer After the Current SR
- 56 Index Written (Close Random File
- 59 Transfer in Progress (ECS Resident Random Files)

ECS FILE SUPPLEMENT
(ECS Resident Library)

59	53	47	35	23	17		11	0	
0 0 7 7_8		C.FBKLK Backward Link	(Reserved for Deferred Special DISPOSE)		0 0	Control Pt. No.			0
	EOI Index	Address of First Subpage		Address of EOI Subpage					1
	Current Index	Addr. First Auxiliary Subpage		Address Current Subpage					2

Word 1 Bit 55 ECS Preallocation Flag
56 1 = Release Bit
59 1 = ECS Buffered File

Word 2 Bit 59 Transfer in Progress

FILE NAME TABLE

ENTRIES FOR PERMANENT FILE TABLES

(LAST ENTRIES IN FNT)

Bit positions: 59, 47, 17, 0

59–18		17–0
0 R B T C 00 00		
C.FEQP Device Type	Pointer to First PRU after Chain	
0 S D 0 0 1 00		
C.FEQP Device Type	Pointer to First PRU after Chain	
C.FSDT SD Entry Count		
0 S D 0 0 2 00		
	Pointer to First PRU of Subdirectory	
C.FSDT SD Entry Count		

ENTRY FOR ON-LINE CARD READER FILE

Bit positions: 59, 53, 41, 35, 23, 17, 11, 0

C.FNAME Logical File Name				0 0	Control Pt. No.	0000
60_8	C.FRECCT Record Count	C.FCREC Current Record	C.FPDEV EST Ordinal	00	E O J	C.FLBL Card Count
00	C.FFETAD FET Address		0000	Perm		Code and Status

ENTRY FOR ON-LINE CARD PUNCH FILE

Bit positions: 59, 53, 35, 23, 17, 11, 0

C.FNAME Logical File Name				0 0	Control Pt. No.	0000
70_8	0 0	0000	C.FPDEV EST Ordinal	C.F2PC 2PC Flag	C.FLBL Card Count	
00	C.FFETAD FET Address		0000	Perm	Code and Status	

FILE NAME TABLE

ENTRIES AND REQUIRED SUPPLEMENT FOR FILES IN OUTPUT QUEUES

WAITING IN OUTPUT QUEUE

59		47	35	23 18	11 0
C.FNAME Job Name				Lock Bit 00	0000
C.FEQP Device Type		C.FFRBA First RBT Word Pair	C.FLRBWP Current RBT Word Pair	C.FCB Curr Byte	C.FLPRU Current PRU
Repeat Count	C.FSDC/C.FIOTID Sector Count or Special Dispose or INTERCOM ID		C.FDC Disposition Code	(See Local RMS File)	C.FPRI Priority

DURING PROCESSING

59	53	47	35	23 17 15	11 0
C.FNAME Job Name				0, Link Bit, JANUS Control Pt. No.	C.FLNKAD Link to Supplement
C.FEQP Device Type		C.FFRBA First RBT Word Pair	C.FLRBWP Current RBT Word Pair	C.FCB Curr Byte	C.FLPRU Current PRU
	C.FFETAD FET Address		C.FDC Disposition Code	(See Local RMS File)	C.FCS Code and Status

59	53	47	35	23 17 15	11 0
0077_8		C.FBKLK Backward Link		0 0, JANUS Control Pt. No.	0000
(Reserved)					
Repeat Count	Sector Count			C.FSPDIS Special Dispose	C.FPRI Priority

DISPOSITION CODE VALUES

Non-Allocatable Devices

CK	XXX1	Checkpoint
IU	XXX2	Inhibit unload
CI	XXX3	Checkpoint and inhibit unload
SV	XXX4	Save
CS	XXX5	Checkpoint and save
	XXX6-7777	Reserved

Allocatable Devices

Bit	11	INTERCOM user file (I)
	10	INTERCOM submitted batch job (RB)
	9	Reserved
	8	Special dispose file (SD)
	7	File named OUTPUT (O)
	6	Interrupted print file (IP)

Low Order 6 Bits

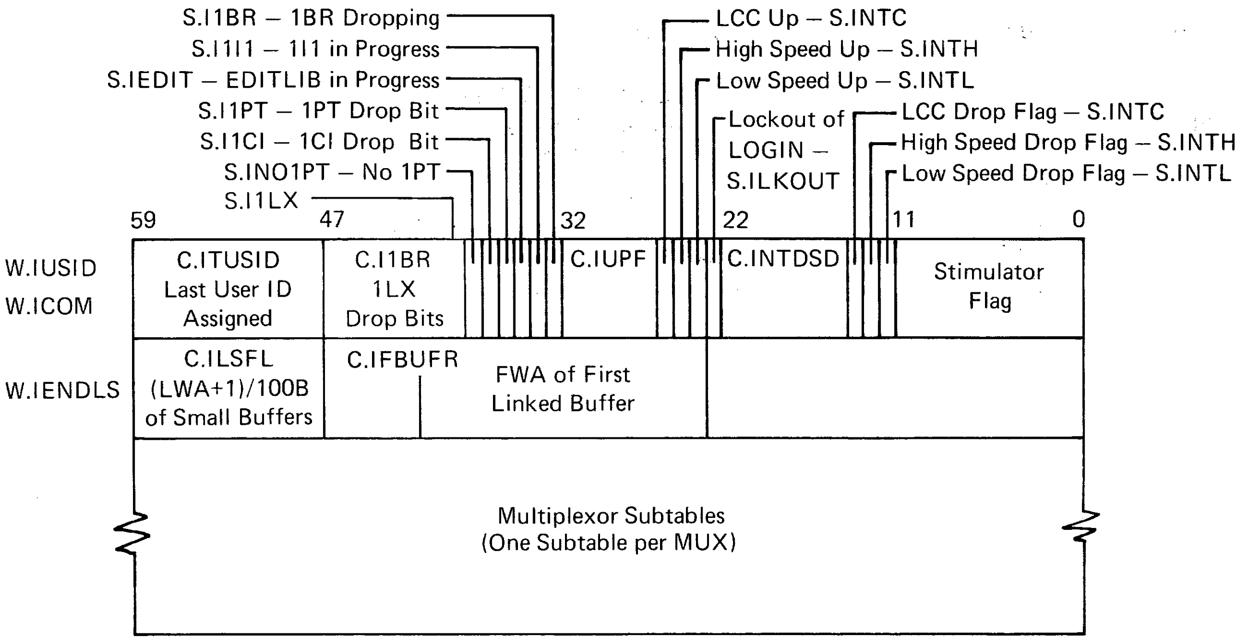
Code	Value	Description
	01	Reserved
	02	Reserved
	03	Reserved
	04	Input job ready for scheduling
	05	Input tape job
	06	Input tape job on VSN display
	07	Reserved
PU	10	Punch installation default character set from display code
PB	12	Punch 6000/7000 format binary
P8	14	Punch 80 column binary
FR*	20	Film print
	21	Reserved
FL*	22	Film plot
	23	Reserved
HR*	24	Hard copy print
	25	Reserved
HL*	26	Hard copy plot
	27	Reserved
PT*	30	Plot
	31-37	Reserved

*Not supported by SCOPE 3.4

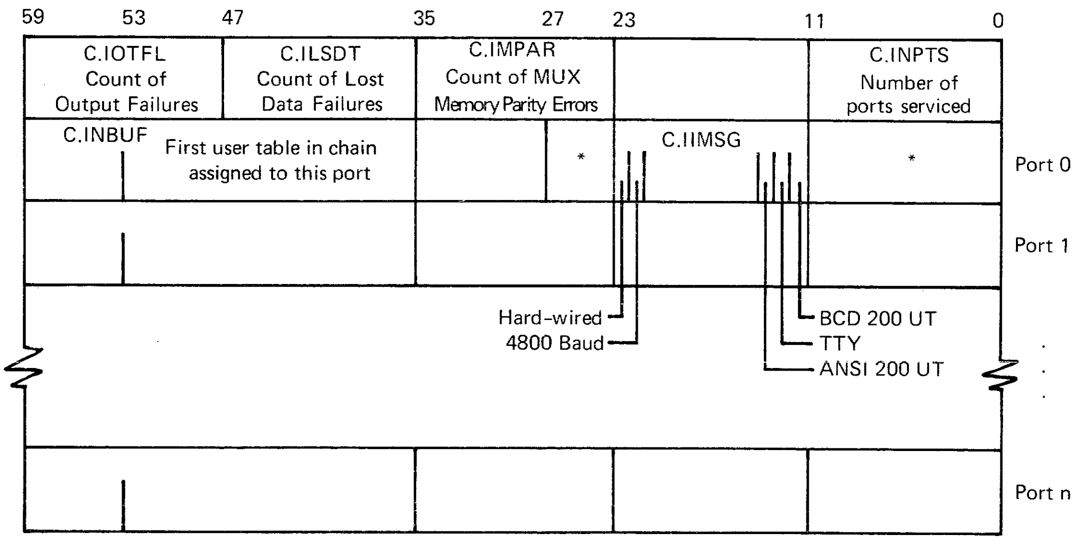
DISPOSITION CODE VALUES (CON'T)

Code	Value	Description
PR	40	Any available 501/505/512
P1	41	Any available 501/505
P2	42	Any available 512
	43	Any available 512 with 48-character set
PE	44	Any available 512 with forms code PE.
	45-67	Reserved
	70-77	Reserved to installations

INTERCOM TABLE



6671 MULTIPLEXOR SUBTABLE

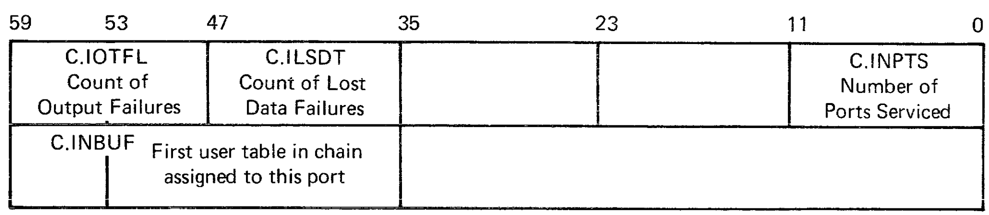


*Treated as one contiguous field (bits "15" - 0)

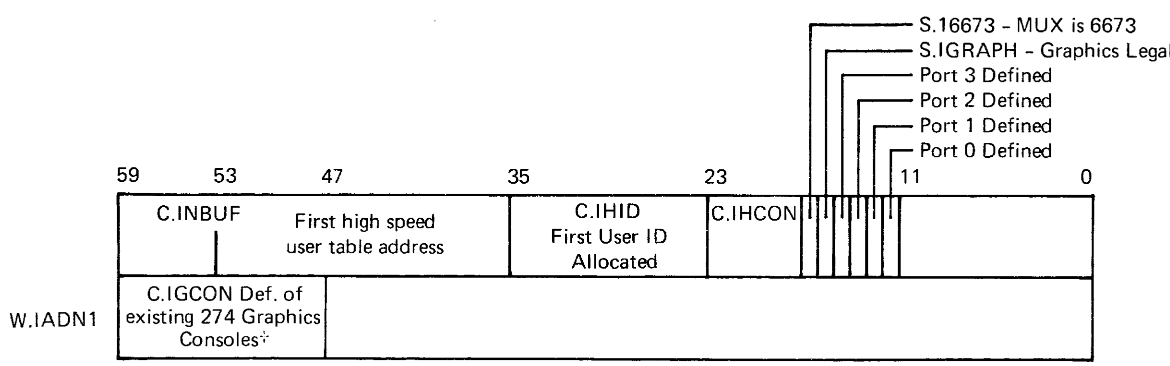
Bits "15" - 0 = 0 : dial-up line

"15" - 0 ≠ 0 : hard-wired: site n exists if bit n = 1

6676 MULTIPLEXOR SUBTABLE

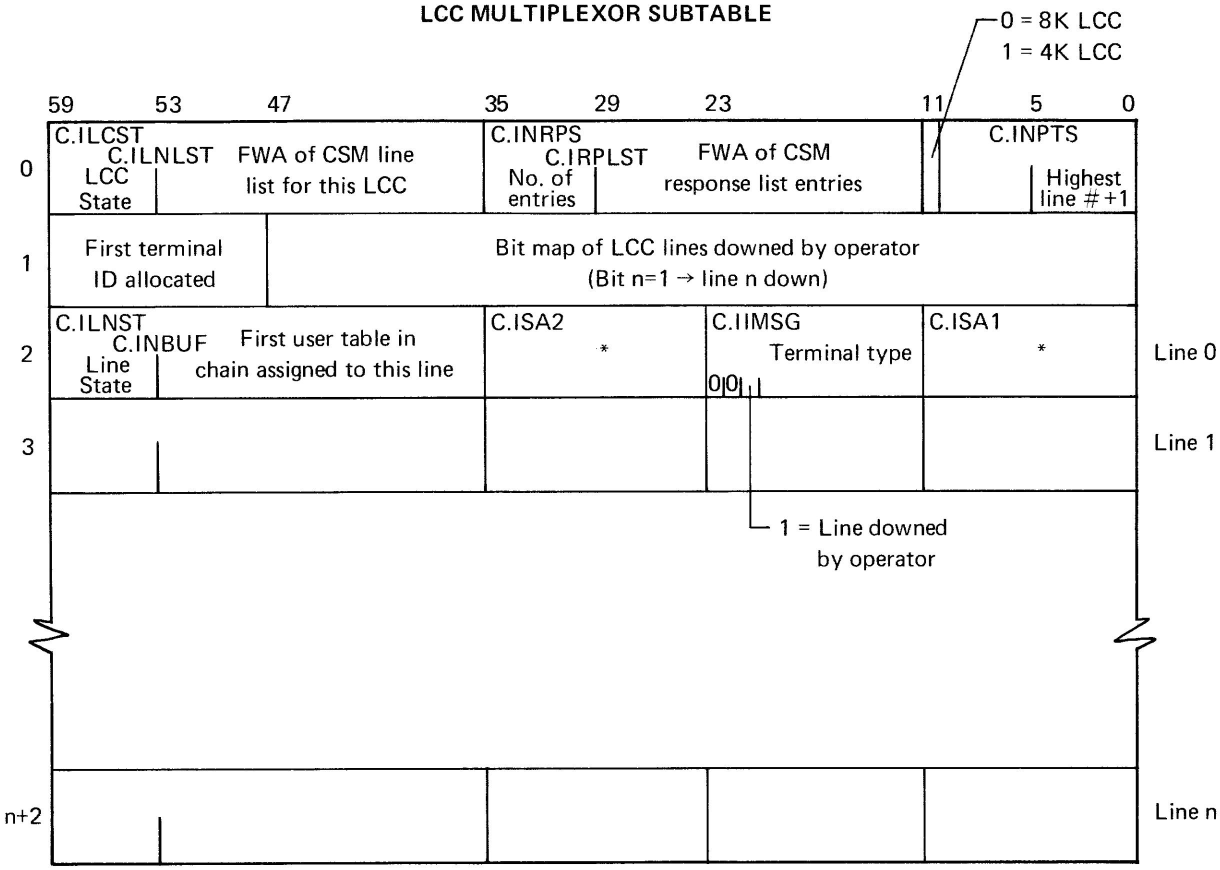


6673/6674 MULTIPLEXOR SUBTABLE



†Word 2 present if S.IGRAPH = 1

LCC MULTIPLEXOR SUBTABLE



Terminal type: 0 = No terminal
2 = TTY
22 = HSBT

*Bit map of streams configured by HSBT operator on this line.

Bits	0	unused			
	1	stream	0	configured	(IDS output)
	2	stream	2	configured	(CRT)
	3	stream	8	configured	(CP1)
	4	stream	12	configured	(LP4)
	5	stream	10	configured	(LP3)
	6	stream	6	configured	(LP2)
	7	stream	4	configured	(LP1)
	8-15	unused			
	16	stream	1	configured	(IDS input)
	17	stream	3	configured	(KBD)
	18	stream	7	configured	(CR2)
	19	stream	5	configured	(CR1)
	20-23	unused			

DEVICE ACTIVITY TABLE ENTRY

59	47	41	23	11	0
C.DATDST DST Ordinal	C.DATEQP Eqp. Type	C.DATACT Activity	(Reserved for Dual Access)	Count Maintained By SPM	

Bit 59: (Reserved for Dual Access)

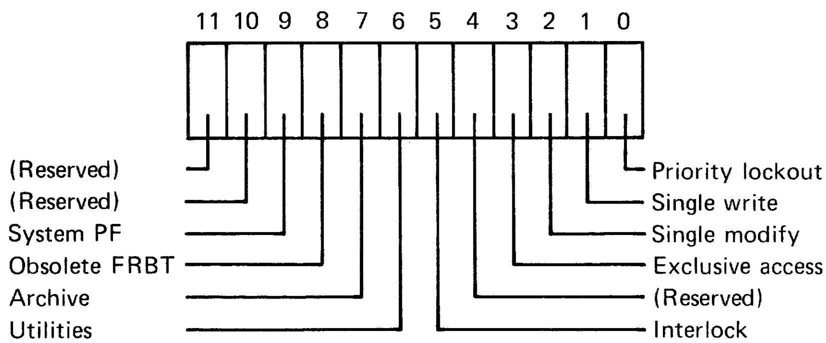
TAPE STAGING TABLE

	59 C.STGMT	47 C.STGNT	35	23	11 0	
W.STGMAX	Number of MT Defined	Number of NT Defined	(Reserved)	(Reserved)	(Reserved)	(Total)
W.STGFRE	Number of MT ON + Unassigned	Number of NT ON + Unassigned				(Available)
W.STGSAT	Number of MT Held by Satisfied Jobs	Number of NT Held by Satisfied Jobs				(Assigned)
W.STGUFD	Unfilled MT Demand	Unfilled NT Demand				(Unfilled Demand)

ATTACHED PERMANENT FILE TABLE

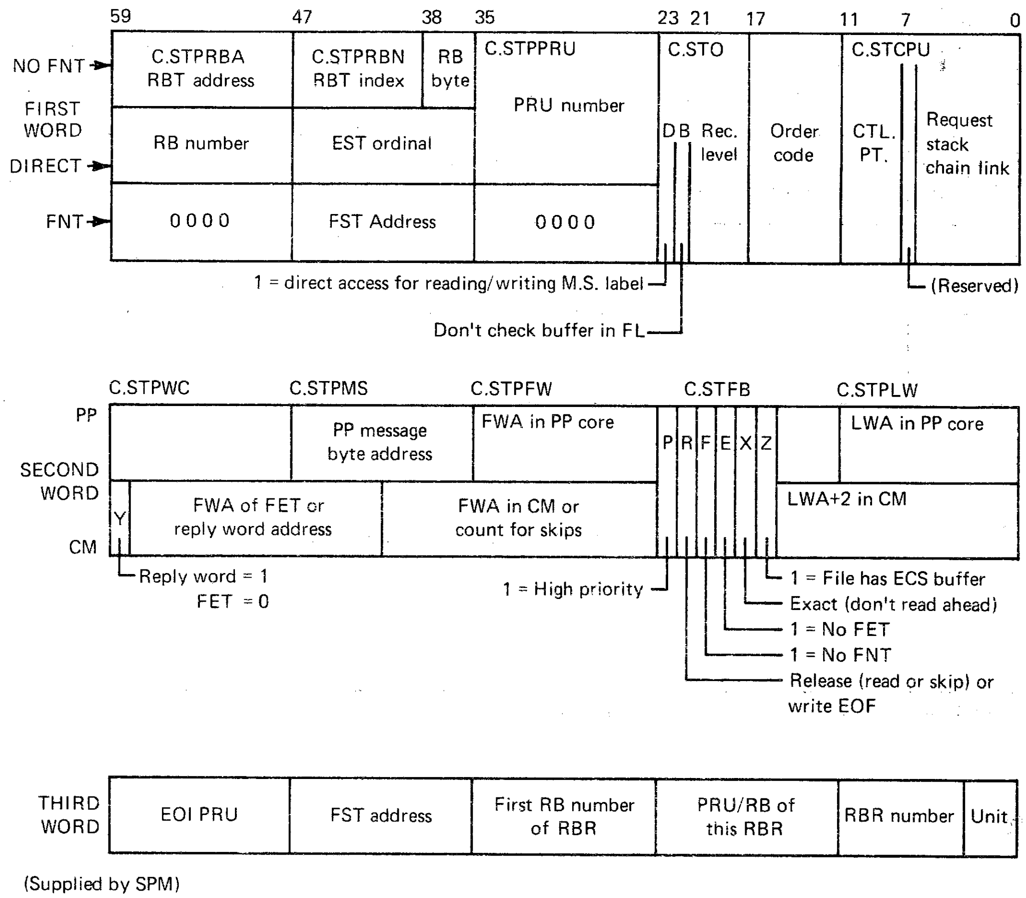
59	47	35	23	17	0	
C.PFD1	C.PFD2	C.PFD3	C.PFD4	C.PFQ* PFQ Address		Word 1
←	PFD Pointer		→			
C.PFRBT First RBTA	C.PFCNT2 Count 2	C.PFCNT Count 1	C.PFCY Cycle No.	C.PFLAG Flag Byte		Word 2

C.PFLAG



*If $\neq 0$, PFQ Address = JDT Address of a job waiting for permanent file.

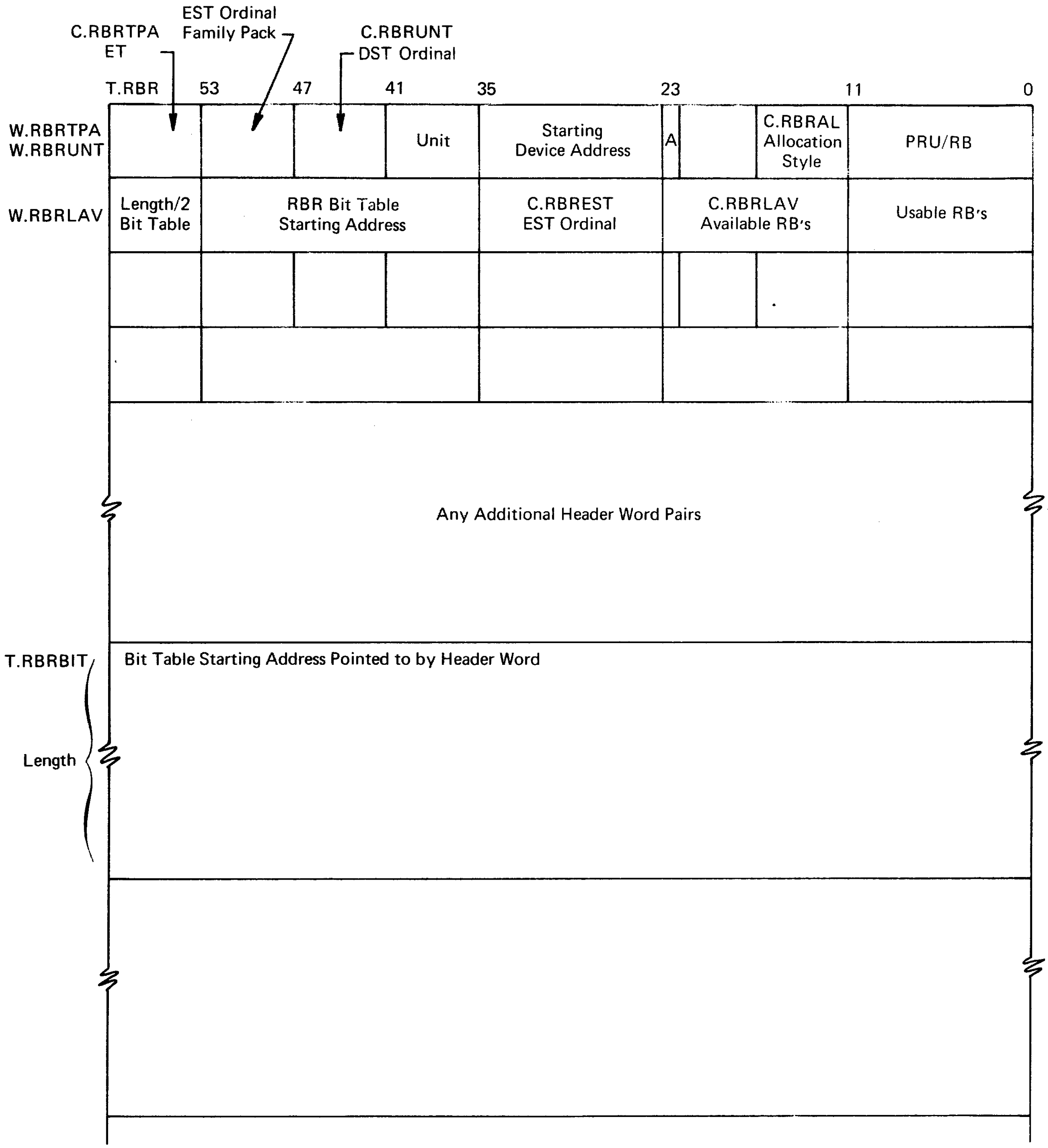
REQUEST STACK ENTRY



STACK PROCESSOR ORDER CODES

00	O.READ	Read into central memory
01	O.RDSK	Readskp into central memory
02	O.RCMPR	Read into central memory, drop first 3 CM words
03	O.RDNS	Read non-stop
04	O.WRT	Write from central memory
05	O.WRTR	Write EOF/EOR from central memory
06	O.RMR	Read multiple records to central memory
07		
10	O.RDP	Read into PPU memory
11	O.RDPNP	Read into PPU, drop first 3 CM words
12	O.SKF	Skip forward
13	O.SKB	Skip backward
14	O.WRP	Write from PPU memory
15	O.WRPR	Write EOF/EOR from PPU memory
16	O.BPRU	Backspace PRU
17	O.RCHN	Evict
20	O.RCTNU	Read nonstop (comparable to tape READN)
24	O.WCTNU	Write nonstop (comparable to tape WRITEN)

RECORD BLOCK RESERVATION TABLE



ET Equipment type
A Files without specified allocation style may go on this device
B EST ordinal of first pack of pack family

DEVICE STATUS TABLE

<table>
<tr><td>59</td><td>47</td><td>41</td><td>35</td><td>23</td><td>17</td><td>14</td><td>11   0</td></tr>
<tr><td>Channel Time Accounting Factor</td><td>Driver Name</td><td>Res. for Inst.</td><td>Pointer to End of Chain</td><td>Alt. Channel</td><td colspan="2">Pri. Channel</td><td>DST Ordinal</td></tr>
<tr><td>Head 1 Position</td><td colspan="2">Head 2 Position</td><td>Pointer to Start of Chain</td><td>Res. for Inst.</td><td></td><td>E</td><td>Non-Zero if a PP is Assigned</td></tr>
</table>

E Equipment

Driver Name A display code letter which is added to the characters 3S to form the overlay name

Valid Letters Are

P	6603-1 Disk
Q	6638 Disk
R	865 Drum
S	854 Disk Pack
T	6603-2 Disk
U	814 Disk
V	821 Disk
W	841 Disk Pack
Y	844 Disk Pack

DEVICE POOL TABLE

Internal Information Passed Between 1EP and 1SP Each Time One Calls the Other

T.DPT

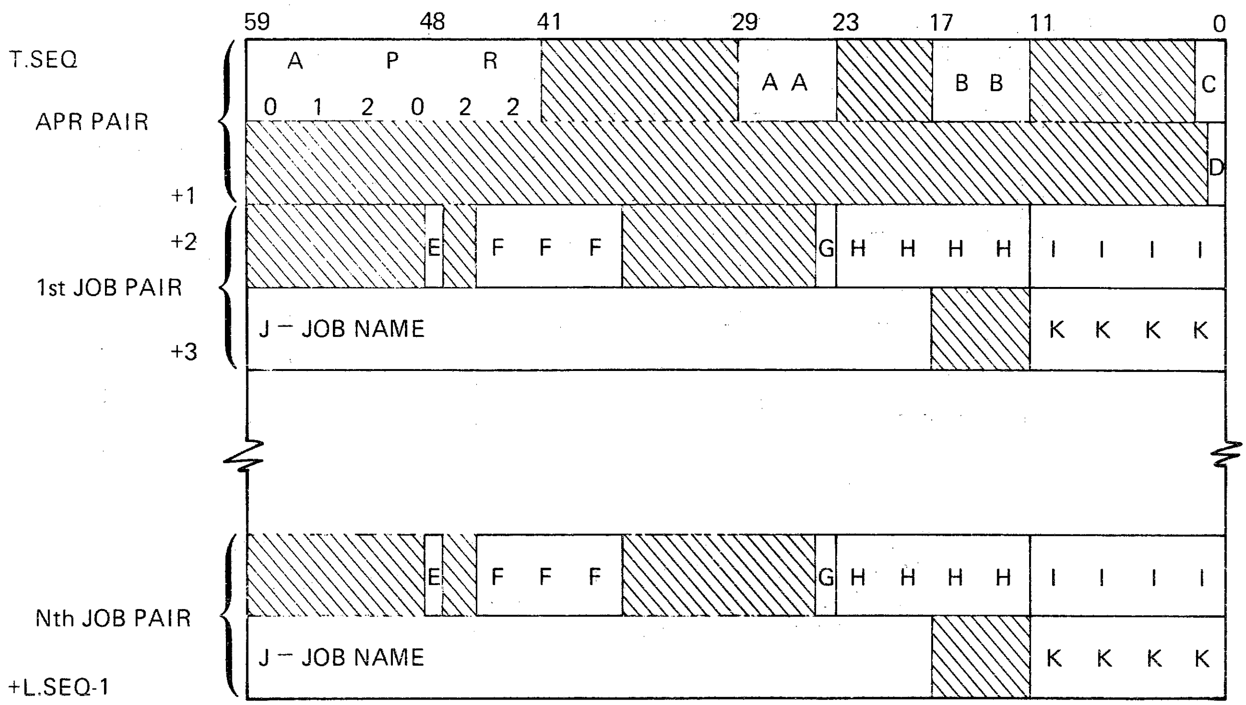
	59	47	35	23	11
1	Y	A_1	A_2	A_3	A_4
2	A_5	A_6	A_7	A_8	A_9
3	A_{10}	A_{11}	A_{12}	B_1	B_2
4	B_3	B_4	B_5	B_6	B_7
5	B_8	B_9	B_{10}	B_{11}	B_{12}
6	C_1	C_2	C_3	C_4	C_5
7	C_6	C_7	C_8	C_9	C_{10}
8	C_{11}	C_{12}	X	Z	(Reserved)

	11 … 5	4	3	2 … 0
A_i	Stack Request Ordinal	ECS Bit	PRI Bit	Unit Number
B_i	First Physical Sector for Latency Calculation	Write Flag	Bypass Bit	Subpriority

	11 … 0
C_i	Standard First RB Number for Positioning Time Calculation

	11 … 0
X	Ordinal of S.R. that may be Reissued
Y	Number of S.R. Waiting in Pool
Z	Limiting RB for Search in RBR During Service of Write Stack Request

SEQUENCER TABLE



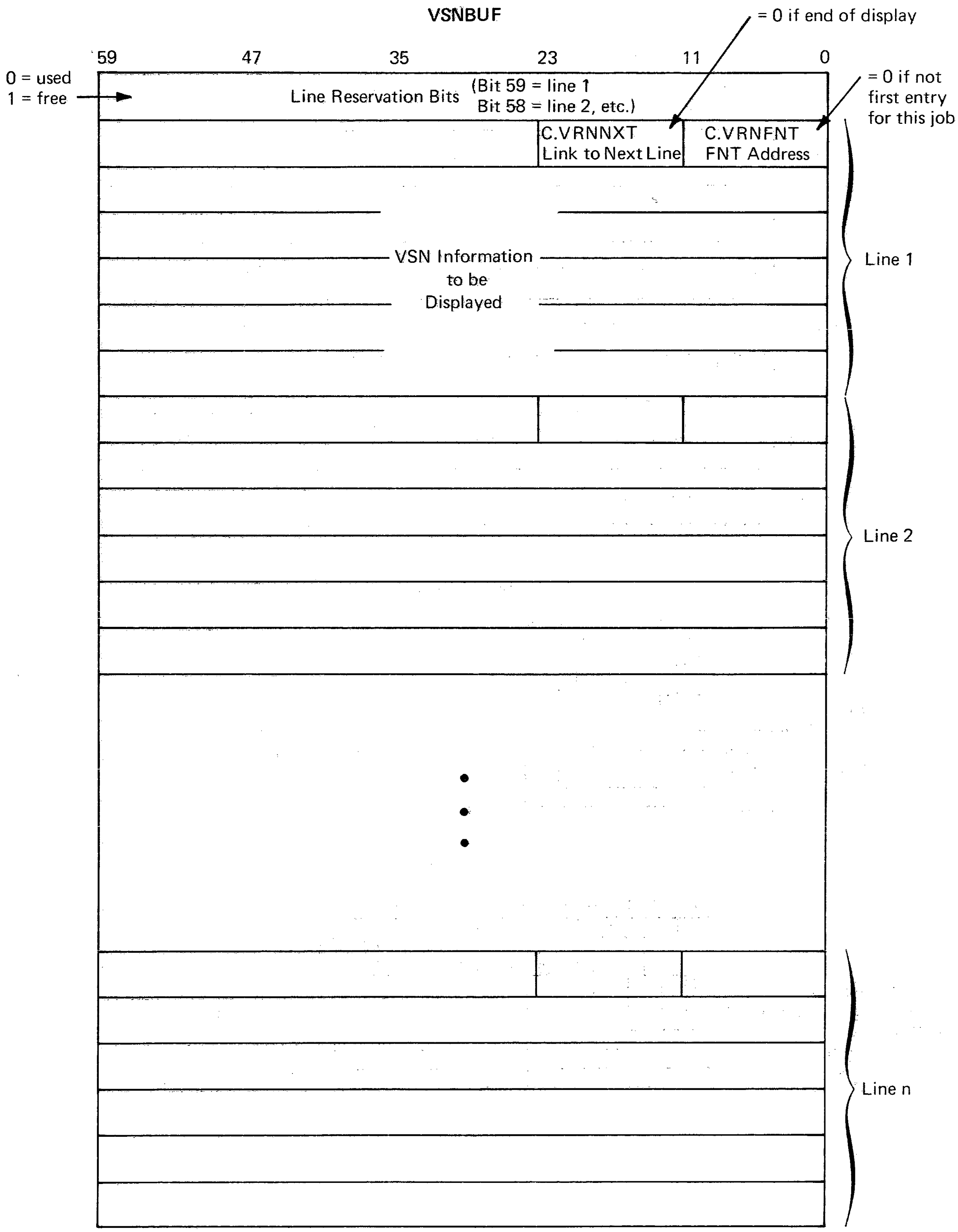
A = Maximum number of job entries (L.SEQ-2/2)
B = Number of jobs in Sequencer Table
C = On/Off/Drop Flag
  0-Off
  1-On
  2-Drop
D = Table Interlock Flag
E= Entry Full Flag
F = Diagnostic Flag Bits
  Bit 0 = CT3
  Bit 1 = MY1
  Bit 2 = CM6
  Bit 3 = CU1
  Bit 4 = ALS
  Bit 5 = FST
  Bit 6 = EC2
  Bit 7 = ALX
  Bit 8 = CEFAP
  Bit 9 = reserved
  Bit 10 = reserved
  Bit 11 = reserved
G = Entry Drop Flag
H = Interval
I = Clock
J = Job Name
K = Last known FNT Address

ROTATING MASS STORAGE DIAGNOSTIC TABLE

T.RMS

59	51	40	29	21	10 0
E0	E0L	E0U	E1	E1L	E1U
E2	E2L	E2U	E3	E3L	E3U
E4	E4L	E4U	E5	E5L	E5U
E6	E6L	E6U	E7	E7L	E7U

E0, E1, E2, etc. — EST ordinal of RMS device that was preallocated
E0L, E1L, etc. — Lower cylinder boundary of preallocated area
E0U, E1U, etc. — Upper cylinder boundary of preallocated area


VSNBUF
= 0 if end of display
59
47
35
23
11
0
0 = used
1 = free
Line Reservation Bits
(Bit 59 = line 1
Bit 58 = line 2, etc.)
= 0 if not
first entry
for this job
C.VRNNXT
Link to Next Line
C.VRNFNT
FNT Address
VSN Information
to be
Displayed
Line 1
Line 2
Line n

T.TAPES TABLE

<table>
<tr><td></td><td>59</td><td>47</td><td>35</td><td>29</td><td>23</td><td>17</td><td>11 … 0</td></tr>
<tr><td></td><td>EST Ordinal (Binary)</td><td>FNT Address</td><td colspan="2">Control Point Number</td><td colspan="2">MT or NT (Display Code)</td><td>EST Ordinal (Display Code)</td></tr>
<tr><td>W.TFLN1</td><td colspan="7">Label Name</td></tr>
<tr><td></td><td colspan="5">Label Name</td><td colspan="2">Position Number</td></tr>
<tr><td></td><td>Edition Number</td><td colspan="2">Retention Cycle</td><td colspan="4">Creation Date</td></tr>
<tr><td>W.TREEL</td><td colspan="4">Multi-File Name</td><td colspan="3">Reel Number</td></tr>
<tr><td>W.TFLGS</td><td colspan="2">C.TFLGS
Flag Bits</td><td colspan="5"></td></tr>
<tr><td>W.TVRN</td><td colspan="4">Volume Serial Number of Current Reel</td><td colspan="3">Channel Byte Count of Previous Record</td></tr>
<tr><td>W.TVRN1</td><td colspan="4">Volume Serial Number of First Reel</td><td colspan="3">PRU Number of Last PRU That Got Noise Warning 1</td></tr>
</table>

C.TFLGS — Job-Oriented Flag Bits

59	58	57	56	55	54	53	52	51	50	49	48
GO	RC		ST					VP	MP	LC	EC

S.TTFGO— Go bit
S.TTFRC — Recheck bit
(Reserved)
S.TTFST — Stop Staging
(Reserved)
(Reserved)

Label expiration checked — S.TTFEC
Label checked — S.TTFLC
Multi-file position bit — S.TTFMP
VSN-printed bit — S.TTFVP
(Reserved)
(Reserved)

C.TFLGS+1 — Unit-Oriented Flag Bits

47	46	45	44	43	42	41	40	39	38	37	36
VT	OV	SC	RD			EL	BL	LB	FS	RI	WN

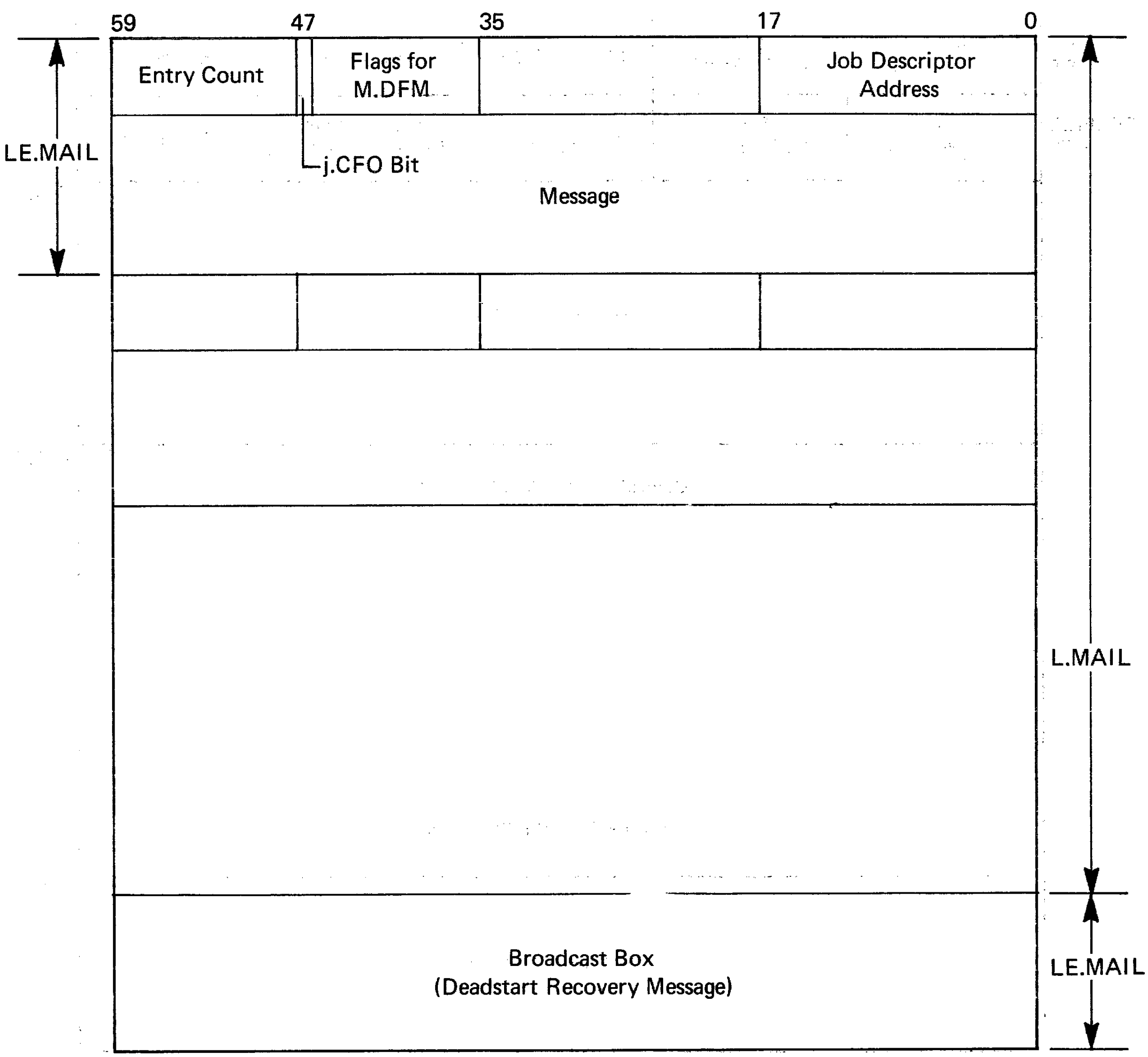
S.TTFVT— VSN read from tape
S.TTFOV— Operator-supplied VSN
S.TTFSC — Scratch
S.TTFRD— Ready
(Reserved)
(Reserved)

Unexpired label w/ring—warning — S.TTFWN
Ring in bit — S.TTFRI
Flash save bit — S.TTFFS
Labeled — S.TTFLB
Blank labeled — S.TTFBL
Expired label — S.TTFEL

REMOVABLE PACK TABLE ENTRY (RPT)

C.RPTEST C.RPTVID	C.RPTFST	C.RPTCP	C.RPTPKN	
59	47	35	23	11 … 0
EST Ordinal (Binary)	FST Address	Control Point Number	A or M, A or P (Display Code)	EST Ordinal (Display Code)
Visual Identifier (VID) of Current Pack			Pack Number of Current Pack	

MAILBOX



DAYFILE FET AND BUFFER AREA

	59	47	35	23	11	0
CP0 FET	Index (IN_0 Pointer)	Size ($LIMIT_0$-$FIRST_0$)	($FIRST_0$-T.DFB)	Remaining Number of Preallocated PRUs	MTR Code Word	T.DFB
CP1 FET	Index (IN_1 Pointer)	Size ($LIMIT_1$-$FIRST_1$)	($FIRST_1$-T.DFB)	(Reserved)	(Reserved)	
. . .						
CPn FET	Index (IN_n Pointer)	Size ($LIMIT_n$-$FIRST_n$)	($FIRST_n$-T.DFB)	(Reserved)	(Reserved)	
CERFILE FET	Index (IN_c Pointer)	Size ($LIMIT_c$-$FIRST_c$)	($FIRST_c$-T.DFB)	Remaining Number of Preallocated PRUs	MTR Code Word	
	System Dayfile Buffer					$FIRST_0$
	Control Point 1 Dayfile Buffer					$FIRST_1$
	. . .					$FIRST_2$
	Control Point n Dayfile Buffer					
	CE Error File Buffer					$FIRST_c$

PERIPHERAL JOB TABLE

PP Input Register Image	One Entry
W.PPMES4 Image	
W.PPMES5 Image	
W.PPMES6 Image	

SCHEDULER PERFORMANCE TABLE (SCHPT)

S.PTNSTD
S.PTINIT
S.PTSTOP
S.PUBNCH
S.PTFJOB

	59	29	5 0
T.SCHPT	F L A G S		(bits 4–0: S.PTFJOB, S.PUBNCH, S.PTSTOP, S.PTINIT, S.PTNSTD)
	Elapsed Time		
	CPU Idle Time		
	User Mode System Program Execution Time		
	PPU Idle Time		
	Number of Storage Moves		
	Number of Words Moved		
	Number of Swap-outs/Roll-outs		
	Number of Words/100B Swapped or Rolled Out		
	Time to Get All Quiet (seconds * 4096)		Time to Swapout Jobs (seconds * 4096)
	Number of INTERCOM Commands Processed		Number of Batch Jobs Processed
	Number of CPMTR Calls		
	Reserved for Storage of Intermediate Times		

S.PTNSTD — Manual/benchmark option selected
S.PTINIT — Initialization of TSPT
S.PTSTOP — Stop data gathering
S.PUBNCH — Benchmark option selected
S.PTFJOB — First Job execution started

JOB CONTROL AREA

	59	47	35	23	17	11	0
Input Queue Entry	C.JCMXB Max # Class 1 Batch	C.JCMTB Max # Class 2 Batch	C.JCCLK 1 IB Flag	C.JCQP Quantum Priority		C.JCBQ Quantum Value	
	C.JCCNB Current # Class 1 Batch	C.JCCTB Current # Class 2 Batch	C.JCEMC # Empty FNT Entry	a | C.JCNTJ # Ready-to-run Tape Jobs		b | C.JCNJI # Ready-to-run Non-tape Jobs	
Class 1 (No Non-allocatable Devices)	C.JCMIN Minimum Class Priority	C.JCMAX Maximum Class Priority	C.JCAR Aging Rate	C.JCQP Quantum Priority		C.JCBQ Quantum Value	
					C.JCFRST Address of First JDT in Chain		
Class 2 (Non-allocatable Devices)	C.JCMIN	C.JCMAX	C.JCAR	C.JCQP		C.JCBQ	
					C.JCFRST		
Class 3 (Interactive Jobs)	C.JCMIN	C.JCMAX	C.JCAR	C.JCQP		C.JCBQ	
					C.JCFRST		
Class 4 (Multi-user Jobs)	C.JCMIN	C.JCMAX	C.JCAR	C.JCQP		C.JCBQ	
					C.JCFRST		
Class 5 (Express Jobs)	C.JCMIN	C.JCMAX	C.JCAR	C.JCQP		C.JCBQ	
					C.JCFRST		

a = 1 Fixed priority tape jobs in Input Queue.
b = 1 Fixed priority non-tape jobs in Input Queue.

JOB DESCRIPTOR TABLE ENTRY

	59	47	35	23	17	11	0
W.JDNAM W.JDLNK	Job Name				C.JDLNK Link to Next JDT in Chain		
W.JDSWP	C.JDEQC Equipment Code	C.JDFRB First RBT	C.JDFLG Flags	C.JDFL FL/100B		C.JDPFL Positive FL	
	First subpage address of ECS swap file						
W.JDDSD	C.JDCPN CP# Priority	C.JDORD J.D. Ordinal	C.JDTL Time Limit	C.JDOPF Operator Flags		C.JDORG SSW Origin	
W.JDMGR	C.JDJST Job St. Class	C.JDTIN PFM C.JDPFM Bits	C.JDTIN Time into Chain or APF Pointer	C.JDBP Base Priority		C.JDRU PP/CP Time	
W.JDINT	C.JDID INTERCOM User ID	C.JDCPT CPU Time	C.JDIUTA	User Table Address		C.JDLPFL Length of Positive FL/100B	

NOTES: JOB DESCRIPTOR TABLE

W.JDSWP **C.JDFLG**

Bit	0	unused	
	1	unused	
	2	unused	
	3	S.JDNFNT	Set if 1AJ should not search FNT table
	4	S.JDROLL	Set if job cannot be swapped out
	5	S.JDSKFL	Set if FL is to be skipped on swap file
	6	S.JDECS	Set if swap file is on ECS
	7	S.JDNJ	Set to indicate control cards must be read from INTERCOM area
	8	S.JDLGO	Set if swap file must not be generated for this INTERCOM job
	9	S.JDLGI	Set if job is a LOGIN command
	10	S.JDNRR	Set if job cannot be rerun
	11	S.JDBCB	Set if recovery took place

W.JDDSD **C.JDOPF** **Contains operator flags:**

Bits	0-2	Error codes:	F.JDKILL, F.JDDROP, F.JDRRUN, F.JDRRNP
	3	S.JDLOK	If job must not be brought to a control point
	4	S.JDNS	If job must not be swapped/rolled out when at control point
	5	S.JDGO	If operator types GO
	6	S.JDEXP	If job is to be placed in express queue
	9-11	0	

C.JDORG

Bits	6-11	Sense switches	
	0-5:		
	5	S.JDINT	Set for standard INTERCOM job
	4	S.JDMUJ	Set for multi-user job
	3	S.JDGR	Set for Graphics job
	2	S.JDRT	Set for real time job

W.JDMGR **C.JDJST:**

Bits	48-53:	Contain job class	
	01	F.JDBAT	For batch job with no non-allocatable device requirements
	02	F.JDBNA	For batch job with non-allocatable device requirements
	03	F.JDINT	For standard INTERCOM job
	04	F.JDMUJ	For multi-user job
	05	F.JDEXP	If express handling requested for this job

NOTES: JOB DESCRIPTOR TABLE (CONT'D)

Bits	54-59:	Describe job status	
	0X	F.JDLMB	Waiting for entry in scheduling structure
	1X	F.JDWCM	Waiting for CM
	2X	F.JDWPF	Waiting for permanent file availability
	3X	F.JDWDA	Waiting for device assignment
	4X	F.JDWOA	Waiting for operator action
	5X	F.JDWIA	Waiting for INTERCOM action
	X1	F.JDSWI	Waiting for swap in
	X2	F.JDACT	Currently executing at control point
	X3	F.JDWCC	Waiting for scheduler action

C.JDPFM

Bit	42	Read permission desired
	43	Extend permission desired
	44	Modify permission desired
	45	Control permission desired
	46	Exclusive access desired
	47	Purge bit

CEFAP BUFFER AREA

T.BCFAP	CEM-Working Flag
	Buffers

Buffer for Error Codes 0-3

X0 = ECS FWA of Transfer
A0 = CM FWA of Transfer
Bj+K = Length of Transfer
Exact ECS Address of Transfer
CEFAP Error Code (Unweighted)
Pointer to Next Buffer (0 = End of Chain)

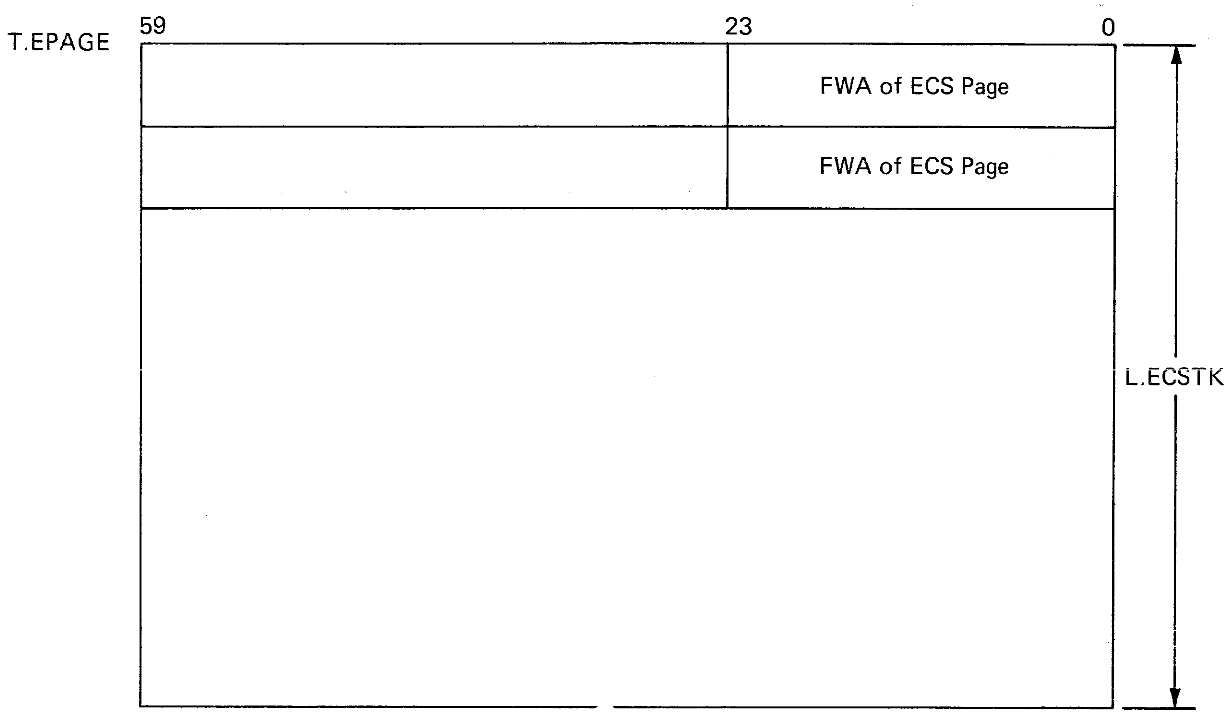
Buffer for Error Codes 20-27

ECS FWA of Transfer (or Unused)
Port Channel
Length of Transfer (or Unused)
PPNT Address + 1
CEFAP Error Code
Pointer to Next Buffer (0 = End of Chain)

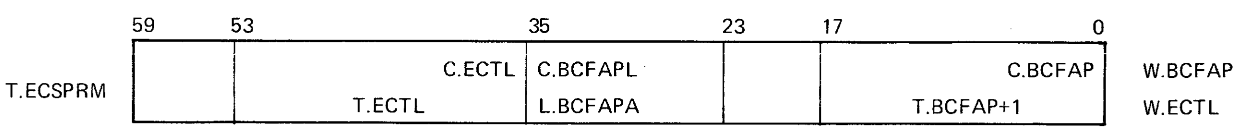
Error Code (Unweighted)

- 0 = Write Abort
- 1 = Read Parity Error
- 2 = Recovered Read Parity
- 3 = Interlock Register Abort
- 20 = Unclassified DDP Error (DDP Reply = 0)
- 21 = DDP Found ECS Abort (DDP Reply = 1)
- 22 = DDP Channel Drops Active
- 23 = DDP Channel Stays Full
- 24 = DDP Found ECS Read Parity Error (DDP Reply = 4)
- 25 = Unclassified DDP Error (DDP Reply = 5)
- 26 = DDP Unable to Connect
- 27 = DDP Logically OFF

EMPTY PAGE STACK



(A6 + B5 in CP.MTR exchange
package points to next entry)



SUBPAGE BUFFER

	59	47	35	23	17	0
T.SUBPG	Subpage Type	FWA of this System Page		Page Address of Current Data if Input Buffer		
	File Name				Number of Available Words in Page if Output Buffer	
			JDT Ordinal	Maximum Buffer Length		
	Record Descriptors					

RECORD DESCRIPTORS

59	55	51	41	23	0	
id= 10XY	Level		Data Length	Data Address		Record

X = 1 Beginning of a new data page
Y = 1 Transmission RMS parity error flag

59		41	23	0	
id= 11XY		Data Length	Data Address		Split Record

X = 1 Beginning of a new data page
Y = 1 Transmission RMS parity error flag

59		23	0	
id= 0100		Next Subpage Address		Forward Link

59		23	0	
id= 0110		Previous Subpage Address		Backward Link

59		47	23	0	
id= 000X		FWA of Available Space in Data Page	Current Buffer Length		End of Descriptor List

X = 0 End of ECS buffer
X = 1 End of Information on disk

59		41	23	0	
id= 0010		Index of First Entry in Preallocated Space †	Address of Preallocated Space in ECS †		Beginning of Descriptor List

† Used only for the library

NOTES: SUBPAGE BUFFER

Subpage Type (bits 59-48)

Subpage Position (bits 49-48)

00 = continuation subpage
01 = first subpage in a file
10 = last subpage

Data Type

Bit	50	release data as read
	51	(reserved)
	52	I/O buffer (with bit 50)
	53	library file (ZZZZZ06)
	54	ECS resident file
	55	swap file
	56	auxiliary file for ECS resident random file
	57	index for random ECS file
	58	(reserved)
	59	(reserved)

id (bits 59-56)

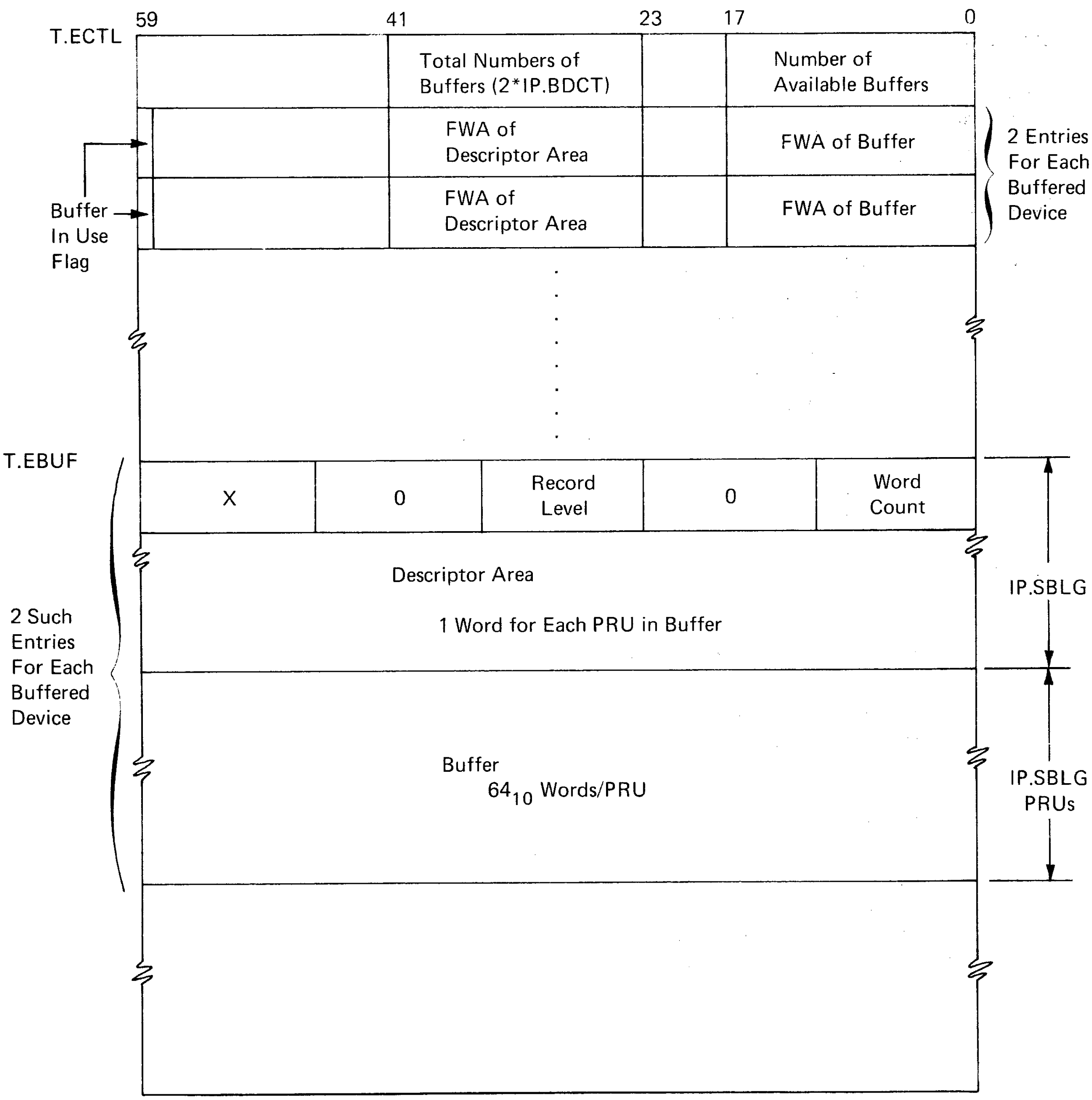
0XXX = system descriptor

0000	=	end of list; continued on disc
0001	=	end of list; EOI on file
0010	=	beginning of list
0100	=	forward link pointer
0110	=	backward link pointer

1XXX = data descriptor

10XX	=	full record descriptor
11XX	=	split record descriptor (full record described by this and next descriptor)
1X0X	=	current data page
1X1X	=	new data page
1XX0	=	no parity error
1XX1	=	parity error in record

ECS BUFFER FOR RMS—ECS TRANSFER



X = 0 No Parity Error
X = 4000B Uncorrectable Disk Read Parity Error

CMR DIRECTORY

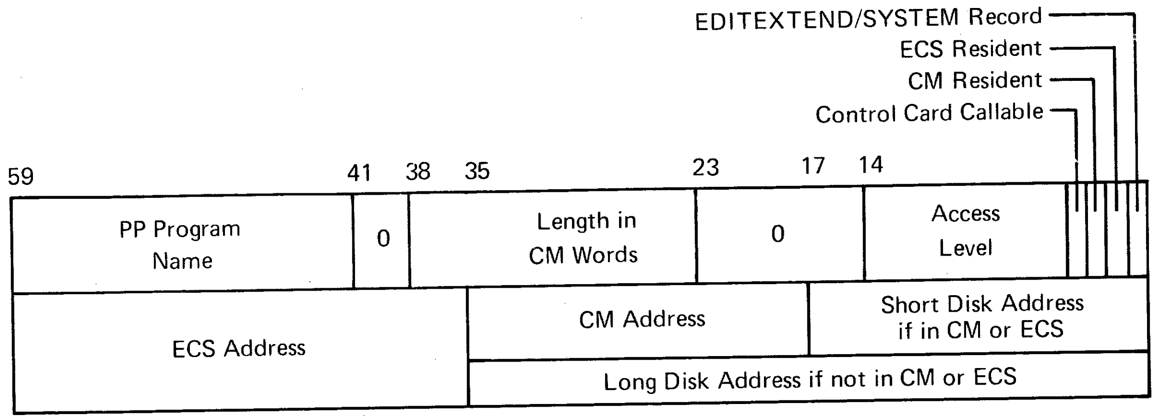
	59–42	41–36	35–24	23–18	17–0
T.LIB					LWA+1 of LNT
	Library Name Table (LNT) (5 Words per Entry)				
		LWA+1 of PP Program Bodies			LWA+1 of PPNT
	PP Program Name Table (PPNT) (2 Words per Program)				
	PP Program Bodies				
	CM Resident Libraries				

LIBRARY NAME TABLE ENTRY

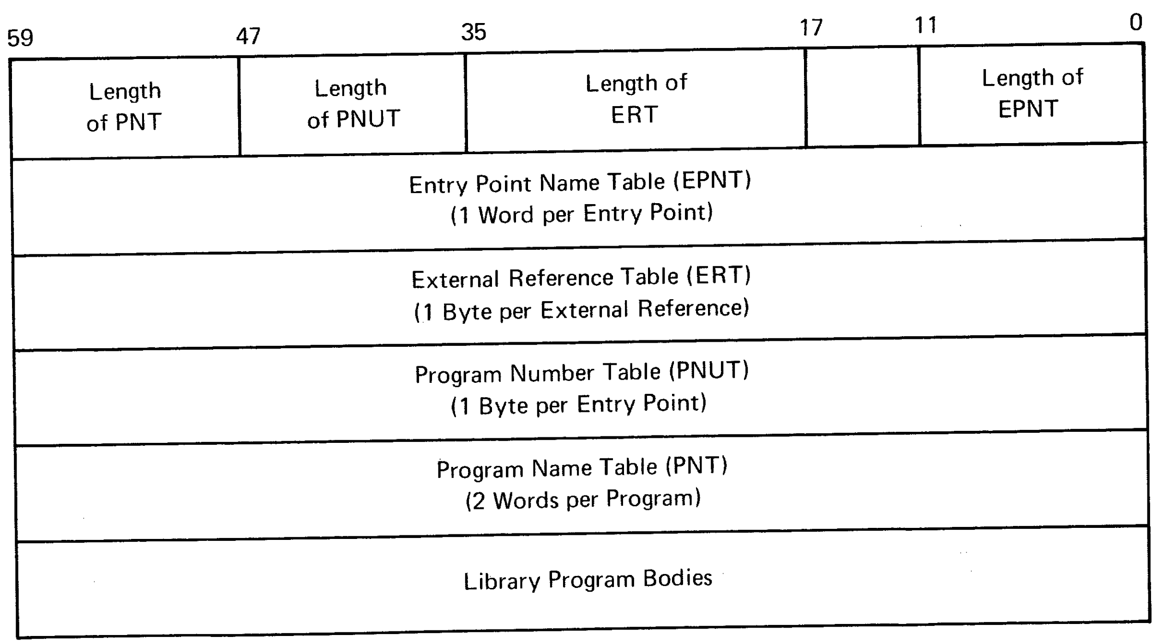
CM Library
ECS Library
EDITEXTEND/
SYSTEM Library
New Library
Modified Library
Use Disk Address

	59–36	35–18	17–0
	Library Name		(flag bits; 14; 3)
Entry Point Name Table	Address of ECS Descriptor	CM FWA of Library	Disk Address of Library
External Reference Table	Address of ECS Descriptor	CM FWA of Library	Disk Address of Library
Program Number Table	Address of ECS Descriptor	CM FWA of Library	Disk Address of Library
Program Name Table	Address of ECS Descriptor	CM FWA of Library	Disk Address of Library

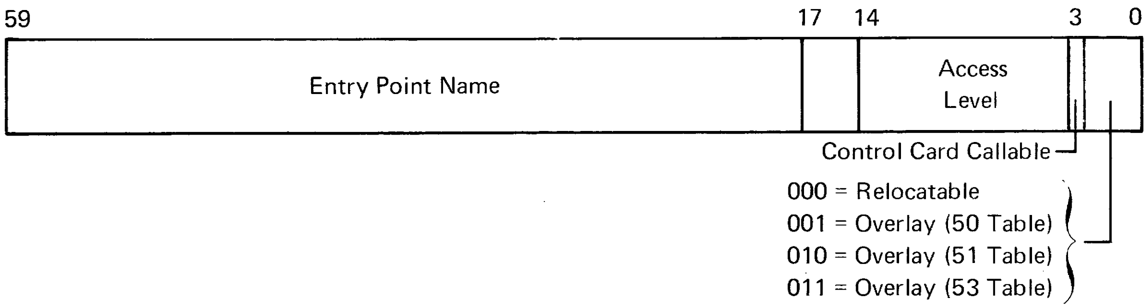
PP PROGRAM NAME TABLE ENTRY



CM RESIDENT LIBRARY FORMAT



EPNT ENTRY FORMAT



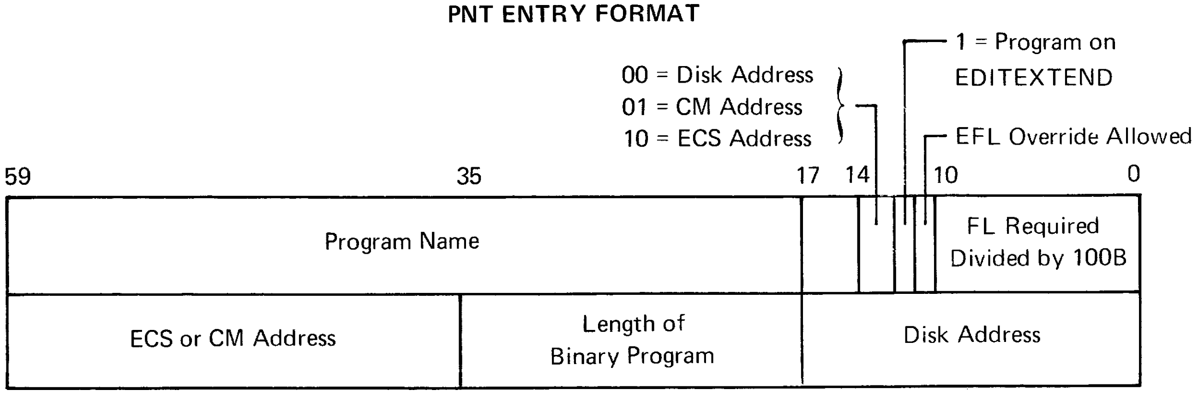
ERT ENTRY FORMAT

Entry Point Number + 1	Entry Point Number + 1	Entry Point Number + 1	Entry Point Number + 1	Entry Point Number + 1 or Continuation

PNUT ENTRY FORMAT

Parcel 0 Relative PNT Address	Parcel 1 Relative PNT Address	Parcel 2	Parcel 3	Parcel 4
Parcel 5	Parcel 6			Parcel n

PNT ENTRY FORMAT



INTERCOM POINTER AREA

	59	53	47	35	23	17	11	0
	Terminal/Central Site Communication Area							24
W.IEMBF	C.INBUF	Small buffer empty chain pointer						25
W.IEMBI	C.INBUF	Small buffer empty chain interlock table pointer		C.IINCT Interlock table length=N.1M1+2			C.IITYPE Interlock type = 0 0 0 1	26
W.IUSER	C.INBUF	LS user table chain pointer		C.INUSER Number of entries in chain				27
W.IUSEI	C.INBUF	LS user table interlock table pointer		C.IINCT N.1M1+2			C.IITYPE 0 0 0 1	30
W.IUSEW	C.IUWEST EST Ord.	Port	C.IUWSTA Station Address	E.IUWSTS 1=1CI has assigned user table	C.IUWUSR	Address of new user table		31
W.IUSWI	C.INBUF	User table request word interlock table pointer		C.ITNCT N.1M1+2			C.IITYPE 0 0 0 1	32
W.IMES	Est Ord	Port	Site Address	HS/LS user table chain index				33
W.IOUTI	C.INBUF	W.IOTPUT interlock table pointer (LS)		C.IINCT N.1M1+2			C.IITYPE 0 0 0 1	34
W.IINPI	C.INBUF	W.IINPUT interlock table pointer (LS)		C.IINCT N.1M1+2			C.IITYPE 0 0 0 1	35
W.IUSTI	C.INBUF	W.ITNPUT/W.ITOPUT interlock table pointer (LS)		C.IINCT N.1M1+2			C.IITYPE 0 0 0 1	36
W.IMUJ	C.INBUF	MUJ table chain pointer		C.INMUJ Current number of MUJs			C.IQPINT 1QP interlock	37
W.I1BR	C.IDROP		C.IPTIME Delay counter small buffer release code	C.ICTR Counter for "no CM obtained" message	C.IMM Time in minutes		C.ITOT Number of times message written	40
W.IT76	S.IT76-Interlock bit for INTERCOM Request Area in 6000 Station Control Point Area						C.IT76	41
W.IDSI	C.INBUF	Data stream interlock table pointer (HS/LCC)		C.IINCT N.1WB+N.1ZZ+2			C.IITYPE 0 0 0 3	42
W.IHSMT	C.INBUF	Large buffer empty chain pointer						43

C.IDROP (W.I1BR):

	53			48		
	LCC	HS	LS	LCC	HS	LS
	Drop complete			Drop in progress		

INTERCOM POINTER AREA

Word	59 53 47	35	23 17	11 0	
W.IHSMTI	C.INBUF Large buffer empty chain interlock table pointer	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 2	43
W.IHUSR	C.INBUF HS user table chain pointer	C.INUSER Number of entries in chain			44
W.ILCUSR	C.INBUF LCC user table chain pointer	C.INUSER Number of entries in chain			45
W.IHUSRI	C.INBUF HS/LCC user table interlock table pointer	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 4	46
W.IHOUTI	C.INBUF W.IOTPUT interlock table pointer (HS/LCC)	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 4	47
W.IHINPI	C.INBUF W.IINPUT interlock table pointer (HS/LCC)	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 4	50
W.IHTI	C.INBUF W.ITNPUT/W.ITOPUT interlock table pointer (HS/LCC)	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 4	51
W.ISTATI	C.INBUF Data Stream Status word interlock table pointer (HS/LCC)	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 4	52
W.IEMBAI	C.INBUF Small buffer empty chain auxiliary interlock table pointer	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 2	53
W.IUSWAI	C.INBUF User table request word auxiliary interlock table pointer	C.IINCT N.1WB+ N.1ZZ+1		C.IITYPE 0 0 0 2	54
W.IGUTC	C.INBUF Graphics user table chain pointer				55
W.IGRES	C.INBUF Graphics console reservation table pointer	C.IGEJC Ports for which GEJ Called	C.IGEJ No. of executing Copies of GEJ	C.IGDSF GEJ Deadstart Recovery Flag	56
W.IGSI	C.INBUF Graphics interlock word pointer	C.IINCT 0 0 0 1		C.IITYPE 0 0 0 1	57

Interlock Area

274 IGS Console Reservation Table

C.IGWT1 Address user table first job in queue	C.IGWT2 Address user table second queued job	C.IGUTA Address user table attached job	0
			N.IGCON-1

S.IGSNM–console in SIGNON mode

S.IGAV–console available in configuration

INTERLOCK TABLE FORMATS

Type 1 interlock table format

	59	11 0	
		Indicator	System & 1BR word
IP.1M1		Indicator	Low-speed driver words
		Indicator	
		Indicator	High-speed & LCC driver word

Indicator = any non-zero value

Type 2 interlock table format

	59	11 0	
		Indicator	System & 1BR word
IP.1WB		Indicator	High-speed driver words
		Indicator	
IP.1ZZ		Indicator	LCC driver words
		Indicator	

Indicator = any non-zero value

Type 3 interlock table format

	59	17 0	
		Indicator	System word
IP.1WB		Indicator	High-speed driver words
		Indicator	
IP.1ZZ		Indicator	LCC driver words
		Indicator	
		Indicator	1BR word

Indicator = 18 bit CM address of data stream word pair (in auxiliary high-speed user table)

Type 4 interlock table format

	59	17 0	
		Indicator	System & 1BR word
IP.1WB		Indicator	High-speed driver words
		Indicator	
IP.1ZZ		Indicator	LCC driver words
		Indicator	

Indicator = 18 bit CM address of user table

MULTI-USER JOB TABLE

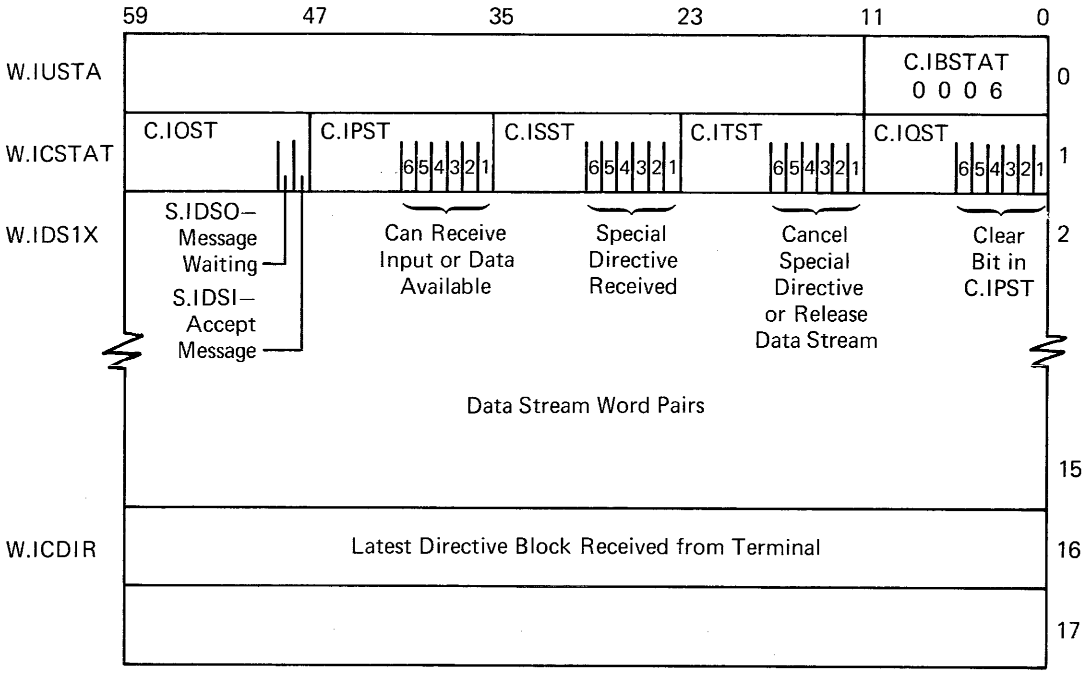
Word	59	47	35	23	11–0	
W.IUSTA	C.INBUF Address of Next MUJ Table		C.IUSID MUJ ID	C.ITIME 1CI Timer for Swaps	C.IBSTAT 0005	0
W.IMNAME	MUJ Name					1
W.IMQP	C.IMUC Total Number of MUJ Users	C.IMAC Activity Count	C.IMSTAT MUJ Status Byte	C.IMSIF Swapin Flag	C.IMSOF Swapout Flag	2
W.IUFST	C.IUJDA MUJ JDT Address				C.IUFILE — End of Execution	3
W.IMDES	C.IMED EDITOR Flag	C.IMSID Swapin Delay	C.IMSOD Swapout Delay	C.IMFL MUJ FL/100B		4
W.IMTIN	C.IMTIP Address of TERMIN		C.IMTIL Length of TERMIN	C.IMTHDR Address of TERMIN/TERMOUT Header		5
W.IMTOUT	C.IMTOP Address of TERMOUT		C.IMTOL Length of TERMOUT	C.IMTHDR Address of TERMIN/TERMOUT Header		6
	(Reserved for CDC)					7–15
	(Reserved for Installation)					16–17

W.IMQP(2)	C.IMSTAT
	0001 = Waiting for I/O
	0003 = MUJ Active

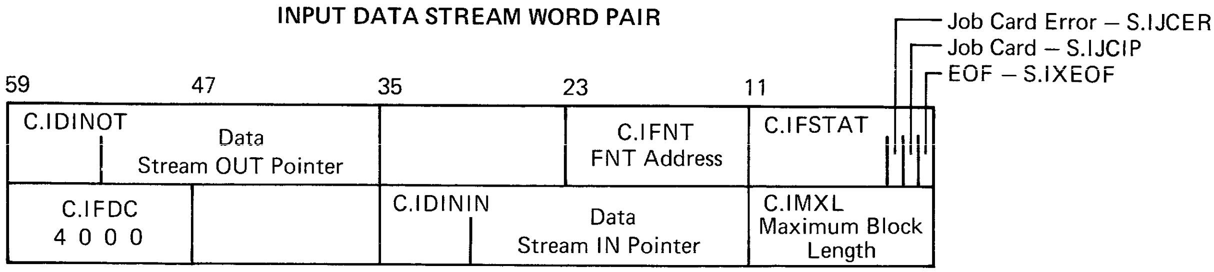
HIGH SPEED USER TABLE

	59	47	35	23	11 0	
W.IUSTA	C.INBUF Address of Next HS User Table		C.IUSID Terminal ID		C.IBSTAT 0 0 1 0	0
W.IFLGS W.IJBCRD	C.IDVRT C.IOVFL C.IMESCI		C.IJBCRD Pointer to Job Card		C.ICADS Number of Data Streams	1
						2
						3
W.IINPUT	C.IDINOT Incoming Message OUT Pointer		C.IDININ Incoming Message IN Pointer			4
W.IOTPUT	C.IDOTOT Outgoing Message OUT Pointer		C.IDOTIN Outgoing Message IN Pointer		C.IMXL Max. Buffer Size	5
W.ISDOT	C.IDINOT Special Directives OUT Pointer					6
W.ISDIN			C.IDININ Special Directives IN Pointer			7
W.IUEQP	C.IUEST/C.IUPORT EST Ord / Port Number	C.IGSON	C.IDAUT Address of Aux. User Table		C.IUMSG 0 0 1 0	10
W.IUSTAT	1WB Temporary Storage				C.IUSTAT Terminal Status	11
W.IUDRV2	C.ISYNC Number of Synch Errors	C.ICYCL No. of Cyclic Code Errors	C.IIO Number of I/O Errors	C.IDSEQ No. of Sequence Bit Errors	C.ITOTRX Number of Retransmissions	12
	(Reserved for CDC)					13 – 16
W.IINS	(Reserved for Installation)					17

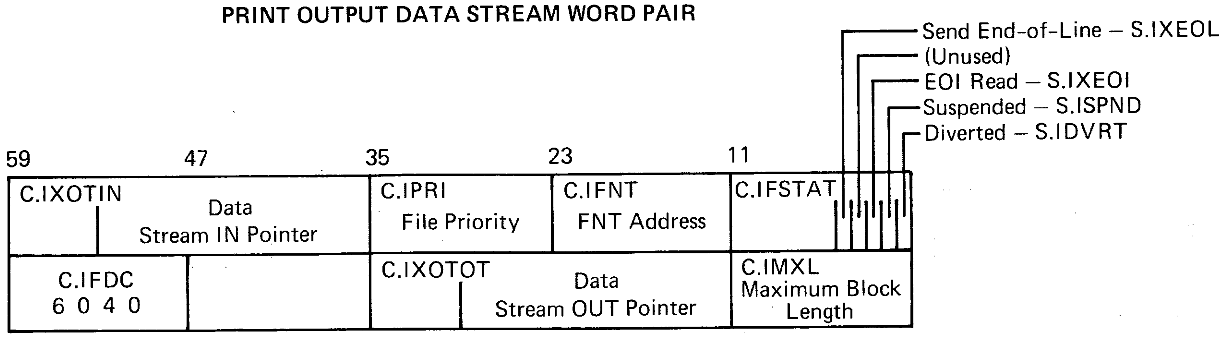
AUXILIARY HIGH SPEED USER TABLE



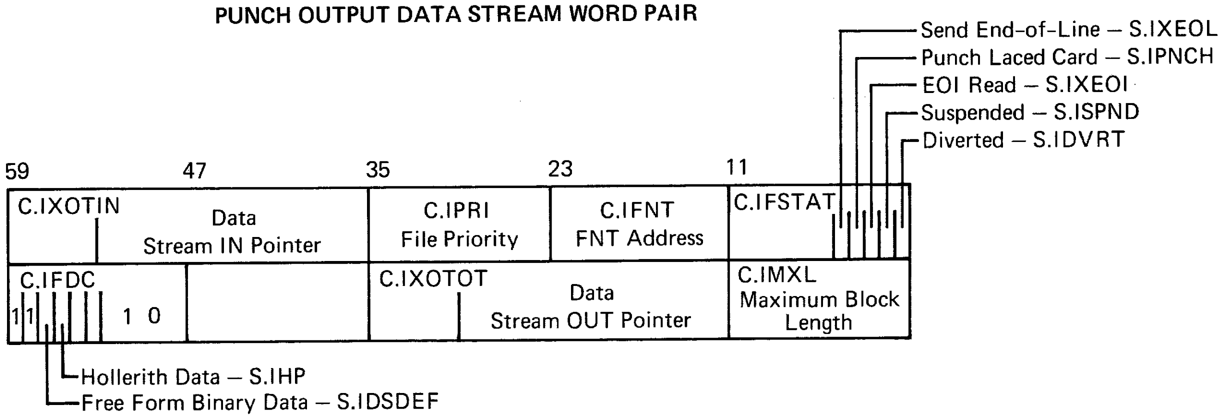
INPUT DATA STREAM WORD PAIR



PRINT OUTPUT DATA STREAM WORD PAIR



PUNCH OUTPUT DATA STREAM WORD PAIR



NOTES: HIGH SPEED USER TABLE

W.IFLGS		**C.IDVRT/C.IOVFL/C.IMESCI**	
Bits	0-5		10 = divert punch output
			40 = divert print output
			50 = divert print and punch output
	6	S.IMAX	Maximum message buffers assigned or no available empty buffers
	7	S.IOVFL	Generated message lost; no available empty buffer
	8-11		(Unused)
W.IUSTAT		**C.IUSTAT**	
Bits	0	S.IXDRP	Terminal shut down
	1	S.IXACK	Drop request acknowledgement
	2	S.IXUP	Start of terminal activity
	3	S.IXOFF	Terminal terminated communication
	4	S.IXUTBY	Release to empty chain
	5-11		(Unused)
W.IUEQP		**C.IGSON**	
Bits	0-5		Graphics output stream number
	6	S.IGRMUX	Graphics MUX
	7	S.IGSON	Graphics streams defined
	8-11		(Unused)

LOW SPEED USER TABLE

Word	59–48	47–36	35–24	23–12	11–0	Word No.	Notes
W.IUSTA	C.INBUF Address of Next LS User Table		C.IUSID User ID	C.ITIME Timer	C.IBSTAT Flags 0 0 0 1	0	
W.IUCMD	C.ICMD Batch Command	C.IPARAM Parameter	C.IJBCRD Pointer to Job Card		C.IUSTAT Flags	1	
W.IUMUJ	C.IUMJS MUJ Status	C.IUMJP MUJ Table Pointer			C.IUMORD MUJ Ordinal	2	
W.IUFST	C.IUJDA JDT Address				C.IUFILE		Executing Command
	C.IUEQC Swap File Equipment Code	C.IUFRB Swap File First RBT Word Pair	C.IUFL Swap File Length/100B	C.IUPFL Positive FL/100B	File Count and Flags	3	Command or Transmission Mode or Attached to MUJ
W.IINPUT	C.IDINOT I/L Input Data OUT Pointer		C.IDININ 0 0 Input Data IN Pointer		C.ISIZLN Page Size	4	
W.IOTPUT	C.IDOTOT 0 0 Output Data OUT Pointer		C.IDOTIN I/L Output Data IN Pointer		C.ISIZLN Line Length	5	
W.ITNPUT	C.IDINOT No. Buffers Transmission Input Data OUT Pointer		C.IDININ 0 0 Transmission Input Data IN Pointer		C.IFSTAT Status	6	
W.ITOPUT	C.IDOTOT 0 0 Transmission Output Data OUT Pointer		C.IDOTIN Flags Transmission Output Data IN Pointer		C.IFSTAT Status	7	
W.IUEQP	S.IUEST/S.IUPORT EST Ord. Port	C.IUSITE Site Address	C.IIFNT FNT Pointer	C.IOFNT FNT Pointer	C.IUMSG Message Ordinal	10	
W.IUDRV1	C.IUDRET Return Address	C.IUDCNT No.of Spaces Left	C.IUDSTA Station Address	C.IUDECD Retry e-Code	C.IDSTAT C.IUDCCD	11	6671 CRT
	Driver Parameters						6676 TTY
W.IUDRV2	Partially Transmitted Word of Data					12	6671 CRT
	Current Data Word Being Assembled/Disassembled						6676 TTY
W.IUAFT	C.ICUFL Current FL/100B	C.IMXFL Max. FL/100B	C.ICUTL Current Time Limit	C.IMXTL Maximum Time Limit	C.IMXFI Maximum Files	13	
W.IUAPP	C.IUACS Access Level	C.IUFLGS User Table Flags	PP Time as Number of Seconds * 4096			14	
W.IUACP / W.IUIUP	C.IUCCA/C.IUIUP Flags Control Card Buffer		CP Time as Number of Seconds * 4096			15	
W.IUATIM		Date of Login Y Y D D D		Time of Login H H M M		16	
W.IINS	Reserved for Installations or Last Command for DSD Q–Display					17	

NOTES: LOW SPEED USER TABLE

W.IUSTA(0)		**C.IBSTAT**	
Bit	4	S.IDISC	Terminal Disconnected
	5-6	S.ILOGO	00 = user logged out
			01 = user logged in
			10 = auto-logout requested
			11 = auto-logout in progress
	7-8	S.ISTATE	00 = transmission state
			01 = waiting input
			10 = waiting output
			11 = active or assigned to control point
	9	S.IRDIS	Request disconnect
	10	S.IUTAPE	Request paper tape reading
	11	S.IABRT	Abort request

W.IUCMD(1)	**C.ICMD**	**C.IPARAM**
0	No command	
1	REW	
2	BSP,nnn	PRU count to backspace
3	CONTINUE	
4	END,CR	
5	END,LP	
6	GO	
7	DIVERT	User ID
10	READ	FNT address
11	SUSPEND	
12	REP,n	
13	No command - MES active	
14	No command - attached to MUJ	
15	No command - flag for 1CI	Parameter for 1CI overlay

W.IUCMD(1)		**C.IUSTAT**	
Bit	0-5		Used by IDS to determine type of H display wanted
	6	Not used	
	7	Not used	
	8	Not used	
	9	Not used	
	10	S.IMPORT	Import on flag
	11	S.ICTAPE	Paper tape on flag

W.IUMUJ(2)		**C.IUMJS**	
Bit	0	S.IMNUS	New user
	1	S.IMRUN	RUN command in progress
	2	S.IMDIS	Reconnected after disconnect
	3-6		(Unused)
	7	S.IMBKS	Break sent
	8	S.IMLGS	Logout sent
	9	S.IMWO	Waiting for output to complete
	10	S.IMWI	Waiting for input
	11	S.IMUJ	Attached to MUJ

NOTES: LOW SPEED USER TABLE (CONT'D)

W.IUFST(3)		**C.IUFILE**	
Bits	0-5		Count of FNT entries in swap file
	6	S.IUEOE	End of execution
	7	S.IUDMP	SAVEFL flag
	8	S.IUPS	Pause bit
	9	S.IUECS	Swap file on ECS
	10	S.IUFNT	FNT to be associated on next swap
	11	S.IURED	REDUCE flag

W.ITNPUT(6)		**C.IFSTAT**	
Bit	0	S.ITMIT	Read cards now
	1	S.ITFIL	0 = reading jobs
			1 = reading named file
	2-4	S.ITERR	001 = reader not ready
			010 = input file error
			011 = card read error
			100 = job card error
	5	S.IEOF	End-of-file card read
	6	S.IFEOF	Stop at end of current message
	7	S.IREAD	READ command issued
	8-11	S.ICRDS	Number of words of current card already written

W.ITOPUT(7)		**C.IDOTIN**
Bits	10-6	Count of data words in output buffer past IN
	11	1 = at least a partial word past IN

W.ITOPUT(7)		**C.IFSTAT**	
Bit	0	S.ITMIT	Send data
	1	S.ITDEV	0 = printer to receive
			1 = CRT to receive
	2-4	S.ITERR	001 = printer not ready
			011 = PM message flag
	5	S.IEOI	EOI reached
	6	S.IENDLP	END,LP issued
	7	S.ITSSP	SUSPEND active
	8	S.ITACT	Printing to be done

W.IUEQP(10)		**C.IUMSG**	
Bit	0-5	S.IUMSG	001_2 = BCD 200 UT
			010_2 = TTY
			100_2 ANSI 200 UT
	10	S.ILS	4800 baud UT
	11	S.ISA	Hardwired terminal

NOTES: LOW SPEED USER TABLE (CONT'D)

W.IUDRV1(1) **C.IUDSTA**

Bits	0-1	Station address ordinal received
		00 = address 141
		01 = address 140
		10 = address 161
		11 = address 160
	2-5	Zero
	6-9	Send message type
		0 = blanks to fill display input line
		1 = display output line
		2 = E3 write to request card input from UT
		3 = data to UT printer
		4 = send clear-write message after page wait
		10 = poll
	9-10	Sending station address ordinal
	11	(Unused)

W.IUDRV1(11) **C.IDSTAT/C.IUDCCD**

Bits	0-5		0 = read
			1 = acknowledge
			2 = reject
			3 = error
	6	S.IUDPAG	Page wait flag
	7	S.IVEROR	Transmission error on reply
	8	S.IUBDCH	Bad character received
	9	S.IUBDWR	Check for previous bad write
	10		(Unused)
	11	S.IUNODT	No data expected

W.IUAPP(14) **C.IUFLGS**

Bits	0-5	S.IUSSW	Sense switch setting changes
	6-9		(Unused)
	10	S.ITLOK	Lock bit
	11		Login with unrestricted password

W.IUACP/W.IUIUP(15) **C.IUCCA/C.IUIUP**

Bit	6	S.IUCCP	Control cards moved to control point area
	7-10		(Unused)
	11	S.IUEDC	Buffer contains control cards sent from EDITOR

274 IGS USER TABLE

	59 53	47	35	23	11 0	
W.IUSTA	C.INBUF	Address next user table in chain	C.IUSID User id		C.IBSTAT Flags 7 0 1 1 1	0
						1
						2
W.IUFST	C.IUIDA	Job descriptor addr.	C.IUCLAS Job class before entering IGS queue			3
W.IINPUT	C.IDINOT	Graphics input data out pointer	C.IDININ	Graphics input data in pointer		4
W.IOTPUT	C.IDOTOT	Graphics output data out pointer	C.IDOTIN	Graphics output data in pointer		5
W.IUGCAC	C.IUGCAC Console Access Flag	C.IUGEF Err. Flag Saved by GEJ	C.IUGEN Err. Message Console No.			6
						7
W.IGHSU W.IUEQP	C.IUPORT Est ordinal / Port no.	C.IGFLGS	C.IGHSU	Pointer to high speed user table	C.IUMSG 0 0 1 0	10
						11
W.IUACP	C.IUCCA	Control Card Buffer Address				15
						17

C.IGFLGS bits:

- S.IG3TT 3TT interlock flag
- S.IGQUE flag job queued for first console
- S.IGSNON SIGNON user table flag
- S.IGDRP IGS job termination flag
- S.IGSNIT SIGNON initialization
- S.IGCON console number for which job is queued

LCC USER TABLE

Word	59–48	47–36	35–24	23–12	11–0	
	59 53	47	35 29	23	11 3 0	
W.IUSTA	C.INBUF 00 | Address of Next LCC User Table		C.IUSID User ID	C.ITIME Timer	C.IBSTAT Flags | 1001	0
W.IUCMD / W.IFLGS / W.IJBCRD	C.ICMD	C.IPARAM Parameter	C.IJBCRD | Pointer to Job Card		C.IUSTAT Flags	1
W.IUMUJ	C.IUMJS MUJ Status	C.IUMJP | MUJ Table Pointer			C.IUMORD MUJ Ordinal	2
W.IUFST	C.IUJDA | JDT Address				C.IUFILE File Count and Flags	3
	C.IUEQC Swap File Equipment Code	C.IUFRB Swap File First RBT Word Pair	C.IUFL Swap File Length/100B	C.IUPFL Positive FL/100B		
W.IINPUT	C.IDINOT I/L | Input Data OUT Pointer		C.IDININ 00 | Input Data IN Pointer		C.ISIZLN Page Size	4
W.IOTPUT	C.IDOTOT 00 | Output Data OUT Pointer		C.IDOTIN I/L | Output Data IN Pointer		C.ISIZLN Line Length	5
W.ISDOT	C.IDINOT | Command OUT Pointer		C.IDININ | Command IN Pointer		C.IDVTID Destination ID for DIVERT,LP	6
W.ISDIN					C.IDVTID Destination ID for DIVERT,CP	7
W.IUEQP	C.IUEST/C.IUPORT EST Ord | Line Number		C.IUAUT | Address of Aux. User Table		C.IUMSG Terminal Type	10
W.IUDRV1	C.ISTST Interactive stream output state | input state	C.IBCNT | Interactive input stream byte count	C.IDININ C.IBPOS		C.ILNCNT | Current line no.	11
W.IUDRV2	Batch Stream States					12
W.IUAFT	C.ICUFL Current FL/100B	C.IMXFL Max. FL/100B	C.ICUTL Current Time Limit	C.IMXTL Maximum Time Limit	C.IMXFI Maximum Files	13
W.IUAPP	C.IUACS Access Level	C.IUFLGS User Table Flags	PP Time as Number of Seconds * 4096			14
W.IUACP / W.IUIUP	C.IUCCA/C.IUIUP Flags | Control Card Buffer		CP Time as Number of Seconds * 4096			15
W.IUATIM	| Date of Login Y Y D D D			Time of Login H H M M		16
W.IINS	Reserved for Installations or Last Command for DSD Q-Display					17

Executing Command

Command or Transmission Mode or Attached to MUJ

NOTES: LCC USER TABLE

W.IUSTA(0)		**C.IBSTAT(4)**		
Bits	0-3	S.ITYPE	1001 LCC User Table Type	
	4	S.IDISC	Terminal disconnected	
	5-6	S.ILOGO	00 = user logged out	
			01 = user logged in	
			10 = auto-logout requested	
			11 = auto-logout in progress	
	7-8	S.ISTATE	01 = waiting input	
			10 = waiting output	
			11 = active or assigned to control point	
	9	S.IRDIS	Request disconnect	
	10	S.IUTAPE	Request paper tape reading	
	11	S.IABRT	Abort request	
W.IFLGS(1)		**C.ICMD(0)**		**C.IPARAM(1)**
Bits	0-11	0 No command		
		13 No command-MES active		
		14 No command-attached to MUJ		
		15 No command-flag for 1CI		Parameter for 1CI overlay
W.IFLGS(1)		**C.IUSTAT(4)**		
Bits	0-5	Used by 1DS to determine type of H display wanted		
	6	S.IDVTCP	Divert all CP files	
	7	S.IDVTLP	Divert all LP files	
	8	S.IOFFL	Offline	
	9	S.ILUTBY	Set by 1LX; user table can be released	
	11	S.ICTAPE	Paper tape on flag	
W.IUMUJ(2)		**C.IUMJS(0)**		
Bit	0	S.IMNUS	New user	
	1	S.IMRUN	RUN command in progress	
	2	S.IMDIS	Reconnected after disconnect	
	3-6		(unused)	
	7	S.IMBKS	Break sent	
	8	S.IMLGS	Logout sent	
	9	S.IMWO	Waiting for output to complete	
	10	S.IMWI	Waiting for input	
	11	S.IMUJ	Attached to MUJ	
W.IINPUT(4)		**C.ISIZLN(4)**		
Bits	0-11		No. of lines/page	
			= 7777B TTY	
			= 14D HSBT with 16x80 screen	
			= 16D HSBT with 18x64 screen	

NOTES: LCC USER TABLE (CONT'D)

W.IOTPUT(5)		**C.ISIZLN(4)**	
Bits	0-11		No. of characters/line = 72D TTY = 80D HSBT with 16x80 screen = 64D HSBT with 18x64 screen

W.IUEQP(10)		**C.IUAUT(2)**
Bits	0-17	Auxiliary user table pointer
	23	=0 Auxiliary user table available
		=1 Auxiliary user table not available
W.IUEQP(10)		**C.IUMSG(4)**
Bits	0-11	Terminal type
		=2 TTY
		=22B HSBT
W.IUDRV1(11)		**C.IBCNT(1)**
Bits	0-10	Interactive input stream byte count
	11	=0 Current input line terminated with CR
		=1 Current input line terminated with LF
W.IUDRV1(11)		**C.IDININ(2)**
Bits	0-17	=0 previous input line terminated with CR
		≠0 IN pointer for previous input line terminated with LF
W.IUDRV1(11)		**C.IBPOS(2)**
Bits	6-8	Byte position for previous input line terminated with LF
	9	=0 Store next input character in right half of byte
		=1 Store next input character in left half of byte
	10	=0 Last operation was input
		=1 Last operation was output
	11	=1 Teletype output was interrupted
W.IUDRV2(12)		**Byte 0(0)**
Bits	0-5	Batch stream state – stream no 5
	6-11	Batch stream state – stream no 4
W.IUDRV2(12)		**Byte 1(1)**
Bits	0-5	Btach stream state – stream no 7
	6-11	Batch stream state – stream no 6
W.IUDRV2(12)		**Byte 2(2)**
Bits	0-5	unused
	6-11	Batch stream state – stream no 8

NOTES: LCC USER TABLE (CONT'D)

W.IUDRV2(12)		**Byte 3(3)**	
Bits	0-5	unused	
	6-11	Batch stream state – stream no 10	
W.IUDRV2(12)		**Byte 4(4)**	
Bits	0-5	unused	
	6-11	Batch stream state – stream no 12	
W.IUAPP(14)		**C.IUFLGS(1)**	
Bits	0-5	S.IUSSW	Sense switch setting changes
	6-9		Unused
	10	S.ITLOK	Lock bit
	11		Login with unrestricted password
W.IUACP(15)		**C.IUIUP(0)**	
Bit	6	S.IUCCP	Control cards moved to control point area
	7-10		Unused
	11	S.IUEDC	Buffer contains control cards sent from EDITOR

AUXILIARY LCC USER TABLE

No. of data streams defined and ON

		59	53	35	23		11	3 0
W.IUSTA	0	C.INBUF 0 0	0 0 0 0 0 0	C.IUSID Terminal ID	C.ILCADS		C.IBSTAT	0110
W.ICSTAT	1	Last Stack Request Response						
W.IDS1X	2	Data Stream Word Pairs						
	3							
W.IDS2X	4							
	5							
W.IDS3X	6							
	7							
W.IDS4X	10							
	11							
W.IDS5X	12							
	13							
W.IDS6X	14							
	15							
W.IDS7X	16							
	17							

OUTPUT DATA STREAM WORD PAIR

	59 56 53	47	35 29	23	11 0
W.IDS-X	C.IXOTIN Byte Pos.	Output IN Pointer	C.IFCM Special Disposition Code	C.IFNT FNT Address	C.IFSTAT
W.IDS-X+1	C.IFDC Disposition		C.IXOTOT C.IBPOS Byte Pos.	Output OUT Pointer	C.IFSTAT

NOTES: LCC OUTPUT DATA STREAM WORD PAIR

W.IDS-X		**C.IFSTAT(4)**	
Bit	0	S.IOFF	Set/cleared by 1LX to request 1ZZ off/on stream
	1	S.IXABT	Set by 1LX to request 1ZZ abort stream
	2	S.ISTOP	Set by 1LX to request 1ZZ stop processing stream
	3	S.IETX	Set by 1LX to request 1ZZ sent ETX block
	4		Unused
	5	S.IEOL	End-of-line flag
	6	S.IXNDLP	End flag
	7	S.ISUP	Suppress-carriage-control flag
	8	S.IBAN	No-banner flag
	9	S.IHDR	Header flag
	10	S.IWEOJ	Wait-end-of-job flag
	11	S.IWAITX	Wair-for-driver flag
W.IDS-X+1		**C.IFDC(0)**	
Bits	0-5		Disposition code
	5	S.ILP	Print file
	6	S.ISNT	Banner/laced card not sent
	7-9	S.IDC	Mode code
	9	S.IPBIN	Punch binary
	10-11		Unused
W.IDS-X+1		**C.IFSTAT(4)**	
Bit	0	S.IOFF	Set/cleared by 1ZZ to indicate stream off/on
	1	S.IXABT	Set by 1ZZ to indicate stream aborted
	2	S.ISTOP	Set by 1ZZ to indicate stream stopped
	3	S.IETX	Set by 1ZZ to indicate ETX block sent
	4	S.IXABTI	Set by 1ZZ to indicate abort issued
	5		Unused
	6-11	S.IDSN	Stream no.

INPUT DATA STREAM WORD PAIR

	59 56 53 47	35	29 23	11 0
W.IDS-X	C.IDINOT Byte Pos. Input OUT Pointer	C.IRPC Compression Count	C.IFNT FNT Address	C.IFSTAT
W.IDS-X+1	C.IBCNT Byte Count	C.IDININ C.IBPOS Byte Pos. Input IN Pointer		C.IFSTAT

0 = ETX Block Received
1 = ETB Block Received

NOTES: LCC INPUT DATA STREAM WORD PAIR

W.IDS-X		**C.IFSTAT(4)**	
Bit	0	S.IOFF	Set/cleared by 1LX to request 1ZZ off/on stream
	1	S.IXABT	Set by 1LX to request 1ZZ abort stream
	2	S.ISTOP	Set by 1LX to request 1ZZ stop processing stream
	3		Unused
	4		Unused
	5	S..IXEOF	End-of-file flag
	6	S.IJCIP	Job-card-in-progress flag
	7	S.IJCER	Job-card-error flag
	8	S.IASC	ASCII mode
	9	S.IBIN	Binary mode
	10	S.IWEOJ	Wait-end-of-job flag
	11	S.IWAITX	Wait-for-driver flag

W.IDS-X+1		**C.IFSTAT(4)**	
Bit	0	S.IOFF	Set/cleared by 1ZZ to indicate stream off/on
	1	S.IXABT	Set by 1ZZ to indicate stream aborted
	2	S.ISTOP	Set by 1ZZ to indicate stream stopped
	3	S.IETX	Set by 1ZZ to indicate ETX block received
	4	S.IXABTI	Set by 1ZZ to indicate abort issued
	5		Unused
	6-11	S.IDSN	Stream no.

RECORD BLOCK TABLE ENTRY

First RBT Word Pair:

59	47	S.RBTRBR	35 29	23	11 0
C.RBTWPL Next Word Pair	C.RBTRBR C.RBTFB RBR Ordinal	3	C.RBTAL Alloc. Type	C.RBTPRU Last PRU + 1	C.RBTBIT Flags
RB3	RB4		RB5	RB6	RB7

Other Word Pairs:

59	47		35	23	11 0
C.RBTWPL Next Word Pair	C.RBTRBR C.RBTFB RBR Ordinal	0	RB0	RB1	RB2
RB3	RB4		RB5	RB6	RB7

C.RBTBIT

Bit	Field
11	S.RBTDEV — Assigned by device type
10	S.RBTEST — Assigned by EST ordinal
9	S.RBTPFD — PF device assigned
8	S.RBTNEW — RBT new
7	S.RBTOVF — Overflow
6	S.RBTRND — Random
5	FO = IS/DA — S.RBTSAM
4	(Reserved)
3	(Reserved)
2	(Reserved)
1	(Reserved)
0	(Reserved)

RECORD BLOCK TABLE BYTE

(Computation of physical addresses for default allocation styles)

<table>
<tr><th></th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>814</td><td colspan="12">Track Number</td></tr>
<tr><td>6603</td><td></td><td>Z</td><td colspan="7">Head Position</td><td colspan="2">HG</td><td>O/E</td></tr>
<tr><td>6638</td><td colspan="6">Head Position</td><td>S</td><td colspan="4">Head Group</td><td>O/E</td></tr>
<tr><td>865</td><td colspan="3">000</td><td>O/E</td><td colspan="6">Head Group</td><td colspan="2">SG</td></tr>
<tr><td>854</td><td colspan="11">Track Number</td><td></td></tr>
<tr><td>821</td><td colspan="12">N</td></tr>
<tr><td>841</td><td colspan="12">N</td></tr>
</table>

Z	Zone	0 = Outer 1 = Inner
HG	Head Group	
O/E	Odd/Even	0 = Even 1 = Odd
S	Stack	
SG	Subgroup	

821 Cylinder = $\frac{N-1}{2}$

Track = 0

Sector = 0 if N-1 is even; 1 if N-1 is odd

841 Cylinder = $\frac{N-1}{5}$

Track = 8 * remainder of $\frac{N-1}{5}$ modulo 20

Sector = 0 if remainder of $\frac{N-1}{5} \leq 2$; 1 if remainder of $\frac{N-1}{5} > 2$

SECTION 2

JOB CONTROL POINT TABLES

RA COMMUNICATION AREA

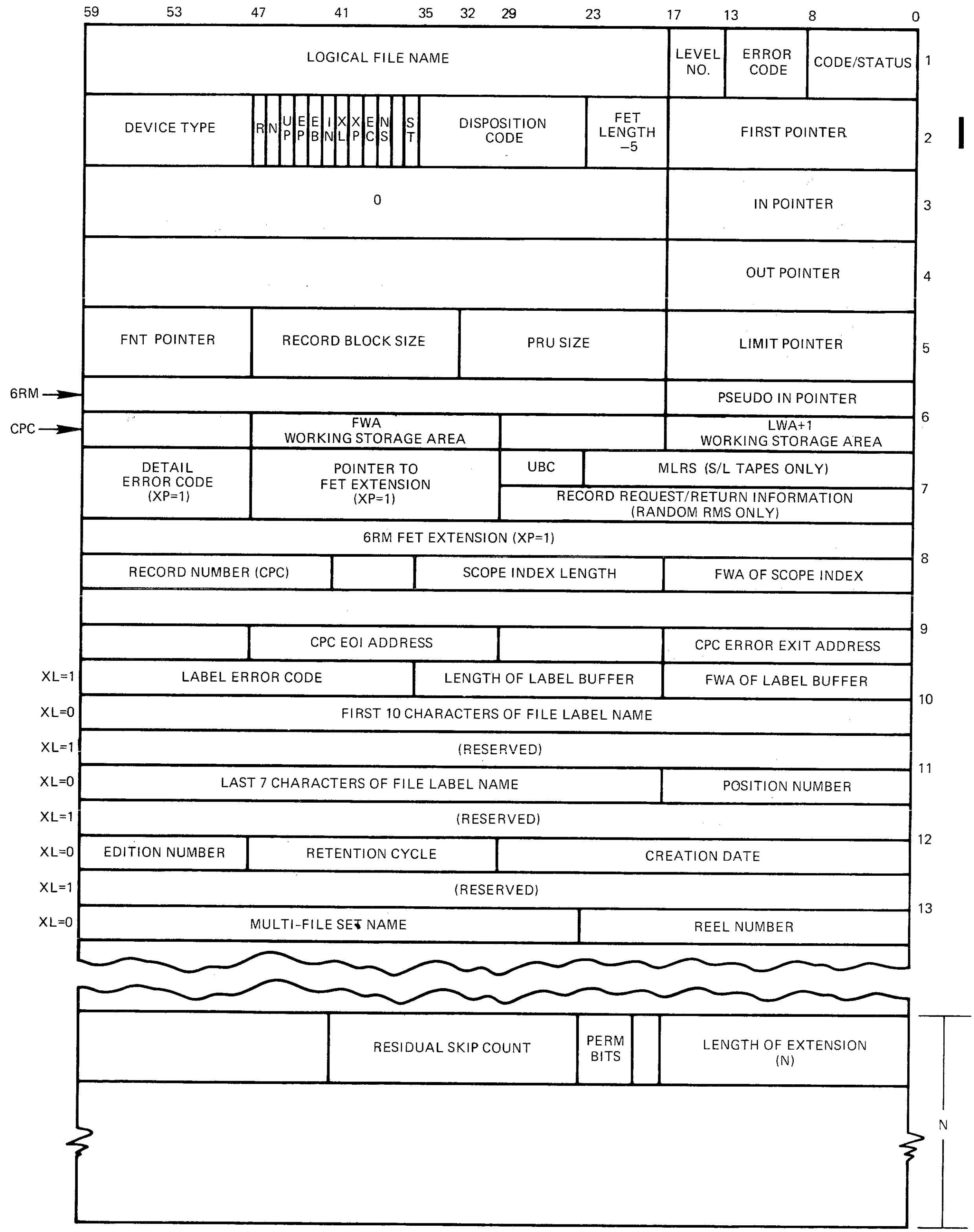
Bit positions: 59, 35, 29, 23, 17, 11, 5, 0

Word	Fields (bit 59 → bit 0)
RA+0	(blank) | R | (blank) | T | P | SS | SL
RA+1	User/System Interface
RA+2 … RA+53	Parameter | (Reserved) | Code
RA+54 … RA+63	1AJ Bootstrap for Absolute Programs
RA+64	Name/Library Name | Number of Parameter Words, Beginning in RA+2
RA+65	M | LWA+1 of Loadable Area in ECS | (blank) | L | LWA+1 of Loadable Area in CM
RA+66	X | FWA of Loadable Area in ECS | (blank) | D | FWA of Loadable Area in CM
RA+67	(blank) | C | (blank)
RA+70 …	Control Statement Card Image (Replaced by Operator Message if CFO Type-in)
RA+77	Run 2.3 ASA Flag +0 = Non-ASA; -0 = ASA
	User Program

NOTES: RA COMMUNICATION AREA

R	Job dependency recheck bit
T	Storage move flag (1 = move being attempted)
P	Pause flag (1 = control point pausing)
SS	Sense switches
SL	Sense lights
CODE	00 = Continuation
	01 = Comma
	02 = Equals sign
	03 = Slash
	04 = Left parenthesis
	05 = Plus sign
	06 = Minus sign
	07 = Blank
	10 = Semi-colon
	11 =
	12 =
	13 = (reserved)
	14 =
	15 =
	16 = Other
	17 = Termination
L	Library/file flag (1 = name is library name)
X	XJ flag: if XJ = 1, and XJ can be issued
C	LDV completion flag (bit 29)
D	DIS RSS flag (bit 18)
M	CMU Bit

FILE ENVIRONMENT TABLE



SCOPE CIO CODES IN OCTAL

All codes are shown for coded mode operations; add 2 for binary mode. Example: 010 is coded READ, 012 is binary READ.

000	RPHR	170	CLOSE,UNLOAD
004	WPHR	174	CLOSE,RETURN
010	READ	200	READC
014	WRITE	204	WRITEC
020	READSKP	210	READLS
024	WRITER	214	REWRITE
030	-	220	RESERVED FOR KRONOS READEI
034	WRITEF	224	REWRITER
040	BKSP	230	-
044	BKSPRU	234	REWRITEF
050	REWIND	240	SKIPF
054	-	244	-
060	UNLOAD	250	READNS
064	-	254	-
070	RETURN	260	READN
074	-	264	WRITEN
100	OPEN,NR	270-274	-
104	OPEN WRITE,NR	300	OPEN,NR
110	POSMF	304-324	-
114	EVICT	330	CLOSER
120	OPEN,NR	334	-
124	-	340	OPEN
130	CLOSE,NR	350	CLOSER
134	-	354-364	-
140	OPEN	370	CLOSER,UNLOAD
144	OPEN WRITE	374-474	-
150	CLOSE	500-574	RESERVED FOR INSTALLATIONS
154	-	600-634	-
160	OPEN	640	SKIPB
164	-	644-774	-

LOCAL SCRATCH FILE NAMES

ZZZZZ01	(EDITLIB)	Reset file
ZZZZZ02	(EDITLIB)	Restore file
ZZZZZ03	(EDITLIB)	System extend file
ZZZZZ04	(EDITLIB)	System file
ZZZZZ05	(EDITLIB)	Interpreted directives
ZZZZZ06	(EDITLIB)	ECS resident routines library file
ZZZZZ07	(EDITLIB)	Entry point name table spill file
ZZZZZ08	(EDITLIB)	Program number table spill file
ZZZZZ09	(DIAXNOS)	
ZZZZZ10		Program name table spill file
ZZZZZ11		External reference table spill file
ZZZZZ12		External reference collection spill file
ZZZZZ13	(EDITLIB)	Library or deadstart program collection file
ZZZZZ14		Scratch
ZZZZZ15		PP program name table spill file
ZZZZZ16		Library name table spill file
ZZZZZ17	(LOADER)	Entry point list
ZZZZZ18	(LOADER)	Owned global blocks
ZZZZZ19	(FORM)	
ZZZZZ1A-1Z	(SORT/MERGE)	
ZZZZZ20	(FORM)	
ZZZZZ21	(FORM)	
ZZZZZ22	(6RM)	
ZZZZZ23	(EDITLIB)	Current directory file
ZZZZZ24	(QUERY/UPDATE)	
ZZZZZ25	(LOADER)	Global library set
ZZZZZ26	(GRAPHICS)	
ZZZZZ27	(LOADER)	Overlay/segment generator
ZZZZZ28	(DEBUGGING AIDS)	
ZZZZZ29	(LOADO)	ECS hold file
ZZZZZ2A-2Z	(SORT/MERGE)	
ZZZZZ30	(LOADER)	SEGBILD scratch file (random)
ZZZZZ31	(LOADER)	SEGBILD sort file (random)
ZZZZZ32	(LOADER)	SEGBILD sort file (random)
ZZZZZ3A-3Z	(SORT/MERGE)	
ZZZZZ41-47	(COBOL)	
ZZZZZ50-59	(PL1)	
ZZZZZAA-A9	(Index Processor)	
ZZZZZBA-B0	(Index Processor)	
ZZZZZC3	(CDC Special Systems)	
ZZZZZC4	(CDC Special Systems)	
ZZZZZCP	(INTERCOM)	Copy permanent files
ZZZZZDF	(6RM/LOADER)	
ZZZZZEF	(6RM)	Error message file
ZZZZZI1	(ALGOL)	Scratch file
ZZZZZI2	(ALGOL)	Scratch file
ZZZZZIN	(PAGE Utility)	
ZZZZZOP	(FTN 4.0/COMPASS)	
ZZZZZOU	(PAGE Utility)	
ZZZZZQU	(QUERY/UPDATE)	

LOCAL SCRATCH FILE NAMES (CONT'D)

ZZZZZPA	(PFM)	Scratch file
ZZZZZPB	(PFM)	Scratch file
ZZZZZPC	(PFM)	Attached RBTC
ZZZZZPD	(PFM)	Attached PFD
ZZZZZPE	(PFM)	Reserved
ZZZZZPF	(PFM)	Attached PF
ZZZZZPG	(PFM)	Reserved
ZZZZZPT	(PFM)	PF dump tape
ZZZZZPW	(PFM)	Attached PF DUM
ZZZZZRE	(INTERCOM)	Restricted passwords
ZZZZZRL	(FTN 4.0/COMPASS)	
ZZZZZRM	(FTN 4.0/COMPASS)	
ZZZZZRN	(PAGE Utility)	
ZZZZZSA-SD	(SIFT)	
ZZZZZSE	(EDITOR/SETUP)	
ZZZZZSF	(SETUP)	
ZZZZZUN	(INTERCOM)	Unrestricted passwords
ZZZZZVx-Zx	(Installations)	

ENTRY POINT NAMES

AGxxxxx, ALxxxxx	ALGOL
ATxxxxx	APT
BAxxxxx	BASIC
CBxxxxx, COxxxxx	COBOL
CPxxxxx	COMPASS
D.xxxxx	COBOL
DIxxxxx	CE Diagnostics
EBxxxxx	8231 IMPORT
ECxxxxx	EXPORT IMPORT 200
EHxxxxx	6000 EXPORT High Speed
FExxxxx, FXxxxxx	FORTRAN Extended
FMxxxxx	FORM
FTxxxxx	RUN
G6xxxxx	IGS/6000 EXPORT HS
G7xxxxx	IGS/1700 IMPORT
INxxxxx	INTERCOM
IXxxxxx	Index Processor
ISxxxxx	SIS 1.0
ITxxxxx	INTERCOM
I7xxxxx	1700 IMPORT HS
I8xxxxx	8231 IMPORT HS
JVxxxxx	JOVIAL
MIxxxxx	1700 MSOS IMPORT HS
MRxxxxx	MARS VI
OHxxxxx	OPHELIE
OPxxxxx, OTxxxxx	OPTIMA
PLxxxxx	PL1
PTxxxxx	PERT/TIME
QUxxxxx	QUERY UPDATE
RMxxxxx	6RM
SCxxxxx	SCOPE
SIxxxxx	SIMSCRIPT
SMxxxxx, SOxxxxx	SORT/MERGE
SSxxxxx	SIMSCRIPT
SUxxxxx	SIMULA
Uxxxxxx, Vxxxxxx, Wxxxxxx, Xxxxxxx, Yxxxxxx, Zxxxxxx	Reserved for Installation

FET ERROR CODES - WORD 1

01	End of information
02	End of reel
04	Parity error
10	Device capacity exceeded
21	End of multi-file set
22	Fatal error
23	Index buffer full
24	Reserved
25	Index full on random read/write of record n
26	Nonexistent record named on random read
27	Nonexistent record named on random write and index is full
30	Function undefined on device
31	Permission not granted
32	Function illegal on permanent file
33-37	Reserved

FET FLAG BITS - WORD 2 - Meaning if bit is set

Bit	47	(R)	Process SCOPE index if OPEN/CLOSE; else random read/write
	46	(N)	Release record blocks as read
	45	(UP)	User processing at end of volume
	44	(EP)	User processing on error condition
	43	(EB)	Reserved
	42	(IN)	INTERCOM (multi-user job or graphics)
	41	(XL)	Extended label processing
	40	(XP)	Extended error processing
	39	(EC)	Disallow automatic allocation of ECS buffer
	38	(NS)	File has non-standard labels; processing of label records is left to user
	37		Reserved
	36	(ST)	6000 Station control point

FET DETAIL ERROR CODES - WORD 7

Software Warning

0000-0017	(Reserved)
0020	25 feet erased (not ANSI compatible)
0021	Erase limit (installation-defined)
0023	Bad erase
0024	Read opposite mode successful
0025	Noise in inter-record gap
0026	Function not complete
0027	Possible record fragment
0030	Record length greater than PRU size or MLRS as applicable

Bad Hardware Status

0040	Lost data
0041	Tape parity error

FET DETAIL ERROR CODES – WORD 7 (continued)

Hardware Malfunction

0050	MMTC memory parity error
0051	Transmission parity error

Position Uncertain

0100	Valid data probably destroyed – highly unlikely that a close will be successful
0101	Valid data probably intact – likely that a close will be successful

FILE INFORMATION TABLE

Bit positions: 59, 53, 47, 41, 35, 29, 23, 17, 11, 5, 0

Word	Bits 59–18	Bits 17–0
0	LFN Logical File Name	
1	RL Current Record Length; PM; FO; MIX; BT; BCK; DT; RT; DKI; I; PD	FDT File Description Table; FET FET Address
2	PTL Partial Transfer Length; OF; VF; CF; LT; ULP; FP	LX Address of Label Routine
3	HL Header Length of T Record; MNR Minimum Length of R Record; BFS Buffer Size (in Words)	DX Address of End-of-Data Routine
4	TL Trailer Length of T Record; SES; IRS; ES	EX Address of Error Routine
5	VNO; ECT Error Count; ERL Error Limit; FPB; SVO; SPR; SDS; OC; FWI; ON; LCR; HB; LVL	FWB FWA of User Buffer
6	FL Length of F/Z Record; MRL Maximum Record Length; CM; EO	WSA (WSAL) FWA of Working Storage Area
7	KP; KL Key Length; MKL Major Key Length; RKP; RKW; IP; DP	PNA (PNAL) Address of Partition Name; KA Address of Key
8	MNB Minimum Block Length; RMK Record Mark Character; PC; KT	LA FWA of Label Area
9	LP BCP of D Record Length Field; CP BCP of T Record Trailer Count Field; RB No. of Records in K Block	PAR Parameter List Address
10	LL Length of D Record Length Field; CL Length of T Record Trailer Count Field; WPN; LOP; RC Record Count	RC Record Count
11	LBL Length of Label Area; MUL; BN Current Block Number; TRC	BN Current Block Number; ECL Error Code Location
12	MBL Maximum Block Length; NL; FLM; DL	FLM; DL; PRS
13	IBL Index Block Length; WA Current Word Address	WA Current Word Address
14	Reserved for Installation	
15	HMB Number of Home Blocks; CDT; HRL Address of Key Hashing Routine	HRL Address of Key Hashing Routine; DCT
16		

NOTES: FIT

Word 1:	35	PM	processing mode (6RM only)	
			0	random
			1	sequential
	34-32	FO	file organization	
			000	sequential (SQ)
			001	word addressable (WA)
			010	direct (DR)
			011	indexed sequential (IS)
			100	library (partition) (LB)
			101	direct access (DA)
	31	MIX	reserved for future access method	
	30-28	BT	block type	
			000, 001	default/internal (I)
			010	character count (C)
			011	record count (K)
			100	exact records (E)
	27	BCK	block checksums (not supported for SQ files)	
			0	no
			1	yes
	26	DT	device type (6RM only)	
			0	mass storage
			1	tape
	25-22	RT	record type	
			0000	control word (W)
			0001	fixed length (F)
			0010	record mark (R)
			0011	zero byte (Z)
			0100	decimal character count (D)
			0101	trailer count (T)
			0110	binary character count (B)
			0111	undefined (U)
			1000	SCOPE logical records (S)
	21	DKI	duplicate key indicator; indicates duplicate key permission on an IS file.	
			0	no
			1	yes
	20-18	PD	processing direction	
			000, 001	input (INPUT)
			010	output (OUTPUT)
			011	input/output (I-O)
			100	reverse (REV)

NOTES: FIT

Word 2:	35-34	OF	open flags, positioning of file at OPENM time 00 } rewind (R) 01 } 10 no rewind (N) 11 extend (E)
	33-32	VF	end of volume flags, positioning of file volume CLOSEM time 00 unload (default) 01 rewind (R) 10 no rewind (N) 11 unload (U)
	31-30	CF	close flags, positioning of file at CLOSEM time 00 } rewind (R) 01 } 10 no rewind (N) 11 unload (U)
	29-28	LT	label type 00 ANSI standard (ST) 01 non-standard (NS) 10 unlabeled (UL) 11 any (ANY)
	27-25	ULP	user label processing 000 none 001 VOL/EOV (V) 010 HDR/EOF (F) 011 VOL/HDR/EOF/EOV (VF) 100 UHL/UTL/UVL (U) 101 VOL/UHL/UTL/UVL/EOV (VU) 110 HDR/EOF/UHL/UTL/UVL (FU) 111 all (VFU)
	24-18	FP	file position (in octal) 0 mid record 1 in header label group. (only set during user label processing) 2 beginning of information/volume (BOI/BOV) only set on SKIPBu in connection with DX 4 end of volume (EOV) 10 end of section (EOS) 20 end of record (EOR) 40 end of partition (EOP) 100 end of information (EOI)

NOTES: FIT

Word 4:	35-18	ES	error status
	35	FNF	fatal/non-fatal flag 0 non-fatal 1 fatal
	34-31	reserved	
	30-27	SES	system error severity 01 read parity error level 1 02 read parity error level 2 03 read parity error level 3 04 read parity error level 4 05 write parity error level 1 06 write parity error level 2
	26-18	IRS	invalid request subfield (6RM error code)
Word 5:	59-54	VNO	current volume number of multi-volume sequential file
	35	FPB	file position bit; for use by user
	34	SVO	SIS version number 0 SISV1 user 1 SISV2 user
	33	SPR	suppress read ahead 0 read ahead/write behind (buffered sequential I/O) 1 no read ahead/no write behind (unbuffered sequential I/O)
	32	unused	
	31	SDS	system error message disposition
	30-29	OC	open/close 00 never opened 01 opened 10 closed
	28	FWI	forced write of IS and DA file blocks 0 no forced write (NO) 1 forced write (YES)
	27	ON	old/or new file 0 old (OLD) 1 new (NEW)

NOTES: FIT

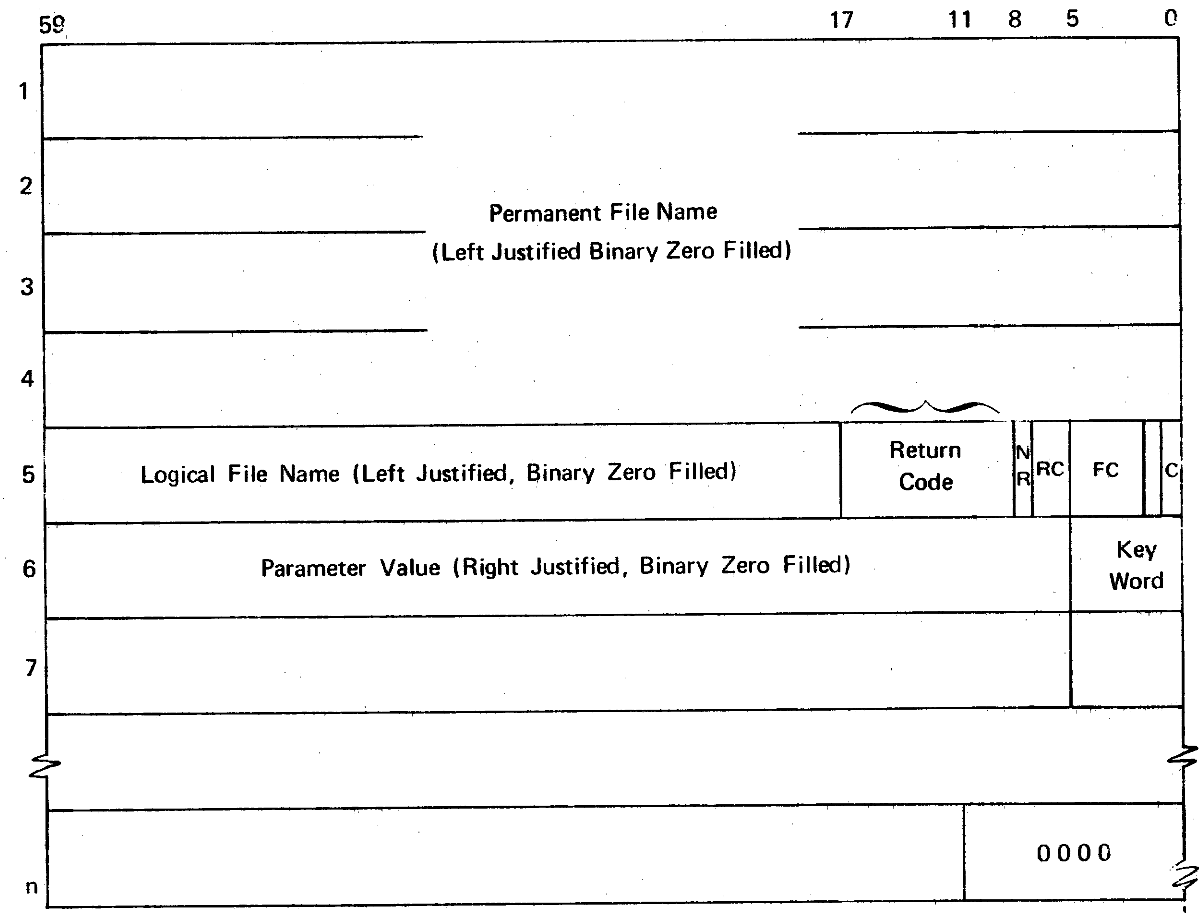
Word 5:	**(Cont'd)**		
	26	LCR	label action on PD=I-O tape 0 create (new) labels (N) 1 check existing labels (E)
	24	HB	reserved for future access method
	23-18	LVL	level number of an S-type record
Word 6:	25	CM	conversion mode (EC to IC) 0 no conversion (NO) 1 conversion (YES)
	24-22	EO	error option 000 terminate job (T) 001 drop erroneous data (D) 010 accept (A) 100 terminate job and display data (TD) 101 drop erroneous data and display data (DD) 110 accept erroneous data and display data (AD)
Word 7:	59-56	KP	beginning character position of key
	37-34	RKP	relative key word position (DA files)
	33-22	IP	index block padding factor (% padding)
	28-22	DP	data block padding factor (% padding)
Word 8:	27-22	PC	padding character for sequential file blocks
	27-22	KT	key type for indexed sequential files 000000, 000001 symbolic (S) 000010 integer (I) 000011 floating (F) 000100 computational-1 000101 actual (A) 000110 computational-2 000111 . . 111111 reserved for CDC

NOTES: FIT

Word 10:	35-30	LOP	last operation code (the high order bit of LOP is a "write bit", indicating whether or not the last operation wrote data to the file). 01 OPENM (OP) 02 CLOSEM (CM) 03 GET or GETP (GE) 43 PUT or PUTP (PU) 44 REPLACE (RP) 04 SEEK (SE) 05 SKIP (SK) or SKIPF (SF) 06 DELETE (DE) 07 GETN (GN) 47 WEOR (WE) 10 REWIND (RE) 11 GETL (GL) or PUTL (PL) 12 SKIPB (SB) 13 CLOSEL (CL) 14 LABEL (LL) 7DM only 63 WTMK (WK) 23 READS (RS) 53 WRITES (WS)
	35	WPN	write bit. The upper bit of LOP is a one-bit subfield that can be accessed separately; it is on (YES) if the last operation was a write.
Word 11:	35-30	MUL	multiple of characters per K, E type block
	23-18	TRC	number of transactions to be traced (IS or DA files)
Word 12:	35-30	NL	number of levels of index blocks in an IS file
	29-0	DL	directory length
	29-0	FLM	file limit, records/file (IS/DA)
	17-0	PRS	previous record size after a forward sequential GET
Word 15:	51-30	CDT	address of collating sequence to display code conversion table for indexed sequential (IS) files
	21-0	DCT	address of display code to collating sequence conversion table for indexed sequential (IS) files

Words 16, 17, 18: Scratch area used by 6RM

FILE DEFINITION BLOCK



<u>NR</u> (Bit 8)

1 = NR option specified

<u>RC</u> (Bits 7-6)

01 = No RC or RT specified
00 = RC option specified
10 = RT option specified (implies RC as well)

<u>FC</u> (Bits 5-2)

0001 = SETP
0010 = ATTACH
0100 = CATALOG
0110 = EXTEND
0111 = ALTER
1000 = PURGE
1010 = RENAME
1100 = PERM

<u>C</u>

1 = Function completed

NOTES: FILE DEFINITION BLOCK

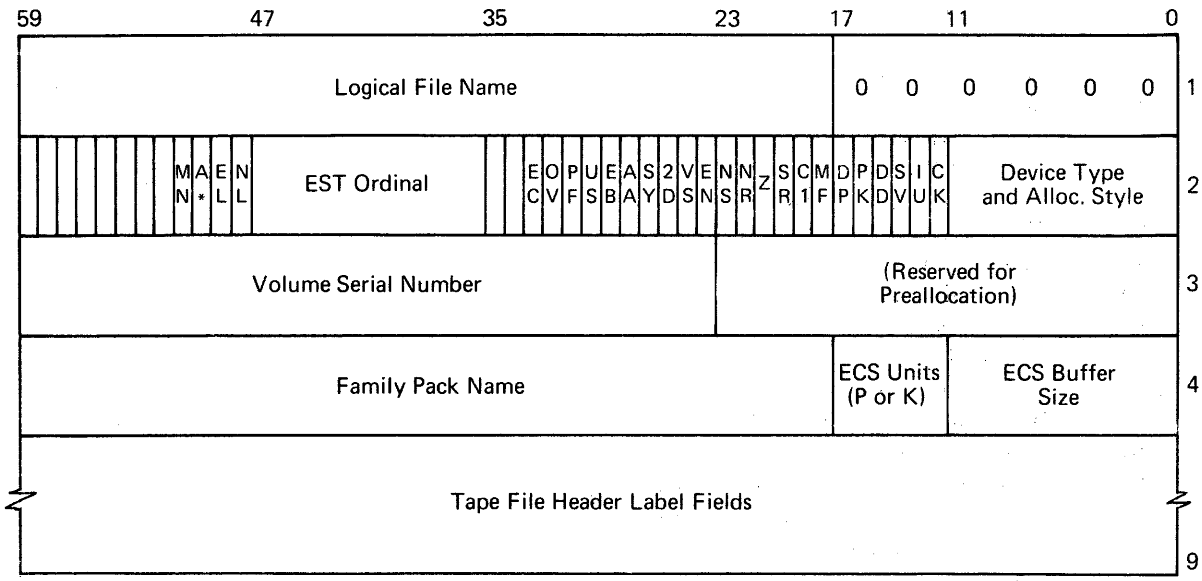
Return code (Bits 17-9)

0	Function successful
1	ID error
2	Lfn already in use
3	Unknown lfn
4	No room for extra cycle
5	RBTC full
6	No lfn or pfn
10	Latest index not written
11	File not on PF device
12	File not in system
13	Archive retrieval aborted
15	Cycle number limit reached
16	PFD Full
17	Function attempted on non-permanent file
20	Function attempted on non-local file
22	File never assigned to a device
23	Cycle incomplete or dumped
24	PF already attached
25	File unavailable
27	Illegal lfn
33	Alter needs exclusive access
35	File already in system
70	PFM stopped by system
71	Incorrect permission
72	FDB address error

Key Word

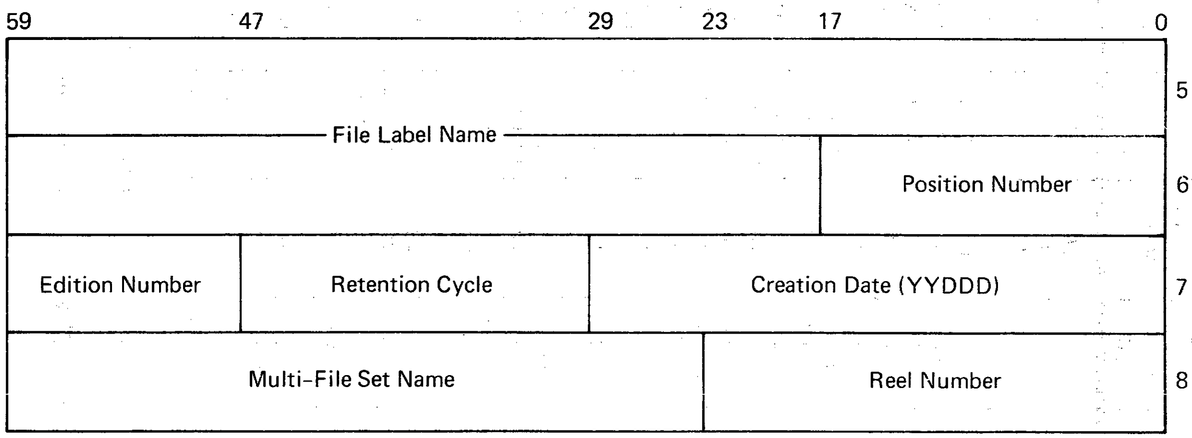
01	PP	– Privacy Procedure PP name
02	RP	– Retention Period (days)
03	CY	– Cycle Number
04	TK	– Turnkey Password Definition
05	CN	– Control Password Definition
06	MD	– Modify Password Definition
07	EX	– Extend Password Definition
10	RD	– Read Password Definition
11	MR	– Multi-Read Parameter
12	SD	– Ignored
13	XR	– Control, Modify, Extend Password Definition
14	ID	– Owner-Identification
15	RN	– Ignored
16	AC	– Account Parameter
17	EC	– ECS Buffering
20, 21, 22, 23, 24	PW	– Passwords
25	FO	– File Organization
26	PS	– Position
31	LC	– Lowest Cycle
32	ST	– Station ID (7000 only)
33	RW	– Multi-Access Rewrite

REQ FUNCTION PARAMETER LIST

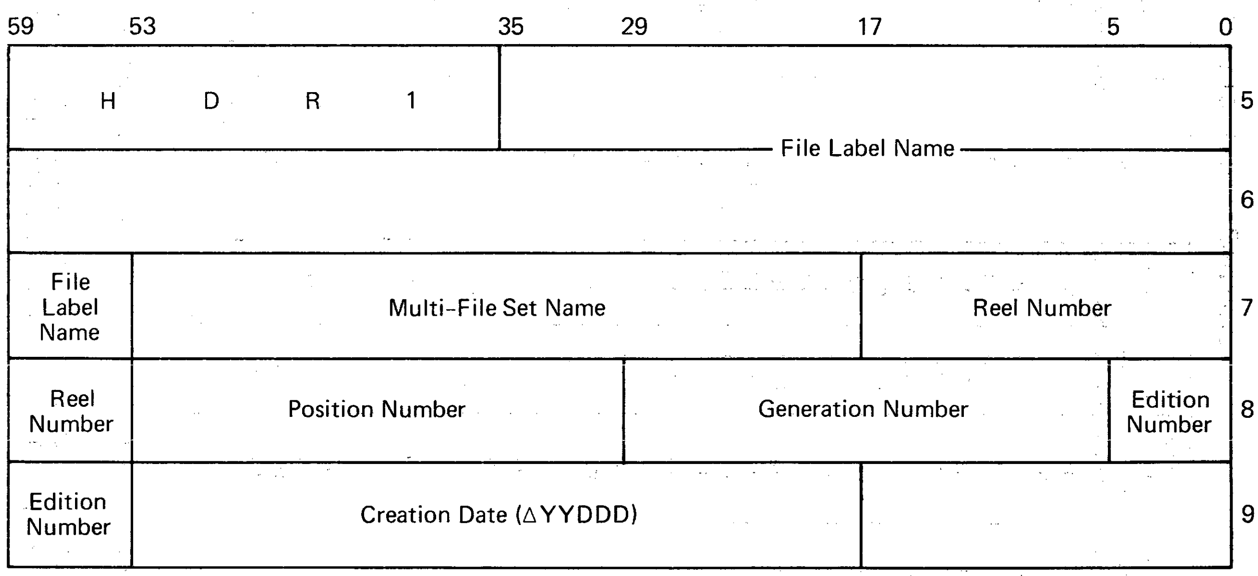


Bits	59-52		Reserved
	51	(MN)	1 = accept either MT or NT assignment
	50	(A*)	1 = assign any RMS device (overrides device type specification)
	49	(EL)	1 = extended label fields in parameter words 5-9
	48	(NL)	1 = normal label fields in parameter words 5-8
	33	(EC)	1 = ECS buffering requested; parameter word 4
	32	(OV)	1 = allow overflow to a different device
	31	(PF)	1 = assign file to a PF device
	30	(US)	1 = ASCII conversion mode on 9-track tape
	29	(EB)	1 = EBCDIC conversion mode on 9-track tape
	28	(AA)	1 = assign automatically
	27	(SY)	1 = print card image from RA+70
	26	(2D)	1 = assign 2 devices (DP,NT,MT only)
	25	(VS)	1 = VSN declared in parameter word 3
	24	(EN)	1 = tape has existing labels
	23	(NS)	1 = tape has non-standard labels
	22	(NR)	1 = disable standard tape parity recovery procedure
	21	(Z)	1 = SCOPE 3.3 labeled tape
	20	(SR)	1 = return error code without dayfile message or operator intervention
	19	(C1)	1 = console checkpoint request
	18	(MF)	1 = multi-user tape request
	17	(DP)	1 = sequential pack request
	16	(PK)	1 = family pack request
	15	(DD)	1 = default density for labels and data
	14	(SV)	1 = save tape
	13	(IU)	1 = inhibit physical unload
	12	(CK)	1 = checkpoint tape

NORMAL LABEL FIELDS



EXTENDED LABEL FIELDS



SCOPE 3.4

STANDARD CHARACTER SETS

CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code	CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code
:†	:	00†	8-2	00	8-2	3A	6	6	41	6	06	6	36
A	A	01	12-1	61	12-1	41	7	7	42	7	07	7	37
B	B	02	12-2	62	12-2	42	8	8	43	8	10	8	38
C	C	03	12-3	63	12-3	43	9	9	44	9	11	9	39
D	D	04	12-4	64	12-4	44	+	+	45	12	60	12-8-6	2B
E	E	05	12-5	65	12-5	45	–	–	46	11	40	11	2D
F	F	06	12-6	66	12-6	46	*	*	47	11-8-4	54	11-8-4	2A
G	G	07	12-7	67	12-7	47	/	/	50	0-1	21	0-1	2F
H	H	10	12-8	70	12-8	48	(	(	51	0-8-4	34	12-8-5	28
I	I	11	12-9	71	12-9	49	)	)	52	12-8-4	74	11-8-5	29
J	J	12	11-1	41	11-1	4A	$	$	53	11-8-3	53	11-8-3	24
K	K	13	11-2	42	11-2	4B	=	=	54	8-3	13	8-6	3D
L	L	14	11-3	43	11-3	4C	blank	blank	55	no punch	20	no punch	20
M	M	15	11-4	44	11-4	4D	, (comma)	, (comma)	56	0-8-3	33	0-8-3	2C
N	N	16	11-5	45	11-5	4E	. (period)	. (period)	57	12-8-3	73	12-8-3	2E
O	O	17	11-6	46	11-6	4F	≡	#	60	0-8-6	36	8-3	23
P	P	20	11-7	47	11-7	50	[	[	61	8-7	17	12-8-2	5B
Q	Q	21	11-8	50	11-8	51	]	]	62	0-8-2	32	11-8-2	5D
R	R	22	11-9	51	11-9	52	%††	%	63	8-6	16	0-8-4	25
S	S	23	0-2	22	0-2	53	≠	" (quote)	64	8-4	14	8-7	22
T	T	24	0-3	23	0-3	54	→	_ (underline)	65	0-8-5	35	0-8-5	5F
U	U	25	0-4	24	0-4	55	∨	!	66	11-0 or	52	12-8-7 or	21
V	V	26	0-5	25	0-5	56				11-8-2†††		11-0†††	
W	W	27	0-6	26	0-6	57	∧	&	67	0-8-7	37	12	26
X	X	30	0-7	27	0-7	58	↑	' (apostrophe)	70	11-8-5	55	8-5	27
Y	Y	31	0-8	30	0-8	59	↓	?	71	11-8-6	56	0-8-7	3F
Z	Z	32	0-9	31	0-9	5A	<	<	72	12-0 or	72	12-8-4 or	3C
0	0	33	0	12	0	30				12-8-2†††		12-0†††	
1	1	34	1	01	1	31	>	>	73	11-8-7	57	0-8-6	3E
2	2	35	2	02	2	32	≤	@	74	8-5	15	8-4	40
3	3	36	3	03	3	33	≥	\	75	12-8-5	75	0-8-2	5C
4	4	37	4	04	4	34	¬	^ (circumflex)	76	12-8-6	76	11-8-7	5E
5	5	40	5	05	5	35	; (semicolon)	; (semicolon)	77	12-8-7	77	11-8-6	3B

†Twelve or more zero bits at the end of a 60-bit word are interpreted as end-of-line mark rather than two colons. End-of-line mark is converted to external BCD 1632.

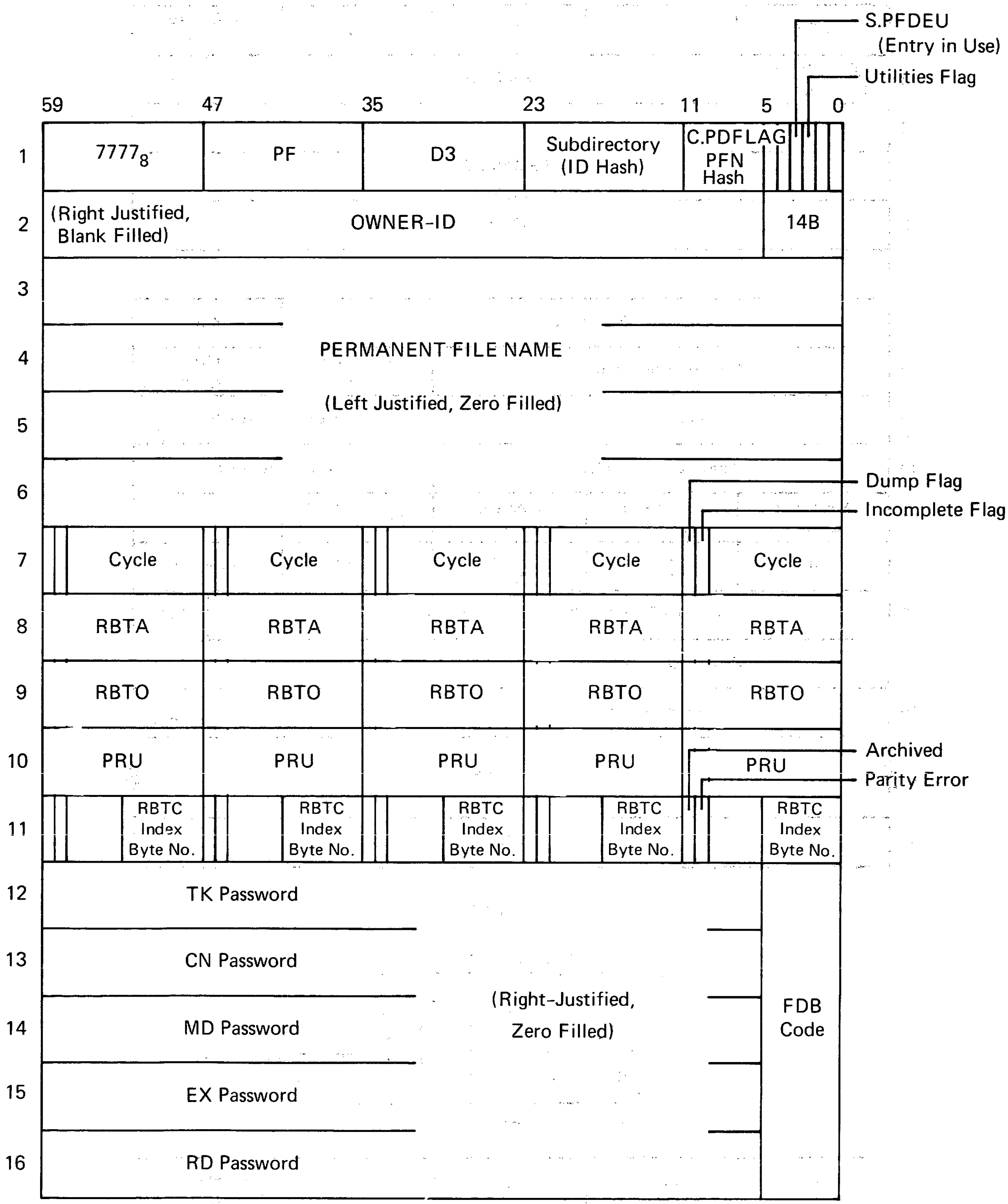
††In installations using the CDC 63-graphic set, display code 00 has no associated graphic or Hollerith code; display code 63 is the colon (8-2 punch).

†††The alternate Hollerith (026) and ASCII (029) punches are accepted for input only.

SECTION 3

DISK TABLES AND FILE FORMATS

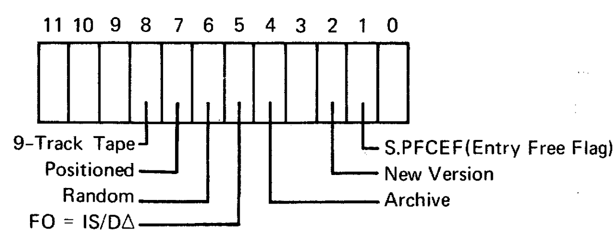
PERMANENT FILE DIRECTORY ENTRY



RBT CATALOG ENTRY

Word	59	47	35	23	11	5–0
1	7777	7777	RB	TC	C.PCFLAG Flags	
2	(Right Justified, Blank Filled)	OWNER-ID				14B
3	PERMANENT					
4	FILE					
5	NAME					
6						
7	PFD Pointer				CYCLE	
8	Creation Date (J-Date)			Retention Period	EST Ordinal	
9	Date of Last Attach			Time of Last Attach		
10	Date of Last Alteration			Time of Last Alter		
11	Number of Attaches	Number of Extends	Number of Rewrites	Size of Entry		
12	0	Pointer to S	Pointer to T	Subdirectory	(Reserved)	
13	Account Parameter (display zero-filled binary)					16B
14	Dump Tape VSN No. 1			Position		
15	Dump Tape VSN No. 2			Position (CKP)		
S	Installation Slot					
T	RBT Chain					

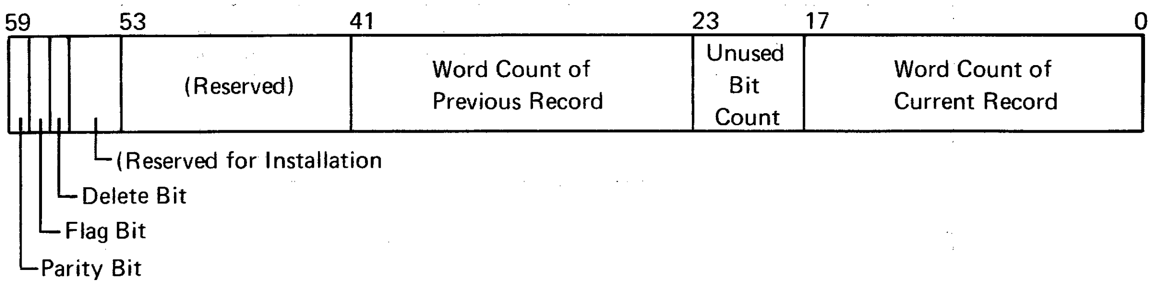
C.PCFLAG



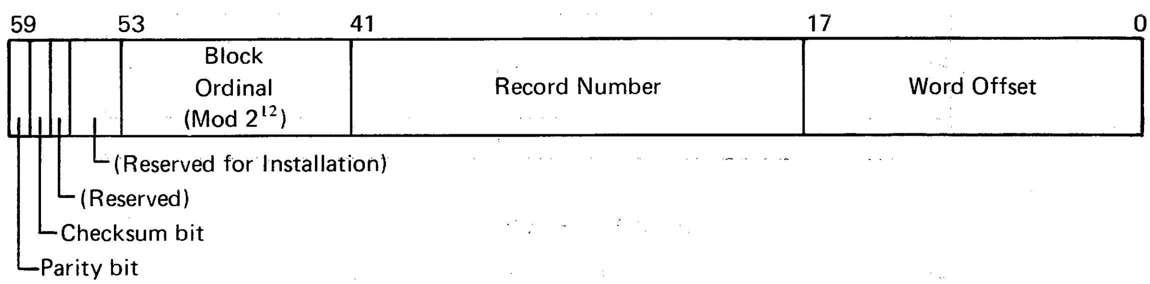
Index$_0$	Index$_1$	Index$_2$	Index$_3$	Number of Free Words in this PRU

Word 0 of an RBTC pru.

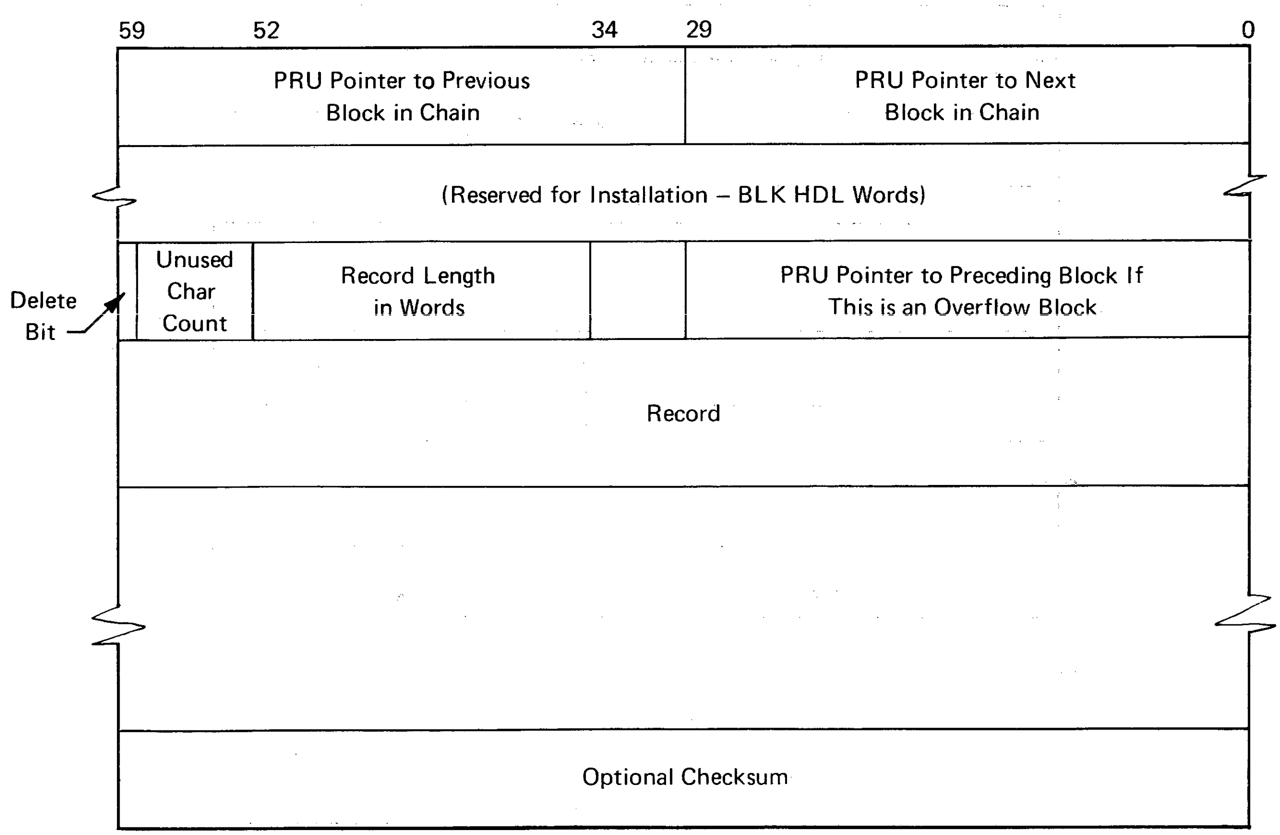
W RECORD HEADER



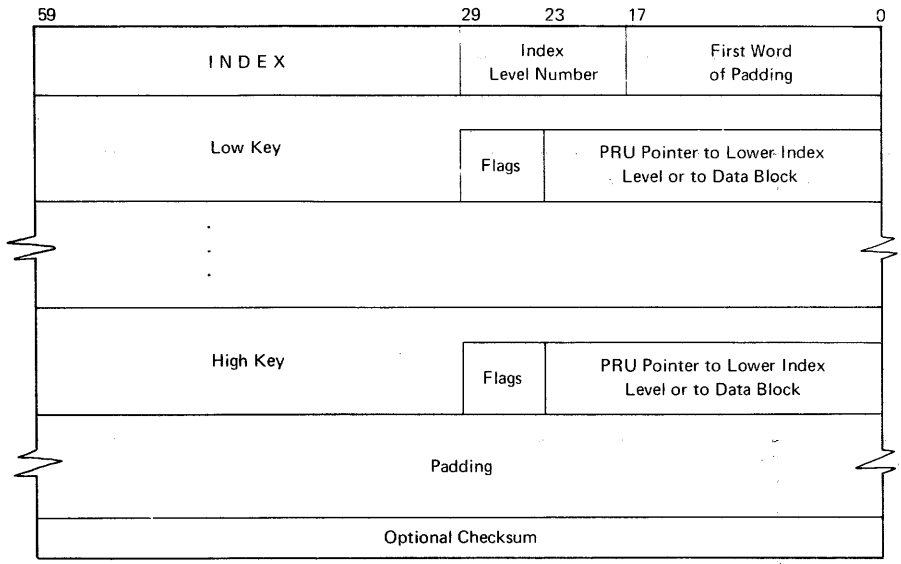
I BLOCK HEADER



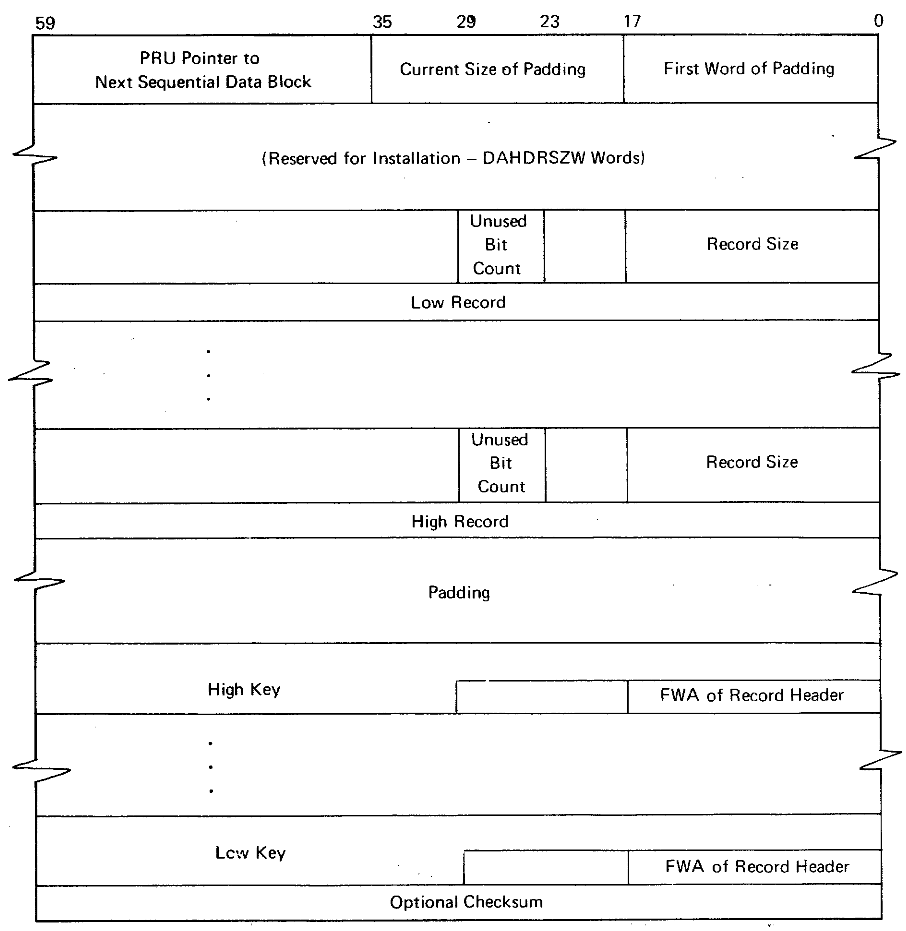
BLOCK FORMAT (DA FILE)



INDEX BLOCK FORMAT (IS FILE)



DATA BLOCK FORMAT (IS FILE)



FSTT (IS FILE)

Bit positions: 59, 44, 35, 23, 17, 0

Field	Contents (bits 59–0)	Word
CHEKSUM	FSTT Checksum	0
HEADWORD	S A A M – S I S V 2	1
TRAILERS	(Reserved for SIP)	2
FSTTSZW	FSTT Size in Words	3
DAHDRSZW	Data Header Size (words) (bits 23–0)	4
ABNORMB	← File Abnormally Terminated (bit 59)	5
AVALDABK	PRU Number of Next Empty Data Block (bits 23–0)	6
AVALINBK	PRU Number of Next Empty Index Block (bits 23–0)	7
CHKSUMB	← Checksum File (bit 59)	10
CODITABL	Collating Sequence to Display Code Conversion Table	11
CODITAB2		12
CODITAB3		13
CODITAB4		14
CODITAB5		15
CODITAB6		16
CODITAB7		17
CODITAB8		20
CURFILPP	PTREE Index (bits 44–36); Key Displacement in Data Block (bits 35–18); Address of Buffer Catalog (bits 17–0)	21
DABKSZP	Data Block Size in PRUs (bits 17–0)	22
DABKSZW	Data Block Size in Words (bits 23–0)	23
DAPADSZW	Data Block Padding Size in Words (bits 23–0)	24
DELETSTO	Total DELETEs	25
DICOTABL	Display Code to Collating Sequence Conversion Table	26
DICOTAB2		27
DICOTAB3		30
DICOTAB4		31
DICOTAB5		32
DICOTAB6		33
DICOTAB7		34
DICOTAB8		35
DUPKEYB	← Duplicate Keys Allowed (bit 59)	36
FEMDABK	Deleted Data Block Chain Pointer (bits 23–0)	37
FEMINBK	Deleted Index Block Chain Pointer (bits 23–0)	40
FILDATE	File Creation Date	41
FILNAME	Logical File Name	42
FIRDABK	First Data Block Pointer (bits 23–0)	43
INBKSZP	Index Block Size in PRUs (bits 17–0)	44
INBKSZW	Index Block Size in Words (bits 23–0)	45

FSTT (IS FILE)

Bit positions: 59 29 23 11 8 2 0

Field	Bits 59–30	Bits 29–24 / 23–0	Word
INPADSZW		Index Block Padding Size in Words	46
KEYBIAS		COMP-1 Floating Bias	47
KEYENSZW		Key Entry Size (words)	50
KEYPACKB	Key Ends in Character Position 0-4 of Last Word		51
KEYSZC		Key Size in Characters	52
KEYTYPE		Key Type	53
LASDABK		Final Data Block Pointer	54
MAXRECSZ		Maximum Record Size in Chars	55
MINRECSZ		Minimum Record Size in Chars	56
NRACSTO	Total GETs		57
NRDABKTO		Total Number of Data Blocks	60
NRDAREC	Total Number of Data Records		61
NREMDABK		Number of Deleted Data Blocks	62
NREMINBK		Number of Deleted Index Blocks	63
NRINBKTO		Total Number of Index Blocks	64
NRINSTO	Total PUTs		65
NRLVLSF		Number Index Block Level	66
NRWUDISK	Number of Unused Words on Disk		67
NXTUNASP		Next Unassigned PRU Number	70
PRINBK		Primary Index Block PRU Number	71
REPLACTO	Total REPLACEs		72
SEQINS$	Limit of Sequential INSERTs	Number of Sequential INSERTs	73
TERMDATE	Date of Last CLOSEM		74
TERMTIME	Time of Last CLOSEM		75
UEPRINBK	Number of Unused Key Entries in Primary Index Block		76

FSTT (DA FILE)

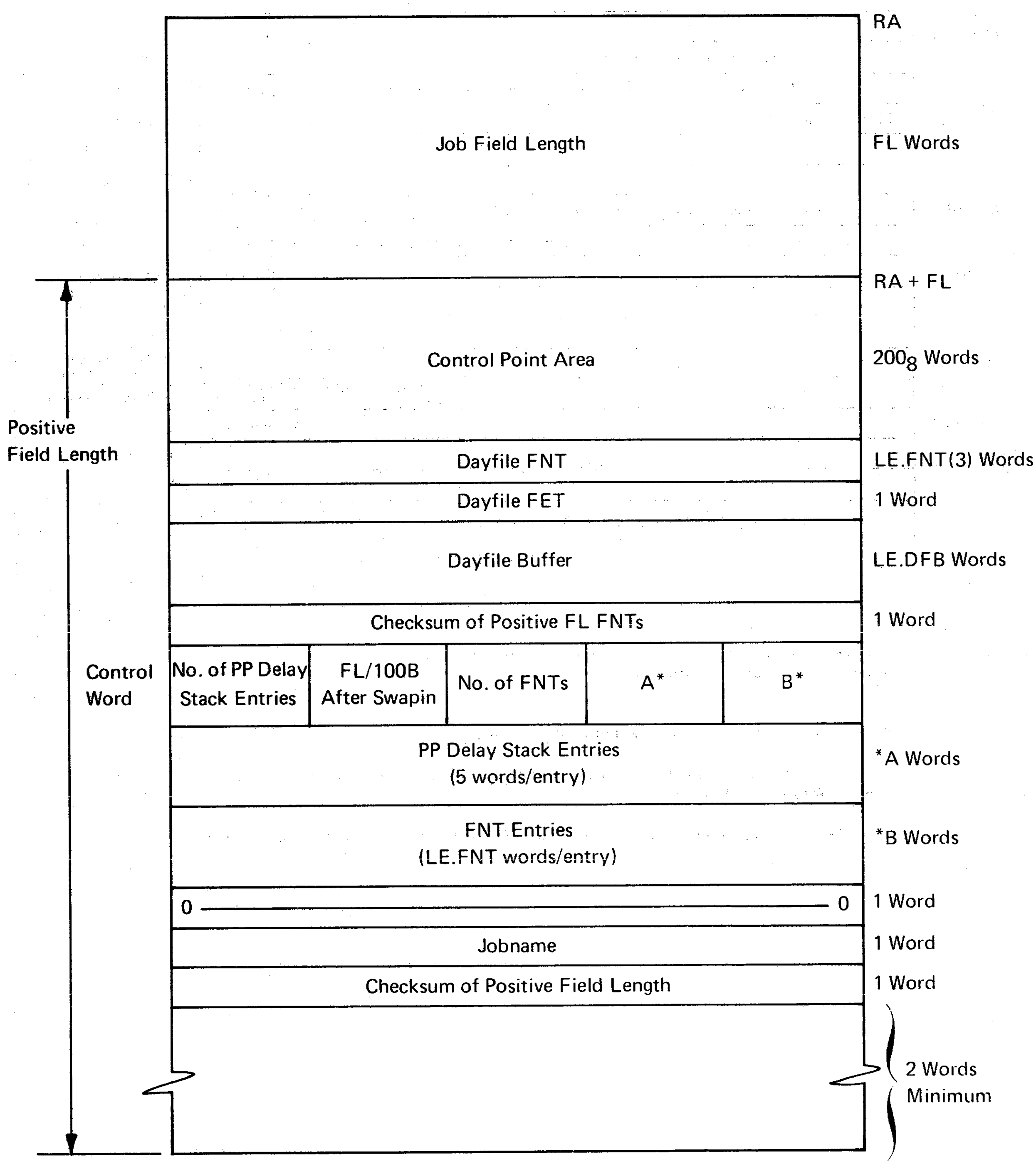
Bit positions: 59 53 35 29 23 17 0

Name	Contents	Word
CKSWORD	FSTT Checksum	0
HEADER	b F S T T b S D A b	1
TRAILER$	(Reserved for SIP)	2
FSTTSZW	FSTT Size in Words	3
DAHDRSZW	Data Header Size (Words)	4
CSFLAG	Block Checksum Option (BCK at OPENM NEW)	5
STATADD	Total Number of PUTs	6
STATDLT	Total Number of DELETEs	7
STATRPL	Total Number of REPLACEs	10
STATRET	Total Number of GETs and GETNs	11
STATOVF	Total Number of Overflow Blocks	12
STATHOM	Number of Home Blocks in Use	13
STATHR	Total Number of Records in Home Blocks	14
STATOR	Total Number of Records in Overflow Blocks	15
FILESZ	Total Number of Home Blocks	16
KEY$	Relative Key Position (RKP) | Relative Key Word (RKW) | Key Length (KL)	17
MINMAX$	Minimum Record Size (MNR) | Maximum Record Size (MRL)	20
NXTUNASP	Next Unassigned PRU Number	21
NXTUPDSK	Next Usable PRU on disk	22
OVF	Overflow Manager	23
DATABKSZ	Data Block Size (words)	24
PRUBLK	Number of PRUs per Block	25
ENDFIXD →		
RUNADD	Number of PUTs – Current OPENM	26
RUNDLT	Number of DELETEs – Current OPENM	27
RUNRET	Number of GETs and GETNs – Current OPENM	30
RUNRPL	Number of REPLACEs – Current OPENM	31
RUNOVF	Number of Overflow Blocks Created – Current OPENM	32
RUNHOM	Number of Home Blocks in Use First Time – Current OPENM	33
RECHOM	Number of Records in Home Blocks – Current OPENM	34
RECOVF	Number of Records Placed in Overflow Blocks – Current OPENM	35
NRCIOCLS	Total Number of CIO Calls	36
CURPRU	Current PRU for Use by GETN	37
RELDBW	Relative Data Block Word for Use by GETN	40
BUFCAT$	Length of Buffer Catalog (bits 20–17) | Address of Buffer Catalog	41

FSTT (DA FILE)

	59–30	29–24	23–18	17–0	
LASBZBUF				Address of Last Accessed Data Block	42
LASTSK		PRU Number of Last SEEK			43
BITMAP$				Address of Bit Map (Creation Run Only)	44
WORDCT	Record Header of Record to be PUT				45
STRNGSP				Stranger Space in Receiving Block	46
EMPTYSP				Empty Space in Receiving Block	47
AVAILSPC				Empty Space in Block of Record to be Removed	50
STRANGER				Stranger Space in Block of Record to be Removed	51
ADDRPL	File Position Undefined for GETN				52
RECLNGTH			Unused Bits	Record Length	53
KEYBUFF				Address of Key Alignment Buffer	54
OVERWRIT			Number PRUs after Last Block		55
FWBVALUE				Address of User Buffer	56
LSTOP	Last Operation Code				57
BCUPDT	BUFCAT Address of Last Update				60
FSTTPOS →	(Reserved)				
	(Reserved)				
Most Empty Table	PRU Address of a Data Block	Empty Space in the Block			66
					67
					70
⋮	⋮				
					77

SWAP FILE FORMAT



PUBLIC RMS AND DISK PACK LABEL (PRU 0)

Bit positions: 59, 47, 35, 29, 23, 17, 11, 5, 0

Symbol	Contents (bits 59–0)	Word	Pack type
W.LBID / W.LBDATE	D E V 1 | (blank) | Creation Date: Y Y D D D	0	
W.LBVID	VID (Right-justified, display zero filled) | 00 | 3 . 4	1	Public Pack
	0 0 0 0 | VID of Current Pack (Right justified, display zero filled)	1	Private Pack
W.LBPFD / W.LBRBTC	PFD RB | RBTC RB | (blank)	2	Public Pack
	(blank)	2	Private Pack
	(blank)	3	Public Pack
W.LBPRIV	Family Pack Name (Left-justified, binary zero filled) | Random Bits | Family Number | n	3	Private Family Pack
	$ S E Q | VID of First Pack (Right-justified, display zero filled)	3	Private Sequential Pack
W.LBRBR	(blank) | RBR Ordinal	4	Public Pack
	(blank)	4	Private Pack
	(blank)	5	
W.LPAKNO	(blank)	6	Public Pack & Private Family Pack
	(blank) | Pack Number of Current Pack (Right-justified, display zero filled)	6	Private Sequential Pack
W.LBCK	(blank) | Checksum of PRU	7	
W.LBFLAW	RBR Header Words	10_8	
		11_8	
	Public Pack Flaw Table (RBR Skeleton) / Private Pack RBR Body	. . .	

n = Number of files in family (0 - 77_8) if Family Number field = 0
Not significant if Family Number field ≠ 0

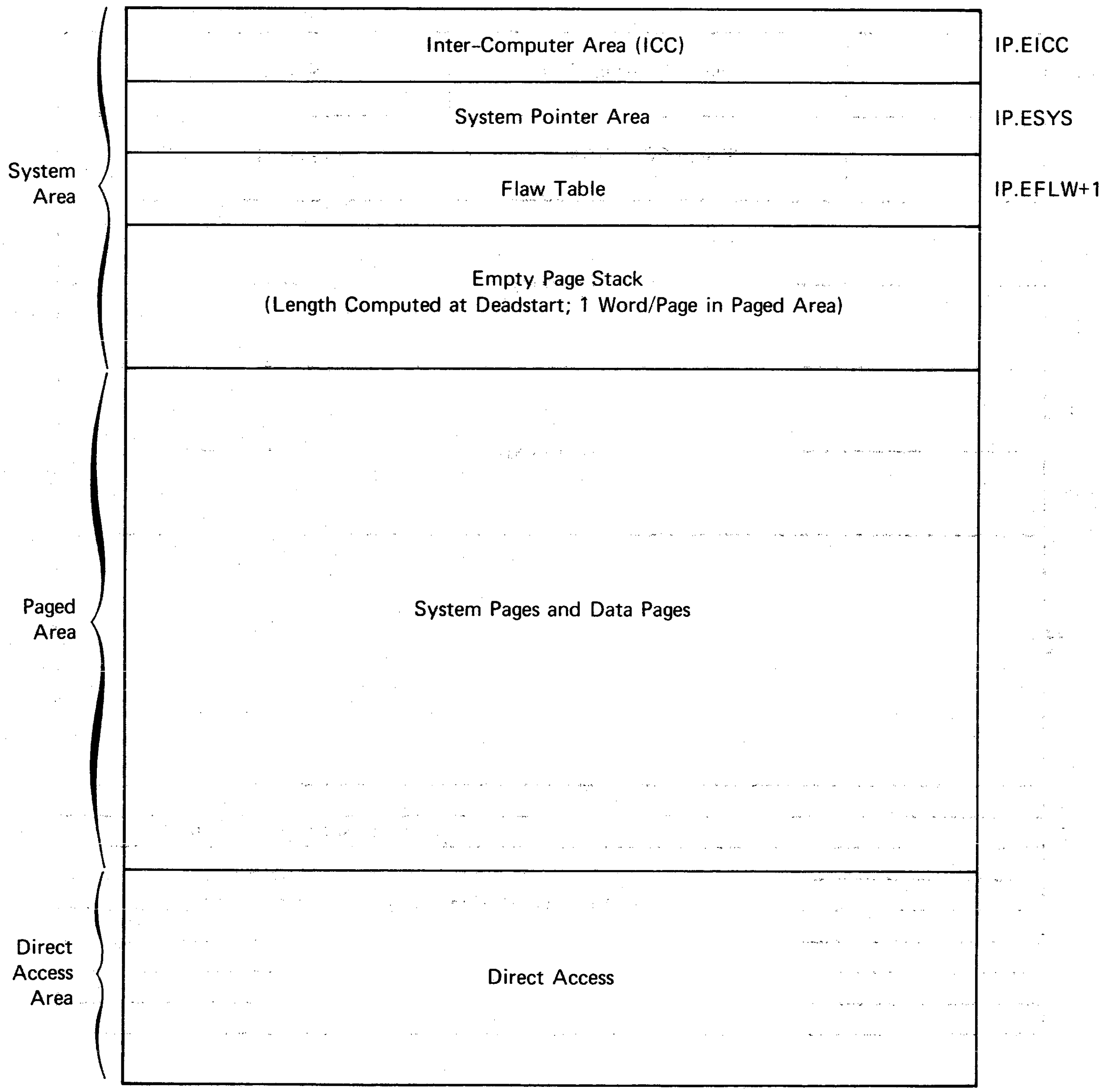
Public RMS Device labels can be more than 1 PRU. Possibilities are:

1. If an RBR has more than 2160D RB-s, label is written on 2 consecutive PRU-s, of which each has half of RB-s along with label header.
2. If a device has multiple RBR-s, each RBR occupies a PRU but the above (1.) holds true for each RBR.

SECTION 4

EXTENDED CORE STORAGE TABLES

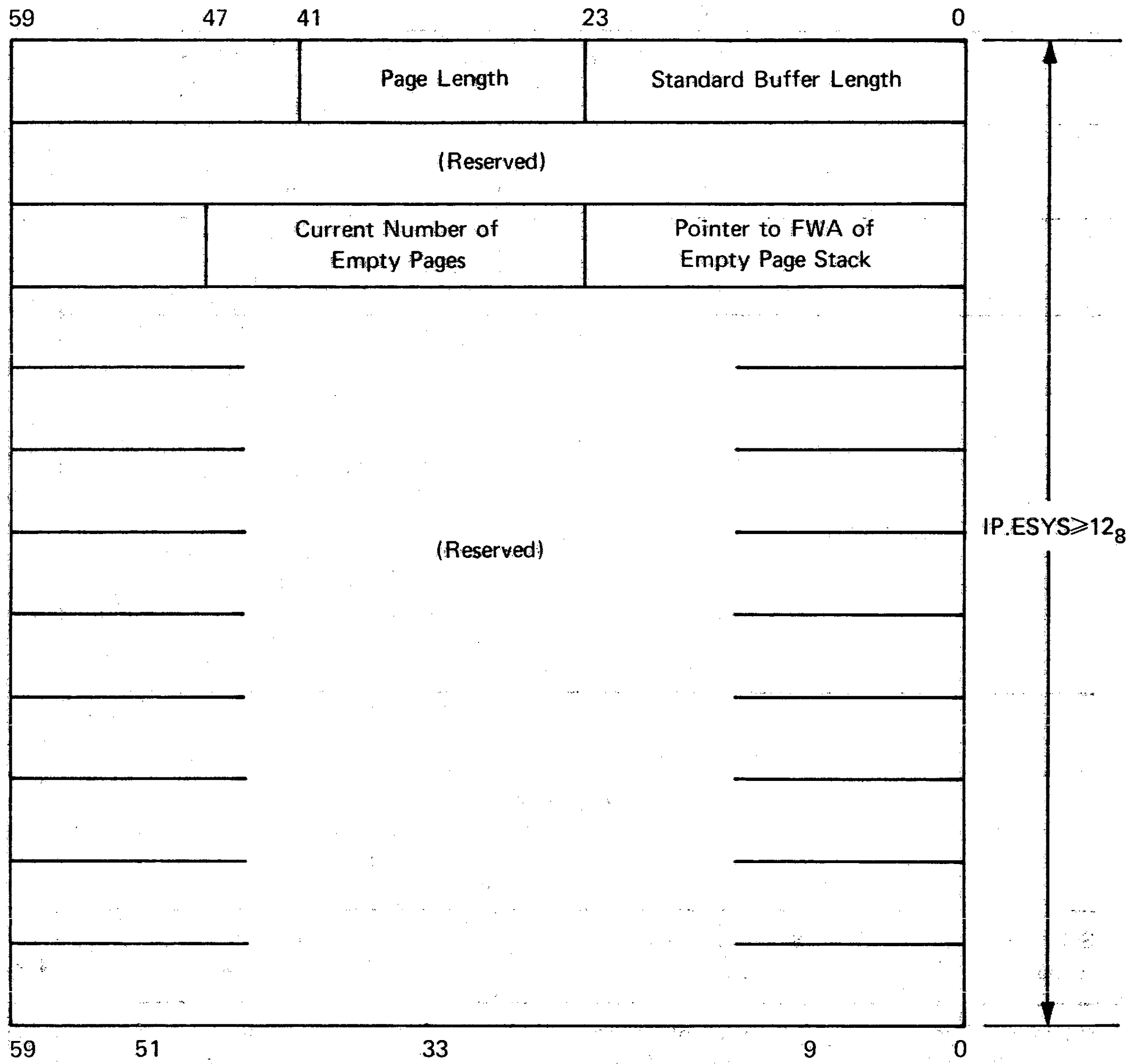
ECS FORMAT


Inter-Computer Area (ICC)
IP.EICC
System Pointer Area
IP.ESYS
Flaw Table
IP.EFLW+1
Empty Page Stack
(Length Computed at Deadstart; 1 Word/Page in Paged Area)
System Area
System Pages and Data Pages
Paged Area
Direct Access
Direct Access Area

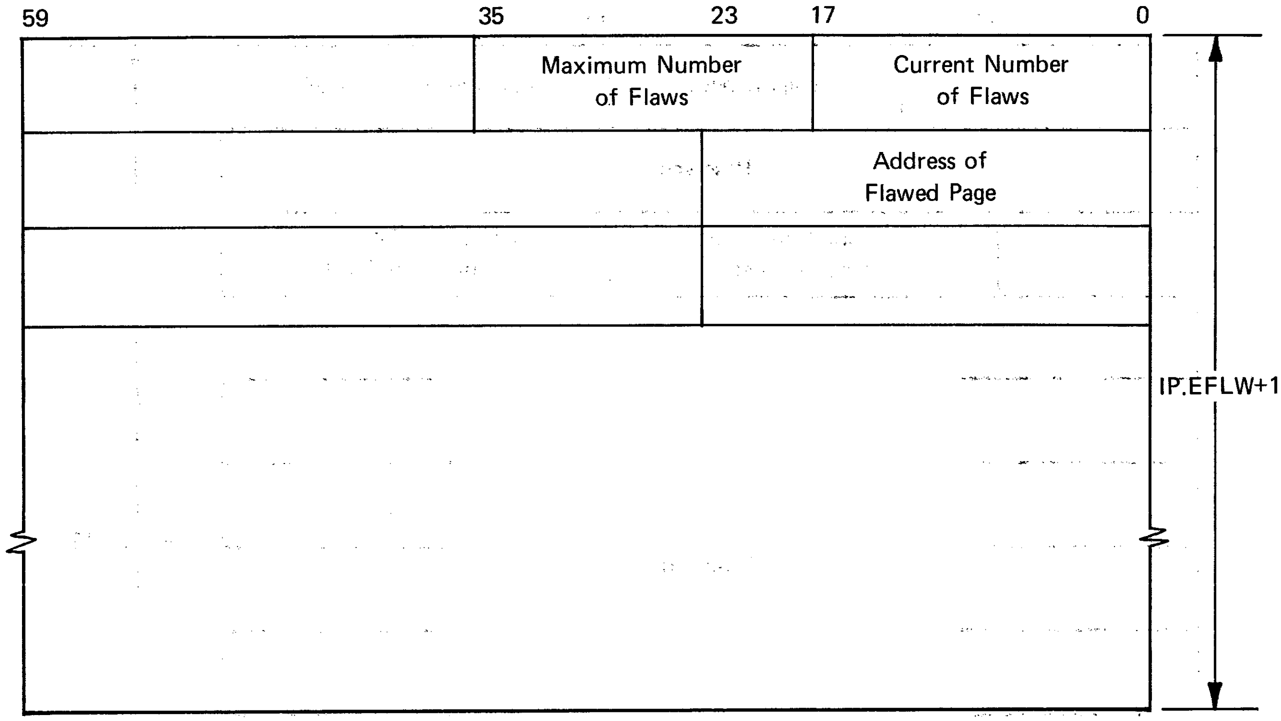
INTER-COMPUTER AREA

Word		47	23
0		Length of System Area	Length of ICC Area
1		Size of ECS/1000B	Library Size/1000B
2–3	(Reserved)		
4–5	(Reserved)		
⋮			
14			
15			Address 1st active system page
	ICC Areas (8 Words/Computer)		
⋮			

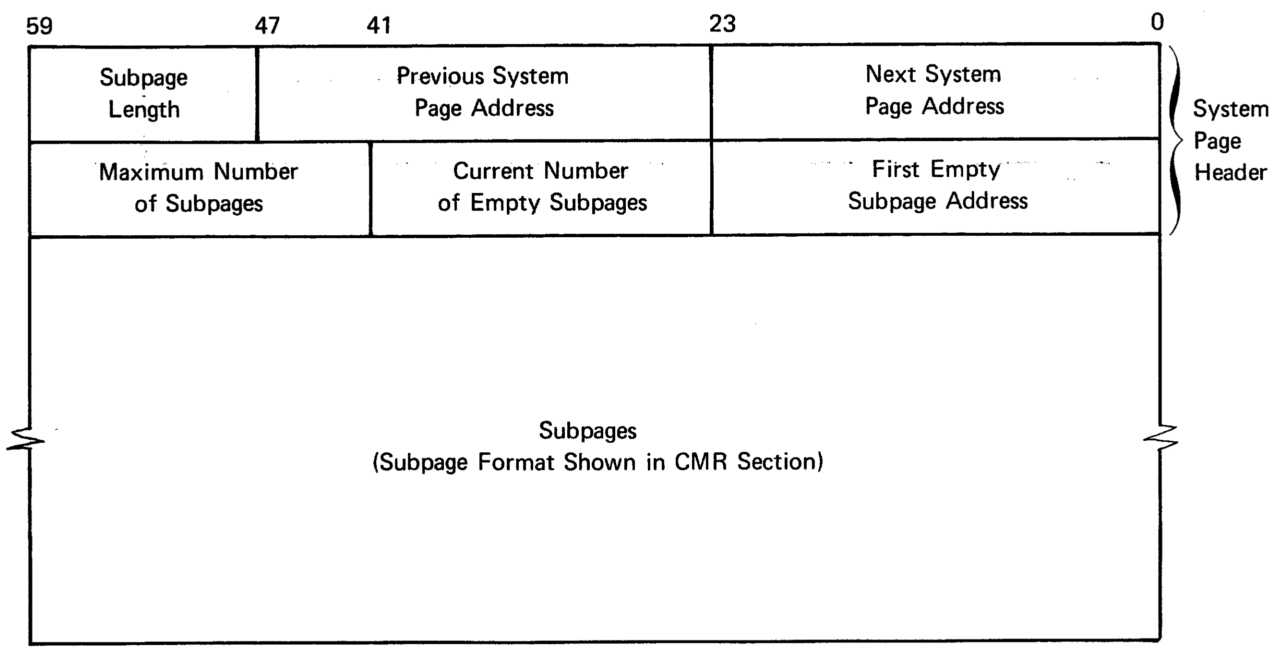
SYSTEM POINTER AREA



FLAW TABLE



SYSTEM PAGE



CUT ON THIS LINE

COMMENT SHEET

CONTROL DATA
CORPORATION

TITLE: SCOPE 3.4 Installation Handbook

PUBLICATION NO. 60307400 REVISION C

This form is not intended to be used as an order blank. Control Data Corporation solicits your comments about this manual with a view to improving its usefulness in later editions.

Applications for which you use this manual.

Do you find it adequate for your purpose?

What improvements to this manual do you recommend to better serve your purpose?

Note specific errors discovered (please include page number reference).

General comments:

FROM NAME: ______________________ POSITION: ______________________

COMPANY NAME: ____

ADDRESS: ____

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
Documentation Department
215 Moffett Park Drive
Sunnyvale, California 94086

CUT ON THIS LINE

FOLD FOLD

STAPLE STAPLE

CONTROL DATA

1/2 3/4 1 1-1/4

► ►CUT OUT FOR USE AS LOOSE-LEAF BINDER TITLE TAB

CONTROL DATA
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

Pub. No. 60307400

Litho in U.S.A.