THE SMALL BUFFER

THE SMALL BUFFER

8 NOVEMBER 1979

VOLUME 503 PAGE

ARTICLES OF GENERAL INTEREST

PERSONNEL CHANGES 1
NEW DIGITAL STANDARD MUMPS (DSM-11) BINARY PROGRAM UPDATE SERVICE 3

PRODUCT RELATED ARTICLES

BASIC-PLUS-2 V1.50
TWO RSTS/E BASIC-PLUS-2 PATCHES WILL BE SUPERSEDED 5

COS-310 V8/V8.01
CONVERTING SOURCE FILES FROM RX01 TO RX02 7

DPM V3.0
DPM BACKUP SUPPORT 9

DSM-11 V1
EDITOR REPLACES WRONG STRING 11
WRONG VERSION OF SMBU ON SYSBLD DISK 12

FORTRAN IV/IAS-RSX
MUST REBUILD SYSLIB FOR FORTRAN V2.2 WITH VIRTUAL ARRAYS 15

FORTRAN IV/RT-11 V2.1
FORTRAN I/O IN REAL-TIME APPLICATIONS 17

MU BASIC-11/RT-11 V2
DISABLING CR/LF USING TTYSET - PATCH P 19

OS/8 V3D
RECENT VERSION NUMBERS FOR OS/8 V3D AND OS/8 COMBINED KIT V3D CUSPS 21

RPG II V8.02
RPGESP - THE RPG EDITOR 33

RSX-11M V3.2
CARD READER DRIVER (CRDRV) MAY CRASH SYSTEM 35
ANSI MAGTAPE ACP ALWAYS UNLOADS THE TAPE ON DISMOUNT 36
FORTRAN CALL TO CREATE/ELIMINATE GROUP GLOBAL EVENT FLAG FAIL 37
SYSGEN PROBLEMS 38

RSX-11S V2.2
OTL AND SAVE MODIFICATIONS 41

RSTS/E V06C
ENABLING LOW SEGMENT ERROR LOGGING UNDER RSTS/E V06C-RSTS/E PATCH 43

RT-11 V03B
MU BASIC ON RL01 BASED SYSTEMS FAILS AFTER ADDITION OF PATCH 51

TRAX
RESETTING SUPPORT ENVIRONMENT TERMINALS 53
SOFTWARE ERROR LOGGING AND SERCTL 55
RANDOM RECORD DELETIONS FROM INDEXED FILES 57
'DYNAMIC MEMORY EXHAUSTED' ERRORS 59

VAX/VMS
A DATA-DEPENDENT DMC-11 PROBLEM 73
ERROR ON LOCAL CONSOLE FLOPPIES 74
SUMMARY OF VAX-11 BLISS-32 PROBLEMS 75

ARTICLE OF THE WEEK

SWS introduces the DSM-11 Binary Program Update Service.

For details, See Page 3

digital
SOFTWARE SERVICES

INTERNAL DIGITAL USE ONLY

THE SMALL BUFFER . . . A WEEKLY PUBLICATION

The Small Buffer is a vehicle for informal, unofficial interchange among mini-midi software people. The Buffer can be used to alert others, on a timely basis, to methods, problems and solutions; it is for Internal Digital Use Only.

SUBCRIPTION:

Software Services personnel must be added/deleted by using the ADS (Automatic Distribution System) Order Forms.

All others may contact the Circulation Coordinator: **Tel:** **223-5886, or DECnet MP37**

WRITE: PK2/E49

Eleanor F. Hunter, Editor
223-5403

Clotelle Atkins, Publication Coordinator
223-3899

INTERNAL DIGITAL USE ONLY

Copyright © 1979 Digital Equipment Corporation

PERSONNEL CHANGES

UNITED STATES AREA

I am happy to report that JIM KNAPP will be assuming the job of unit Software Services Manager in the WNY Commercial Unit effective 11 Oct 1979. Jim will be in the Rochester, N.Y. office in his new duties.

Jim has been an outstanding contributor as a Senior Specialist since April, 1978. He brings a broad understanding of the Commercial/MDC business to his new job.

Please join me in wishing him success in his new position.

Jeff Holmes
Rochester

BASIC-PLUS-2 V1.5Ø
for RSTS/E V6C

BILL ALLENDOERFER, OPERATIONS GROUP, MK1-2/HØ2, 264-5911

TWO RSTS/E BASIC-PLUS-2 PATCHES WILL BE SUPERSEDED

THE BASIC-PLUS-2 V1.50 PATCHES FOR RSTS/E V6C, SEQ 38.3.17 AND SEQ 38.6.17, PUBLISHED IN THE JUNE 1979 RSTS/E SOFTWARE DISPATCH, AND ON PATCH KIT VERSION F OR LATER, WILL BE SUPERSEDED IN A FUTURE DISPATCH.

38.6.17 WITH AN .IDENT OF 04REA WILL GENERATE THE ERROR MESSAGE '?ILLEGAL I/O CHANNEL AT LINE <LINE> IN <MODULE NAME>' WHEN THE ECHO(N%) AND NOECHO(N%) FUNCTIONS ARE USED. 38.3.17 IS THE BASIC2 RUN-TIME SYSTEM COUNTERPART TO THAT PATCH.

NOTE THAT THE RPG UTILITY RPGESP.B2S USES THESE FUNCTIONS.

RPGII V8.Ø2
TRISH WING, OPERATIONS GROUP, MK1-1/HØ2, 264-5911

RPGESP - THE RPG EDITOR

There are two current problems under RSTS with the RPG editor.

1. RPGESP program will not work - error message Illegal I/O Channel occurs. This is due from BASIC+2 patches from the June dispatch or from RSTS patch kit F. There are two solutions to this problem.

   a. Change all occurances of ECHO and NOECHO functions in the source code.

   ```
   ECHO = SYS(CHR$(2%) + CHR$(K0%))
   NOECHO = SYS(CHR$(3%) + CHR$(K0%))
     where K0% is the channel number
   ```

   b. De-install patch #38.3.17 and #38.6.17 from BASIC+2

2. Blank lines are getting into the source file after editing. This problem is currently being checked. The blank line generates a warning message after compiling but you can still task build and execute the program successfully.

RSTS/E VØ6C

WAYNE JARRETT, TONY PRIBORSKY, DIGITAL DIAGNOSIS CENTER,
522-2717

ENABLING LOW SEGMENT ERROR LOGGING UNDER RSTS/E VØ6C - RSTS/E PATCH

Enabling low segment error logging can be a useful debugging tool for isolating problems in user job images for the various layered products supported under RSTS/E. An example of this technique can be found at the end of this article. It is assumed the reader is familiar with the RSTS patching utilities, namely ONLPAT and ODT. All numbers used are octal unless otherwise specified.

Normally, RSTS/E will only log errors which occur in the user high segment (Run Time System) or while the processor is in kernel mode (Monitor phase, FIP, etc.). Errors which occur as a result of a "Peek" sequence or errors which occur in a user low segment (Job image) are not logged. The routine which makes the determination of which error to log is CHKERL which is contained in the Flow control code (RTI) of the RSTS monitor. Thus, the basic function of CHKERL is to screen error traps which have occured as a result of a .PEEK or low segment error.

The following patch to the RSTS Executive will permit low segment errors to be logged in the system error log.

```
RUN $ONLPAT
Command file name? <LF>
File to patch? <LF>
File found in account [0,1]
Module name? RSTS <RETURN>
Base Address? RTI+34 <RETURN>
Offset address? 0 <RETURN>
 Base    Offset  Old      New
075274  000000  101722  ? ^C

Ready
```

The first step in the patching procedure is to determine the absolute address of RTI+34 in the RSTS module. In the above example this address is 075274. This address contains the branch instruction (BLOS) used to exit the CHKERL routine without logging an error (cf attached CHKERL listing). This instruction will eventually be changed to a NOP (240) to ensure that the RTS wants error logging and to

RSTS/E VØ6C
WAYNE JARRETT, TONY PRIBORSKY, DIGITAL DIAGNOSIS CENTER,
522-2717

provide a normal exit from CHKERL. This is important since the RT11 run-time system uses a Reserved Instruction trap (RI) sequences to determine whether floating point context switching is necessary (cf Cautions below).

To apply the patch, ODT (running in account [1,1]) is used to modify the instruction contained in RTI+34. The use of ODT to install (and deinstall) this 'patch' allows the user to turn on and off this 'extended' error logging feature. This modification is a temporary change to the currently running monitor and will be lost if the system is reloaded. The following procedure enables low segment error logging. The reverse procedure can be used to turn off extended error logging.

```
RUN $ODT        RSTS V06C-03 DEC DIAGN CEN
File<MEMORY>? <LF>

*75274/ 101772 240 <RETURN>
*^Z

Ready
```

As can be seen from the above example, ODT is used to examine location 075274 (RTI+34, found in step 1). This location contains the BLOS instruction (101772) with the CHKERL routine. A NOP (240) is now deposited to enable low segment error logging. To turn off this feature, a 101772 would be deposited.

*** CAUTION ***

For those processors without Floating Point hardware the RT11 run-time system should be added with the /NOLOGERR switch.

As was mentioned earlier, RT11 uses the RI trap to determine whether a system has floating point hardware. It does this by executing a CFCC (copy floating condition codes) instruction in the low segment during the RTS initialization phase and trapping this error. Consequently, if error logging is enabled for RT11 and this patch is installed, every time a user runs a RT11 based program a RI error will be logged. This can quickly fill up the system error log. It is strongly recommended that RT11 be added with error logging disabled before this patch is applied to the system.

RSTS/E VØ6C

WAYNE JARRETT, TONY PRIBORSKY, DIGITAL DIAGNOSIS CENTER,
522-2717

Below is a sample ERRDIS display for a RI trap under RT11. Error logging was enabled.

```
RI  Reserved Instr. Seq #37 Occurred on 01-Oct-79 at
15:49:55

User Description:
-----------------
Job Number                  12
KB Number                   1
Account                     [1,95]
Program Name                NONAME
User Job Physical Addr.     00714000
User Job Size               6K
Control Parameters          200
RTS Name                    RT11
RTS Physical Address        00674000

Detailed Description:
---------------------
R0                          000410
R1                          000000
R2                          000454
R3                          000000
R4                          174506
R5                          024126
Virtual PC                  000002
Physical PC              00714002
Processor Status            174000
Stack Pointer               027772
(SP)                        175430
(SP+2)                      175612
(PC-6)                      000034
(PC-4)                      000000
(PC-2)                      170000
(PC)                        000207
CPU ID                        -1
CPU ERR                     000000
```

Notice that the PC-2 contains the CFCC instruction (170000), which was in location 0 of the low segment. The PC constains a RTS PC (207). The PC had already been updated by the time the error trap had occured.

```
                                .SBTTL  CHECK FOR LOGGING AN ERROR

                                ;+
                                ;       CHKERL - CHECK FOR ERROR LOGGING
                                ;
                                ;       SP --> PC,PS
                                ;       CALLING JOB IS "JOB"
                                ;
                                ;       CALL CHKERL,R5
                                ;
                                ;       FIRST EXIT MEANS LOG ERROR
                                ;       SECOND EXIT MEANS DON'T LOG ERROR
                                ;
                                ;-

000000  010046                  CHKERL: MOV     R0,-(SP)        ;;; GET A WORKING REGISTER TO
000002  016600  000004                  MOV     2*2(SP),R0      ;;; HOLD THE PC+2 OF THE TRAP
000006  005766  000006                  TST     3*2(SP)         ;;; FROM USER MODE?
000012  100406                          BMI     30$             ;;; YES, CHECK FOR HIGH SEGMENT
000014  020027  005114'                 CMP     R0,#ERLPEK      ;;; FROM ERROR LOG OR .PEEK PEEK?
000020  001001                          BNE     20$             ;;; NO, LOG ERROR, TAKE FIRST EXIT
000022  005725                  10$:    TST     (R5)+           ;;; DON'T LOG ERROR, TAKE SECOND EXIT
000024  012600                  20$:    MOV     (SP)+,R0        ;;; RESTORE WORKING REGISTER
000026                                  RETURN  R5              ;;; AND EXIT
000026  000205                          RTS     R5

000030  020067  000000G         30$:    CMP     R0,MAXHGH       ;;; WAS TRAP FROM HIGH SEGMENT?
000034  101772                          BLOS    10$             ;;; NO, TAKE SECOND EXIT, NO ERROR LOG
000036  016700  001022                  MOV     JOBRTS,R0       ;;; GET JOB'S RTS POINTER
000042  032760  004000  000036          BIT     #PF.NER,R.FLAG(R0);;; RTS WANTS ERROR LOGGING?
000050  001765                          BEQ     20$             ;;; YES, SO TAKE FIRST EXIT
000052  000763                          BR      10$             ;;; NO, SO TAKE SECOND EXIT

000054                          GLOBAL  <MAXHGH>
```

RSTS/E VØ6C
WAYNE JARRETT, TONY PRIBORSKY, DIGITAL DIAGNOSIS CENTER,
522-2717

DEBUGGING EXAMPLE

Consider the situation where a user receives a memory management violation when running a RSX task. Many system managers would immediately call Field Service with a hardware problem. Using the above technique, the problem can be quickly be isolated to the failing component, namely hardware or software.

The following program is a very simple RSX macro program. It was assembled using the RSX macro assembler (MAC.TSK) and task built using TKB.TSK.

```
RUN TEST.TSK
?Memory management violation
010000  000000  000000  000000  000000  000000  002000
002022

Ready
```

PROGRAM LISTING

```
 1 000000                          START:
 2
 3 000000  005000                  CLR     R0
 4 000002  005001                  CLR     R1
 5 000004  005002                  CLR     R2
 6 000006  005003                  CLR     R3
 7 000010  005004                  CLR     R4
 8 000012  005005                  CLR     R5
 9
10 000014  012700  010000          MOV     #10000,R0
11 000020  005010                  CLR     (R0)
12
13 000022  000000                  HALT
14         000000'                 .END START
```

RSTS/E VØ6C

WAYNE JARRETT, TONY PRIBORSKY, DIGITAL DIAGNOSIS CENTER,
522-2717

ERROR REPORT

KT Memory Mgmt. SEQ #4 Occurred on 03-OCT-79 at 09:34:08

User Description

Job Number	6
KB Number	17
Account	[1,2]
Program Name	TEST
User Job Physical Addr.	01054000
User Job Size	1K
Control Parameters	200
RTS Name	RSX
RTS Physical Address	00324000

Detailed Description:

R0	010000
R1	000000
R2	000000
R3	000000
R4	000000
R5	000000
Virtual PC	002022
Physical PC	01056022
Processor Status	174000
Stack Pointer	002000
(SP)	005000
(SP+2)	005001
(PC-6)	012700
(PC-4)	010000
(PC-2)	005010
(PC)	000000
CPU ID	-1
CPU ERR	000000

Contents of Registers

MMUSR0	040141	Segment Lgth
UISDR0	017506	Read access Write access
USIAR0	010540	

From the above error listing we know that the user job is mapped (APR0) in a virtual address space of 0 to 3777 (1K Job size). In this case the instruction which caused the memory management violation is CLR (R0) contained in (PC-2). Since R0 is 10000 we were attempting to clear a memory location outside our valid address range but within the mapping capability of APR0 (recall that APR0 can map from 0

RSTS/E VØ6C

WAYNE JARRETT, TONY PRIBORSKY, DIGITAL DIAGNOSIS CENTER,
522-2717

- 17777). Thus a segment length error was generated by the memory management hardware rather than a non-resident page error.

With this simple example one can see that this technique can be used to isolate problems down to the failing hardware or software component. In this case, it is clearly a coding problem. The advantage in using the system error log rather than the RSX error broadcast is that additional information is captured ie the last two words on the stack, the last three instructions executed, system error registers etc.

```
                        .SBTTL  CHECK FOR LOGGING AN ERROR

                        ;+
                        ;       CHKERL - CHECK FOR ERROR LOGGING
                        ;
                        ;       SP --> PC,PS
                        ;       CALLING JOB IS "JOB"
                        ;
                        ;       CALL CHKERL,R5
                        ;
                        ;       FIRST EXIT MEANS LOG ERROR
                        ;       SECOND EXIT MEANS DON'T LOG ERROR
                        ;
                        ;-

000000  010046                CHKERL: MOV     R0,-(SP)        ;;; GET A WORKING REGISTER TO
000002  016600  000004              MOV     2*2(SP),R0      ;;; HOLD THE PC+2 OF THE TRAP
000006  005766  000006              TST     3*2(SP)         ;;; FROM USER MODE?
000012  100406                      BMI     30$             ;;; YES, CHECK FOR HIGH SEGMENT
000014  020027  005114'              CMP     R0,#ERLPEK      ;;; FROM ERROR LOG OR .PEEK PEEK?
000020  001001                      BNE     20$             ;;; NO, LOG ERROR, TAKE FIRST EXIT
000022  005725                10$:  TST     (R5)+           ;;; DON'T LOG ERROR, TAKE SECOND EXIT
000024  012600                20$:  MOV     (SP)+,R0        ;;; RESTORE WORKING REGISTER
000026                              RETURN  R5              ;;; AND EXIT
000026  000205                      RTS     R5

000030  020067  000000G         30$:  CMP     R0,MAXHGH       ;;; WAS TRAP FROM HIGH SEGMENT?
000034  101772                      BLOS    10$             ;;; NO, TAKE SECOND EXIT, NO ERROR LOG
000036  016700  001022              MOV     JOBRTS,R0       ;;; GET JOB'S RTS POINTER
000042  032760  004000  000036      BIT     #PF.NER,R.FLAG(R0);;; RTS WANTS ERROR LOGGING?
000050  001765                      BEQ     20$             ;;; YES, SO TAKE FIRST EXIT
000052  000763                      BR      10$             ;;; NO, SO TAKE SECOND EXIT

000054                      GLOBAL  <MAXHGH>
```

digital

INTERDFFICE MEMORANDUM

DATE: 28-Mar-79
FROM: Stephen Kurtzman
DEPT: RSTS/E Development
DTN: 264-6136
LOC: MK1/2L2

TO: RSTS/E Group

CC: Barbara Cochrane MK1/2C5
Gene Cronin
David Foetz
Bill Hall PK3-1/M21
Mike Marcil PK3-1/M21
Jack Rhine MK1/2C5
Nancy Van Winkle MK1/2M3

SUBJ: SSSHHHHHHH.... ANOTHER SYS CALL GOT INTO RSTS/E V7.0

Don't tell anyone (accept our customers) another SYS call got into the bnae level. Due to an unfortunate turn of events and after hours projects we now have a one-shot spool packet.

To make things easier on our users, and ourselves when QUEMAN decides change its interface the one-shot spooling packet was implemented as a UUO call to the RSTS/E monitor. Among other benefits it allows user applications to spool files without having to chain to QUE.

The following checks are done when the monitor receives a spool request.

1) Check the file name to be spooled for legal RSTS format.

2) Check to make sure that the device which the file is located on is a mounted RSTS/E disk and that the user has all necessary privileges to access that disk.

3) If the file specification does not contain any wild card fields check for the existance of the file and that the user has appropriate privilege to access that file.

4) Before sending the message to QUEMAN the monitor does all the appropriate send/receive buffer quota checks, and furthermore also makes sure that QUEMAN is not hibernating.

The above checks should trap all normal error conditions that might occur when spooling a file.

The format of the call is as follows:

Data passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(-28%), the one-shot spool request UUO subcode.
3-4	Unused.
5-6	Project programmer number of the file to spool. If both bytes are zero, the current user's account is used.
7-10	File name in Radix-50.
11-12	Extension in Radix-50.
13-14	Two byte ASCII device name to spool the file to. If both bytes are zero, "LP" is assumed.
15	Unit number for the spooling device specified in bytes 13-14.
16	Unit number flag for the spooling device that was specified in bytes 13-14.
17-18	Reserved.
19-20	Flag word to be passed to QUEMAN. The bits in this word have the following meaning: 1% = File should be spooled using FORTRAN carriage control. 2% = Restart bit. Equivalent to using the /RE option with QUE. 4% = Delete file after spooling. Equivalent to using the /DE option with QUE. 8% = Binary file. Equivalent to using the /BI option with QUE. 16% = End bit. Equivalent to using the /END option with QUE. 32% = No header bit. Equivalent to the /NH option in QUE.
21-22	Unused. But should be zero.
23-24	Device name where the file which is being spooled can be found; must be disk. If both bytes are zero, SY: (the public structure) is used.

25	Unit number of the device where the file which is being spooled can be found.
26	Unit number flag for the device where the file which is being spooled can be found.
27-30	Unused.

Possible Errors:

Meaning	ERR value
?ILLEGAL FILE NAME The file name passed in bytes 7 through 1P is not a valid RSTS/E file name.	2
?CAN'T FIND FILE OR ACCOUNT The file name or account passed could not be found on the disk specified. Or, QUEMAN is not currently a message receiver.	5
?PROTECTION VIOLATION One of two possible errors has occurred: 1) The user does not have the appropriate privileges to access the file requested. 2) The user does not have the appropriate privileges to send a message to QUEMAN.	10
?DISK PACK IS LOCKED OUT The disk on which the file resides is mounted but has been temporarily disabled.	22
?NO BUFFER SPACE AVAILABLE QUEMAN has reached its buffer quota and cannot accept any more messages until the currently pending requests are processed, or there is not enough space left in the monitor buffer pool to send the message to QUEMAN.	32

PATCH OVR

Base .. SPLX

OFFSET Ø

old Ø new 24Ø

RSTS/E UPDATE

OVR

Overview of RSTS/E V7.0

Overview of RSTS/E V7.0

CONTENTS

Introduction OVR-1

New Hardware Support OVR-3
Disks..... OVR-3
Terminals..... OVR-5

Additional VT100 Documentation..... OVR-12
Definition of Each VT100 Set-up Feature..... OVR-12
ANSI/VT52 Mode..... OVR-12
Answerback Message..... OVR-12
Auto Repeat..... OVR-12
Auto XON/XOFF..... OVR-13
Bits Per Character..... OVR-14
Characters Per Line..... OVR-14
Cursor..... OVR-14
Interlace..... OVR-15
Keyclick Tone..... OVR-15
LINE/LOCAL..... OVR-15
Margin Bell..... OVR-15
New Line..... OVR-16
Parity..... OVR-16
Parity Sense..... OVR-16
Power..... OVR-17
Receive Speed..... OVR-17
Screen Background..... OVR-17
Screen Brightness..... OVR-17
Scroll..... OVR-18
Tabs..... OVR-19
Transmit Speed..... OVR-18
Wraparound..... OVR-19
Magnetic Tapes..... OVR-20

Obsolete Hardware..... OVR-22

New Operating System Features..... OVR-23
Large Files..... OVR-23
Data Caching..... OVR-23
RSX Directives in Monitor..... OVR-24
Resident Libraries..... OVR-24

User Logical Assignments with PPN's..... OVR-25
Placed Files..... OVR-25
Tentative Files..... OVR-25
Conditionally Contiguous Files..... OVR-25
No File Supersede..... OVR-26
Zero a Disk File..... OVR-26
Make a Contiguous File Non-Contiguous..... OVR-26
Library Segmentation..... OVR-27
ANSI Magnetic Tape Support..... OVR-27
BASIC-PLUS-2 Only Systems..... OVR-27
COBOL Only Systems..... OVR-28

Utility Programs..... OVR-29
New Utilities..... OVR-29
Discontinued Utilities..... OVR-30

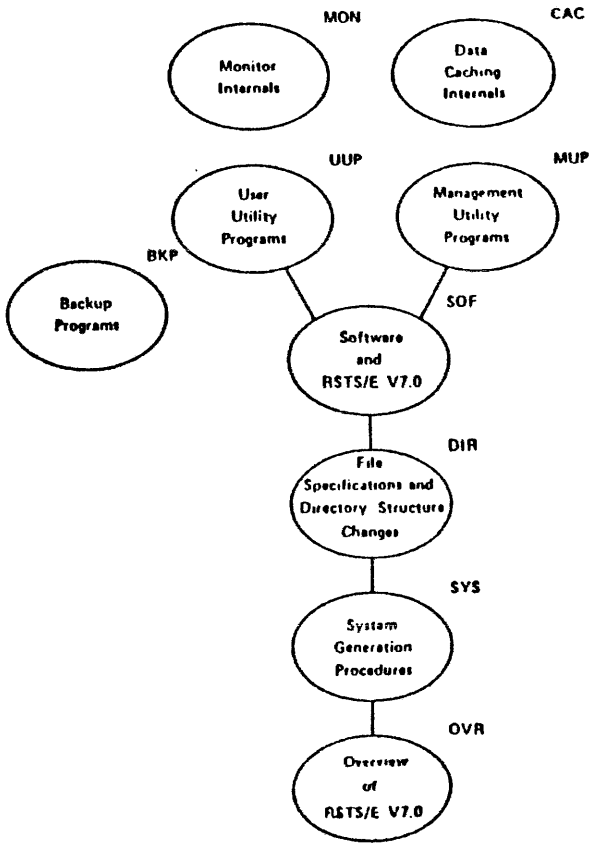
Bundled Software..... OVR-31

Unbundled Software..... OVR-32

Category C Software..... OVR-32

Documentation..... OVR-33

COURSE MAP



Overview of RSTS/E V7.0

INTRODUCTION

RSTS/E V7.0 is a major feature release of the operating system. Performance improvements, enhanced functionality, and new hardware support have been included while maintaining upward compatibility with previous versions of the RSTS/E operating system.

- Have you been concerned with long task build times?
- Have you been concerned by disk I/O throughput time?
- Have you been concerned with the 65K block limitation for files?
- Have you been concerned with COBOL performance?
- Have you been concerned with the limitations of BACKUP?

RSTS/E V7.0 addresses all of the above and more.

This module covers the major features of RSTS/E V7.0 at a conceptual level. The modules that follow address these new features in more detail.

This page is for notes.

Overview of RSTS/E V7.0

NEW HARDWARE SUPPORT

Disks

1. RL02
   - Up to 4 RL01/RL02 disks on a system
2. RX02 - floppy disk
   - Double or single density user selectable
   - Single density is RX01 compatible
   - Uses RX01-K diskettes
   - DMA access device
   - Supported as non-file structured device only
   - Up to 8 RX01/RX02 diskettes on a system
   - Improved diagnostic capability

Table OVR-1. Disk Characteristics

DISK NAME	RL02	RX02 DOUBLE DENSITY	RX02 SINGLE DENSITY
DISK TYPE	CARTRIDGE	DOUBLE DENSITY FLOPPY	SINGLE DENSITY FLOPPY
No. of Surfaces/disk	2	1	1
No. of Blocks/disk	20460	1782	1782
No. of Words/disk	5.2M	256K	128K
Device Clustersize	1	N/A	N/A
Max. No. of Drives	4	8	8
Access Time	100 Milli-seconds	Average 262 Milliseconds	Average 483 Milliseconds
Transfer Time	3.9 words/mi-crosecond	Peak 62kb/ second	Peak 31 kb/ second

Terminals

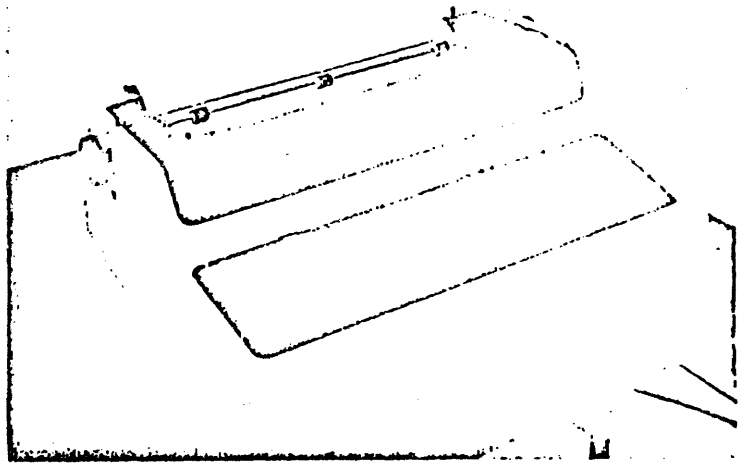


Figure OVR-1. LA34-DECwriter IV

- LA36 equivalent
- Up to 300 baud
- Up to 220 columns (depending on print size)
- Set up mode allows user selectable terminal functions

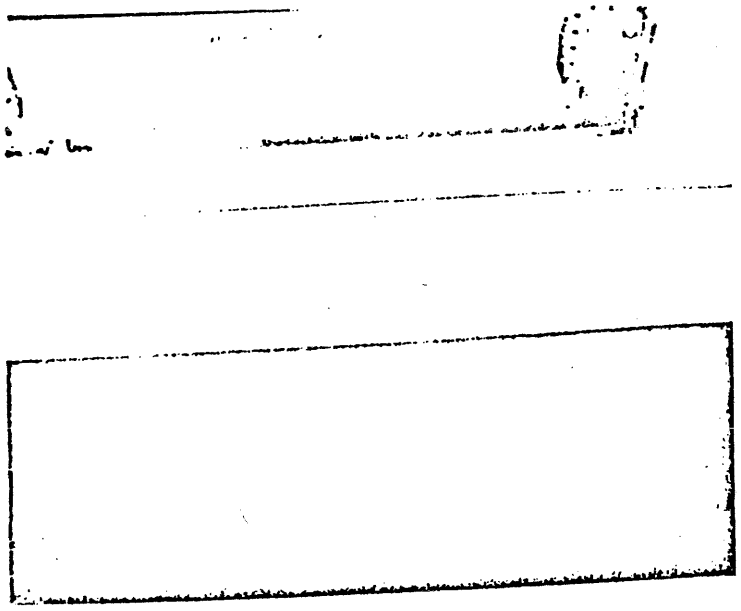
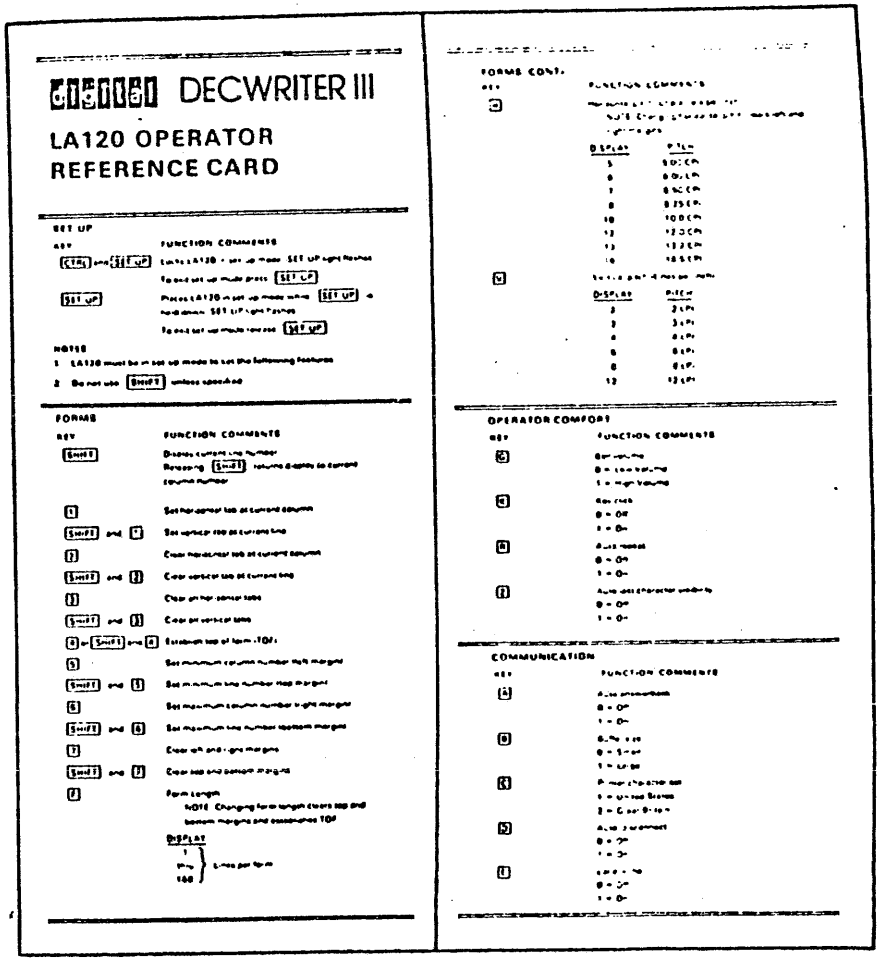


Figure OVR-4. LA120 - DECwriter III Keyboard



DECWRITER III

LA120 OPERATOR
REFERENCE CARD

SET UP

KEY	FUNCTION/COMMENTS
CTRL and SET UP	Enters LA120 set up mode. SET UP light flashes
	To exit set up mode press SET UP
SET UP	Places LA120 in set up mode when SET UP is held down. SET UP light flashes
	To exit set up mode release SET UP

NOTES
1. LA120 must be in set up mode to set the following features
2. Do not use SHIFT unless specified

FORMS

KEY	FUNCTION/COMMENTS
SHIFT	Displays current line number. Pressing SHIFT returns display to current column number
1	Set horizontal tab at current column
SHIFT and 1	Set vertical tab at current line
2	Clear horizontal tab at current column
SHIFT and 2	Clear vertical tab at current line
3	Clear all horizontal tabs
SHIFT and 3	Clear all vertical tabs
4 or SHIFT and 4	Establish top of form (TOF)
5	Set maximum column number left margin
SHIFT and 5	Set maximum line number top margin
6	Set maximum column number right margin
SHIFT and 6	Set maximum line number bottom margin
7	Clear left and right margins
SHIFT and 7	Clear top and bottom margins
8	Form Length NOTE: Changing form length clears top and bottom margins and establishes TOF DISPLAY 1 thru 132 = Lines per form

FORMS CONT.

KEY	FUNCTION/COMMENTS
9	Horizontal tab stops every n columns NOTE: Clears all tabs, sets TOF, clears top and bottom margins, sets left margin at 1

DISPLAY / PITCH

DISPLAY	PITCH
5	5.0 CPI
6	6.0 CPI
7	6.86 CPI
8	8.33 CPI
10	10.0 CPI
12	12.0 CPI
13	13.2 CPI
16	16.5 CPI

KEY	FUNCTION/COMMENTS
V	Vertical tabs every n lines

DISPLAY / PITCH

DISPLAY	PITCH
2	2 LPI
3	3 LPI
4	4 LPI
6	6 LPI
8	8 LPI
12	12 LPI

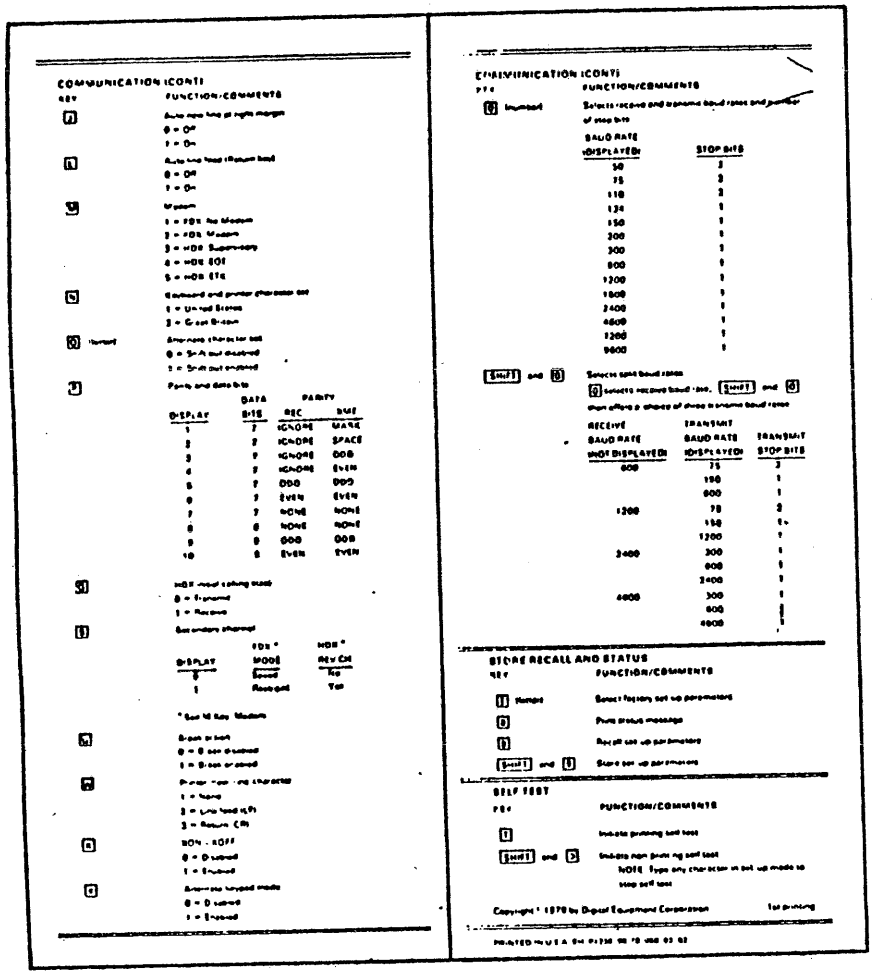
OPERATOR COMFORT

KEY	FUNCTION/COMMENTS
Q	Set volume 0 = Low volume 1 = High volume
R	Auto CRLF 0 = Off 1 = On
S	Auto repeat 0 = Off 1 = On
T	Auto underscore underline 0 = Off 1 = On

COMMUNICATION

KEY	FUNCTION/COMMENTS
A	Auto pronunciation 0 = Off 1 = On
B	Baud rate 0 = 5 per 1 = 10 per
C	Answer back code set 1 = United States 2 = Canadian
D	Auto turnaround 0 = Off 1 = On
E	Local echo 0 = Off 1 = On

Figure OVR-5. LA120 Operator Reference Card (page 1 of 2)



COMMUNICATION (CONT)

KEY	FUNCTION/COMMENTS
[7]	Auto new line at right margin 0 = Off 1 = On
[1]	Auto line feed (Return key) 0 = Off 1 = On
[2]	Modem 1 = RS232 No Modem 2 = RS232 Modem 3 = HDX Supervisory 4 = HDX EOT 5 = HDX ETX
[5]	Keyboard and printer character set 1 = United States 2 = Great Britain
[0] Numpad	Answerback character set 0 = Shift out disabled 1 = Shift out enabled
[3]	Parity and data bits

DISPLAY	DATA BITS	PARITY REC	PARITY XMIT
1	7	IGNORE	MARK
2	7	IGNORE	SPACE
3	7	IGNORE	ODD
4	7	IGNORE	EVEN
5	7	ODD	ODD
6	7	EVEN	EVEN
7	7	NONE	NONE
8	8	NONE	NONE
9	8	ODD	ODD
10	8	EVEN	EVEN

KEY	FUNCTION/COMMENTS
[4]	HDX mode talking mode 0 = Transmit 1 = Receive
[8]	Set answerback message

DISPLAY	FDX* MODE	HDX* REV CH
0	Send	No
1	Receive	Yes

* Set 10 Key Modem

KEY	FUNCTION/COMMENTS
[6]	Break on key 0 = Break disabled 1 = Break enabled
[9]	Printer new line character 1 = None 2 = Line feed (LF) 3 = Return (CR)
[0]	XON - XOFF 0 = Disabled 1 = Enabled
[0]	Answerback transmit mode 0 = Disabled 1 = Enabled

COMMUNICATION (CONT)

KEY	FUNCTION/COMMENTS
[0] Numpad	Selects receive and transmit baud rates and number of stop bits

BAUD RATE DISPLAYED	STOP BITS
50	2
75	2
110	2
134	1
150	1
200	1
300	1
600	1
1200	1
1800	1
2400	1
4800	1
7200	1
9600	1

[SHIFT] and [0] Selects xmit baud rates

[0] selects receive baud rate, [SHIFT] and [0] then offers a choice of those transmit baud rates

RECEIVE BAUD RATE NOT DISPLAYED	TRANSMIT BAUD RATE DISPLAYED	TRANSMIT STOP BITS
600	75	2
	150	1
	600	1
1200	75	2
	150	1
	1200	1
2400	300	1
	600	1
	2400	1
4800	300	1
	600	1
	4800	1

STORE RECALL AND STATUS

KEY	FUNCTION/COMMENTS
[1] Numpad	Select factory set up parameters
[0]	Print status message
[1]	Recall set up parameters
[SHIFT] and [1]	Store set up parameters

SELF TEST

KEY	FUNCTION/COMMENTS
[1]	Initiate printing self test
[SHIFT] and [2]	Initiate non printing self test NOTE: Type any character to exit set up mode to stop self test

Copyright © 1979 by Digital Equipment Corporation

Figure OVR-5. LA120 Operator Reference Card (page 2 of 2)

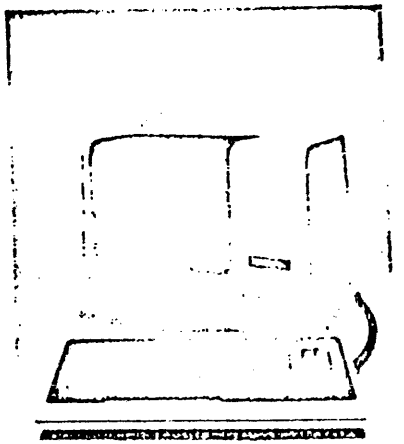


Figure OVR-6. VT100

- VT52 replacement
- Supported in VT52 mode
- Up to 9600 baud
- Screen
  - 132 columns x 14 lines
  - 80 columns x 24 lines
- Set up mode allows user selectable terminal functions
- Advanced video option includes:
  - 132 columns x 24 lines
  - Ability to combine attributes
  - Full terminal capability

ADDITIONAL VT100 DOCUMENTATION

Definition of Each VT100 Set-Up Feature

This section describes each SET-UP feature in detail (in alphabetical order) and states how each feature affects the terminal.

> **Note**
>
> Unless otherwise stated, entering SET-UP mode and changing features does not result in the loss of data displayed on the screen.

ANSI/VT52 Mode

The VT100 terminal is compatible with two different programming standards - American National Standards Institute (ANSI) and VT52. In ANSI mode, the VT100 will generate and respond to coded sequences per ANSI standard X3.41 - 1974. In VT52 mode, the VT100 terminal is compatible with previous DIGITAL software using the VT52 video terminal.

Answerback Message

The VT100 answerback feature provides the terminal with the capability to identify itself to the host. Answerback is a question and answer sequence where the host computer asks the terminal to identify itself. The terminal sends its unique identifier, or answerback message, to verify the identity of the terminal. The entire answerback sequence takes place automatically without affecting the screen or requiring operator action.

Auto Repeat

The auto repeat feature allows a key to be automatically repeated at the rate of about 30 characters per second when the key is held down for more than one-half

second. The auto repeat feature affects all keyboard keys except the following:

SET-UP
ESC
NO SCROLL
TAB
RETURN
ENTER
CTRL and any key

Auto XON/XOFF

The VT100 is capable of automatically generating synchronizing codes XON (DC1) and XOFF (DC3). The XOFF control sequence is used to stop the transmission of data from the computer to the terminal; the XON sequence is used to resume transmission. With the feature enabled, the VT100 will automatically generate the XOFF code when one of the following events occur:

1. The internal buffer is nearly full
2. The NO SCROLL key is pressed
3. The terminal is placed in SET-UP mode

When either the buffer empties, the NO SCROLL key is pressed again, or the terminal is taken out of SET-UP mode, the VT100 will automatically transmit the XON code to resume transmission from the computer to the terminal.

If the host computer software does not support the XON/XOFF codes, data sent during buffer full conditions, or when the terminal is in SET-UP mode, will be lost.

> **Note**
>
> The VT100 will always stop transmission when an XOFF (DC3) code is received and will resume transmission when an XON (DC1) code is received, regardless of the AUTO XON/XOFF feature setting.

Bits Per Character

This feature allows the terminal to transmit and receive 7 or 8-bit characters. In most cases, the terminal will be set for 7-bit operation (normal ASCII standard characters). In those cases where the communication protocol requires 8-bit characters, bit 8 is set to a space (or 0) for characters transmitted and is ignored on all characters received.

Characters Per Line

The VT100 is capable of displaying either 80 or 132 characters per line. In the 80 character per line mode, the screen is 80 characters wide by 24 lines high. In the 132 character per line mode, the screen is 132 characters wide by 14 lines high. In the 132 character per line mode, the displayed lines are physically the same width as in the 80 character per line mode, but the characters are more compact.

> **Note**
>
> When changing from an 80 to 132 character per line mode or vice-versa, the current contents of the screen are lost.

Cursor

The VT100 offers a choice of two cursors to indicate the "active positions" or where the next character will be placed on the screen. The cursor may be displayed as either a blinking underline (_) or a blinking block (■).

Interlace

Interlace changes the scanning rate of the screen to allow for alternating scans of odd and even rows in the character. This feature is used for high resolution memory options. When in the alphanumeric mode, the interlace feature should be turned off since the use of the smooth scroll feature with the interlace feature may cause the display to appear jittery and difficult to read when the screen is scrolling.

Keyclick Tone

The keyclick is a tone which is generated every time a key is pressed. The keyclick may be turned on or off to suit the operator's needs. However, research and experience has shown that an operator is more accurate when there is an audible feedback from the keyboard.

Like the bell tone, the keyclick volume is not adjustable.

LINE/LOCAL

The LINE/LOCAL feature allows the operator to easily place the terminal in either an ON-LINE or LOCAL (off-line) condition. When the terminal is on-line (the keyboard ON-LINE indicator is ON) all characters typed on the keyboard are sent directly to the computer and messages from the computer are displayed on the screen. In the LOCAL condition (the keyboard LOCAL indicator is ON), the terminal is electrically disconnected from the computer, messages are not sent to or received from the computer, and characters typed on the keyboard are echoed on the screen directly.

Margin Bell

The margin bell feature is much the same as the bell in a typewriter. When the cursor is eight characters from the end of the current line, the VT100 sounds a tone to alert the operator.

New Line

The new line feature enables the RETURN key on the terminal to function like the RETURN key on an electric typewriter. When the new line feature is enabled, pressing the RETURN key generates the carriage return (CR) and line feed (LF) codes. When a line feed code is received, the code is interpreted as a carriage return and line feed.

When the new line feature is disabled, the RETURN key generates only the CR code; an LF code causes the terminal to perform a line feed only.

> **Note**
>
> If double line feeds occur consistently, turn this feature off since the computer is already performing this function automatically.

Parity

When enabled, parity checks for correct data transmission. If a transmission error occurs, the VT100 can detect it and indicate its presence by placing a checkerboard character (▒) on the screen in place of the character with the error. The parity sense feature determines if the parity is even or odd.

Parity Sense

The parity sense feature defines which of the two methods of parity checking, odd or even, is being used by the VT100. If the parity feature is on, the terminal's parity sense must be matched to the parity the computer is sending. If the parity sense features do not match, every character sent to the computer will be rejected even though the character was received correctly by the VT100. If a parity incompatibility occurs, the checkerboard character will be shown on the screen in place of the received character.

> **Note**
>
> If the parity feature is turned off, the parity sense selection will be totally disregarded.

Power

During the initial installation, the terminal display must be synchronized to the power line frequency. In the U.S. this is set to 60 hertz.

Receive Speed

The receive speed determines how fast the characters sent by the computer can be accepted by the terminal. The VT100 is capable of receiving at any one of the following preselected speeds: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, and 9600 baud.

The receive speed is independent of the transmit speed; the terminal may receive data at one speed and transmit data at a totally different speed.

Screen Background

The screen background feature of the VT100 allows the operator to determine the background of the screen. In the normal screen mode, the display contains light characters on a dark background; in the reverse screen mode, the display contains dark characters on a light background.

Screen Brightness

Unlike most video terminals, the VT100 does not contain switches or knobs to adjust screen brightness. Instead, the VT100 electronically controls the screen brightness. This feature eliminates the high failure rate of mechanical knobs and allows the operator to select the desired level of brightness for maximum comfort under varied lighting conditions. This setting may be saved like any other feature in the terminal.

Scroll

Scrolling is the upward or downward movement of existing lines on the screen to make room for new lines at the bottom or top of the screen. It can be performed in two ways: jump scroll or smooth scroll. In jump scroll mode, new lines appear on the screen as fast as the computer sends them to the terminal. At the higher baud rates, the data is very difficult to read due to the rapid upward movement of the lines. In smooth scroll mode, a limit is placed on the speed at which new lines of data may be sent to the terminal. The upward movement of lines occurs at a smooth steady rate allowing the data to be read as it appears on the screen.

> **Note**
>
> Smooth scroll mode allows a maximum of six lines of data per second to be added to the screen. The Auto XON/XOFF feature must be enabled and supported by the host computer to ensure that data is not lost when smooth scroll mode is enabled.

Tabs

Just like a typewriter, the VT100 can jump or tab to preselected points on a line. These tab stops may be individually changed, or totally cleared and then reset.

Transmit Speed

Transmit speed determines how fast the characters will be sent from the terminal to the computer. The VT100 is capable of transmitting at any one of the following preselected transmit speeds: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, and 9600 baud.

Transmit speed is independent of receive speed; the terminal may transmit data at one speed and receive data at a different speed.

Wraparound

The wraparound feature removes the maximum limit on the number of characters that may be entered for one line. When this feature is enabled, the 81st or 133rd character (depending upon the line size selected) typed on a line is automatically placed in the first character position of the next line without a carriage return (CR) and a line feed (LF) code is generated. If the wraparound feature was not enabled, the 81st or 133rd character and all following characters would be overwritten into the last character position of the current line until a CR and an LF code are generated.

> **Note**
>
> Since wraparound is a VT100 terminal feature dealing with physical lines, the CR and LF codes must still be generated to end the logical line.

(shifted)

The VT100 contains character sets for the U.S. and the United Kingdom. The difference between the two character sets is one character, the # or £ symbol. When the U.S. character set is selected, the uppercase 3 key on the main keyboard displays the # character. The £ character is displayed when the U.K. character set is selected.

Magnetic Tapes

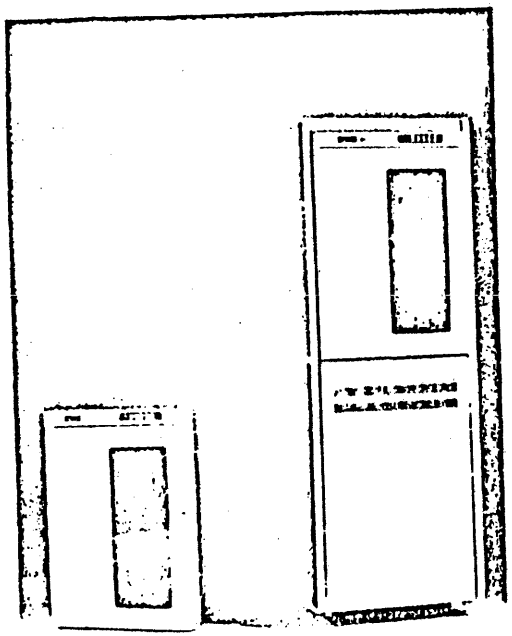


Figure OVR-7. TS04

- 9-track magtape tape
- 800 or 1600 BPI preset at factory
- Up to 8 units on a system

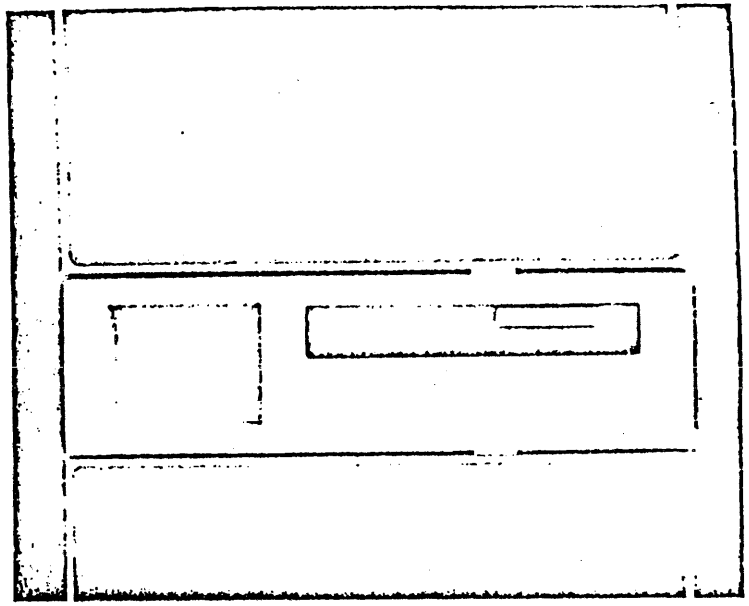


Figure OVR-8. TU58 - DECtape II

- Low cost tape cartridge
- Random access to 512 blocks of 512 bytes each
- 10 second average access time
- Up to 2 drives per controller
- Up to 4 controllers/8 units on a system
- Uses a DL11A terminal interface

KMC11

- A microprocessor used by the 3271 package
- Up to 16 allowed on a system

OBSOLETE HARDWARE

- RC11 - fixed head disk
- LK11 - hardware which forced device interrupts

NEW OPERATING SYSTEM FEATURES

Large files

1. System generation option
2. Data files limited only by the size of a disk
3. Resource cost
   - 2K words in monitor
   - One additional small buffer for each file opened, no additional buffers for multiple opens of the same file.
4. Benefits include:
   - Extension of non-contiguous files opened for update/write-shared access
   - Access to non-file structured disk by block rather than device cluster number
   - Some performance improvement

Data Caching

1. System generation option
2. Data, as well as directories, can be cached in memory
3. User selectable random or sequential data caching (privileged operation)
4. Resource cost
   - Minimum 2K words of XBUF
   - Heavier CPU usage
5. Benefits
   - Overall I/O throughput improvement on disk bound systems

RSX Directives In Monitor

1. System generation option
2. User jobs do not have the RSX Run-Time System as part of the user virtual space.
3. Resource cost
   - o 1K words in the monitor
4. Benefits
   - o 31K word user space when using the RSX Run-Time System
   - o 24K word user space when using RMS resident library

Resident Libraries

1. System generation option
2. RMS routines memory resident and shareable by multiple users.
3. Recommended for systems with a minimum of 124K words and applications that use RMS-11K.
4. Resource cost for RMS Resident Library
   - o Up to 24K words of physical memory
   - o 8K words in user virtual space
   - o RSX directives in the monitor are required
5. Benefits for RMS Resident Library
   - o Reduced task build time
   - o Smaller task images on disk
   - o Faster execution speed - reduced overlaying
   - o Overall performance improvements on systems with RMS-11 applications

User Logical Assignments with PPN's

1. Logical assignments that associate a device and PPN with the logical assignment
2. Benefits
   - o Less programming restrictions
3. Limitations
   - o Only three logical assignments available if a PPN is included

Placed Files

1. First block of a file can be positioned at or near a specified physical disk location.
2. Benefits
   - o Optimal placement of files to minimize head movement on disks

Tentative Files

1. Files can be created without becoming permanent until closed.
2. Benefits
   - o Less program code
   - o Data security in case of a program abort or a system crash

Conditionally Contiguous Files

1. Files are created contiguous or non-contiguous, based on available disk space.

2. Benefits

   - Less error handling overhead

No File Supersede

1. Files opened for output return an error if a file with the same name already exists

2. Benefits

   - Less program code
   - Greater file protection

Zero a Disk File

1. Null characters are written over the specified file if it is explicitly or implicitly deleted.

2. Privileged operation that can be specified at open time

3. Benefits

   - Security for files containing sensitive information

4. Limitations

   - This operation on large files can be very time consuming.

Make a Contiguous File Non-Contiguous

1. Contiguous files can be made non-contiguous while a file is being accessed by multiple users.

2. Benefits

   - Ability to extend a contiguous file while application programs are executing

Library Segmentation

1. Utility programs and their associated files can be stored in accounts other than [1,2] .

2. Benefits

   - Faster access to utility programs
   - Reduced overhead on the system library account

ANSI Magnetic Tape Support

1. Multi-volume magnetic tape support through PIP.SAV only.

2. Benefits

   - Transportability of RMS-11 Sequential files between RSTS/E and VAX or RSX-11M
   - Allows selective backup of large files to tape

3. Limitations

   - RMS-11 Relative and Indexed files transportable only to other RSTS/E systems

BASIC-PLUS-2 Only Systems

1. BP2COM can be installed as the system default run-time system.

2. Benefits

   - Utility programs as task images
   - Less memory used for the system default run-time system

3. Limitations

   - No immediate mode

COBOL Only Systems

1. RSX is the system default run-time system
2. Benefits
   - o No need for BASIC-PLUS or BASIC-PLUS-2
   - o No need for RMS-11 Run-Time System if using the RMS resident library
   - o Utility programs as task images

UTILITY PROGRAMS

All BASIC-PLUS utility programs can be compiled and executed under BASIC-PLUS-2 without translation.

New Utilities

1. SAVRES.SAV
   - o Fast volume save/restore
   - o Image copy of disks
   - o Executes stand alone or during time-sharing
   - o Handles bad blocks
   - o On-line for unmounted disks
   - o Off-line for system disk
2. ONLCLN.SAV
   - o On-line disk clean utility
   - o More thorough disk clean than the SYS call clean
3. MAKSIL.TSK
   - o Converts task images into resident library format
   - o Converts task images or .SAV images into run-time system format
4. FIT
   - o Supersedes PIPEXT.BAS
   - o File transfer program for DECtape II, floppy disk, RT11 disks and DOS disks
5. DEVTST
   - o Supersedes UETP
   - o Device testing package
6. RMSIFL
   - o Performs fast loading of indexed files

CSPCOM (CUSP. Compiler) Program

The CUSP COMPILER

A new feature of RSTS/E V7.0, CSPCOM (the CUSP Compiler) generates object modules from the BASIC-PLUS utility programs that can be linked against the RSX Run-Time System, eliminating the need for the BASIC-PLUS Run-Time System on those systems which would not otherwise use BASIC-PLUS or BASIC-PLUS-2.

> **Note**
>
> CSPCOM will be included in the RSTS/E kit as a maintenance tool only. The use of CSPCOM is not supported for any use other than the compilation of Digital-supplied utility programs that are included with the RSTS/E distribution kit.
>
> The use of CSPCOM is not supported for the compilation of the utility programs supplied with any optional RSTS/E software product unless the installation guide for that product specifically states that the software is compatible with CSPCOM.

Autopatching Package

AUTOPATCH

The autopatching mechanism for the CUSPs has been enhanced to allow:

- Autopatching of the CUSPs when the library is built (via BUILD), rather than the old sequence of OLD-COMPILE-PATCH-COMPILE,
- Checksum verification of source-level patches, and
- Comments in all phases, except within the text of a patch.

EXAMPLE SYSTEM GENERATION

```
RUN $SYSGEN

**08-Mar-79**

Beginning of RSTS/E  system  generation.

Questions come in long and short  forms.
If  you  are  familiar with them, answer
'S' for short; otherwise, answer 'L' for
long form.

Form ?                          $S $
Same system ?                   $Y $
Distribution medium ?           $??$    SY
Output medium ?                 $SY$
Delete files ?                  $NO$
LP for SYSGEN ?                 $Y $    NO
Generate monitor ?              $Y $    T/L
Monitor name ?                  $RSTS$  SVS
Monitor patching ?              $??$    NO
Generate BASIC-PLUS ?           $Y $
BASIC-PLUS RTS name ?           $BASIC$ SVSBAS
BASIC-PLUS patching ?           $??$    NO
RSX as default RTS ?            $NO$

Now you must specify the  hardware  con-
figuration  on  which this RSTS/E system
will run.

KL11,LC11,DL11A,DL11B's ?       $03$
DL11C, DL11D's ?                $00$
DC11's ?                        $00$
DL11E's ?                       $01$
DJ11's ?                        $00$
DH11's ?                        $01$
DH11 unit 00 lines enabled ?    $16$
Dataset support for DH11's ?    $Y $
DZ11's ?                        $00$
Pseudo keyboards ?              $04$
```

Figure SYS-2. System Generation (Page 1 of 7)


```
2741 support ?                  $NO$
Multi-terminal service ?        $Y $
Echo control ?                  $Y $
One-line status report ?        $Y $
RF/RS11's ?                     $NO$
RS03/RS04's ?                   $01$
RK05's ?                        $08$    2
Overlapped seek ?               $Y $    N
RL01's ?                        $00$
RK06/RK07's ?                   $00$
RP02/RP03's ?                   $00$
RM02/RM03's ?                   $00$
RP04/RP05/RP06's ?              $02$
Overlapped seek ?               $Y $
TU16/TE16/TU45's ?              $08$    2
TU10/TE10/TS03's ?              $00$
TS04's ?                        $00$
DECtapes ?                      $00$
TU58's ?                        $00$
Printers ?                      $01$
RX01/RX02's ?                   $00$
CR11/CM11 card reader ?         $NO$
CD11 card reader ?              $NO$
P.T. reader ?                   $NO$
P.T. punch ?                    $NO$
DMC11's ?                       $01$
DECnet network support ?        $Y $    NO
KMC11's ?                       $00$
Extended buffering for LP ?     $Y $
2780 support ?                  $NO$
Large files ?                   $Y $
Maximum jobs ?                  $10$
```

Figure SYS-2. System Generation (Page 2 of 7)

```
Small buffers ?                 @1700
System wide logicals ?          @100
Monitor statistics ?            @NO@    YES
FIP buffering ?                 @Y @
Extended data buffering ?       @NO@    YES
Disk driver phase ?             @Y @
Resident libraries ?            @Y @
RSX directives ?                @Y @
Resident disk handling ?        @Y @
Resident send/receive ?         @NO@
Resident simple SYS calls ?     @NO@
Resident file delete/rename ?   @NO@
Res. login/attach/attribute ?   @NO@
Resident extended lookup ?      @NO@

The following questions deal with the
BASIC-PLUS run-time system

FPP ?                           @Y @
Math precision ?                @2@
Log functions ?                 @Y @
Trig functions ?                @Y @
Print using ?                   @Y @
Matrices ?                      @NO@    YES
String arithmetic ?             @NO@

The system generation dialog is finish-
ed. If you have any special requirements
which require editing the generated file
CONFIG.MAC(system configuration file) or
SYSGEN.CTL(batch control file) you may
do it now.  When ready type "R SYSDAT".

Ready

RUN @SYSDAT


SYSGEN batch processing has started.
If any problems develop during the batch
process it may be aborted by typing
"Control/C". To restart type "R SYSDAT".
```

Figure SYS-2. System Generation (Page 3 of 7)


```
^C
HELLO 1/2
Password:
Jobs 3 4 5 6 7 8 are detached under this account
Job number to attach to?
7 other users are logged in under this account


Ready

RUN $UTILTY.SAV
$SWITCH RT11
@^C

.SIZE 24

.ASSIGN  SY:  IN

.R PIP.SAV
@TDL.OBJ,TTDINT.OBJ,TTDVR.OBJ/DE:NOVARN
@TDL.LST,TTDINT.LST,TTDVR.LST/DE:NOVARN
@RSTS.SAV,TER.SAV,DSK.SAV/DE:NOVARN
@EMT.SAV,FIP.SAV,RSX.SAV,OVR.SAV/DE:NOVARN
@RSTS.MAP,TER.MAP,DSK.MAP/DE:NOVARN
@EMT.MAP,FIP.MAP,RSX.MAP,OVR.MAP/DE:NOVARN
@RSTS.STB,TER.STB,DSK.STB/DE:NOVARN
@EMT.STB,FIP.STB,RSX.STB,OVR.STB/DE:NOVARN
@NSP.SAV,XMDVR.SAV,RJ2780.SAV/DE:NOVARN
@NSP.MAP,XMDVR.MAP,RJ2780.MAP/DE:NOVARN
@NSP.STB,XMDVR.STB,RJ2780.STB/DE:NOVARN
@^C

.R MACRO.SAV
@TTDVR,TTDVR/C=INICOMMON,KERNEL,DKICONFIG,INICHECK,KBDEF,TTDVR
ERRORS DETECTED:  0
@^C

.R MACRO.SAV
@TTDINT,TTDINT/C=INICOMMON,KERNEL,DKICONFIG,INICHECK,KBDEF,TTDINT
ERRORS DETECTED:  0
@^C

.R MACRO.SAV
@TPL,TPL/C=INICOMMON,KERNEL,DKICONFIG,INICHECK,TDL
ERRORS DETECTED:  0
@^C

.R LINK.SAV
@RSTS/Z,RSTS/A/U,RSTS=TPL,@ERR.STB/X/BIG/E:812500/U:81000/I/C
@TTDINT/C
@INIRSTB
Extend section? MORBUF
Round section? MORBUF
Library search? SYSLRO
Library search? BUFEXT
Library search? LPDVRX
Library search? LIGHTB
Library search?
@^C
```

Figure SYS-2. System Generation (Page 4 of 7)

```
.R LINK.SAV
*DSA,Z:DSA/A/U:DSR=INIDBR:DKIRSTS.STB/X/B10117000/U101000/I/C
*IN:RSTS
Round section? DSKPAT
Library search? DDSEEK
Library search?
*^C

.R LINK.SAV
*TER/Z:TER/A/U:TER=INITER:DKIRSTS.STB/X/B10117000/U101000/I/C
*TTDVR/C
*IN:RSTS
Round section? TERPAT
Library search? TTSYST
Library search?
*^C

.R LINK.SAV
*XNDVR/Z:XNDVR/A/U:XNDVR=IN:XNP:DKIRSTS.STB/X/B10117000/U101000/I/C
*IN:RSTS
Round section? XNDPAT
Library search? XNDVR
Library search?
*^C

.R LINK.SAV
*ENT/Z:ENT/A/U:ENT=INIENT:DKIRSTS.STB/X/B10117000/U101000/I/C
*IN:RSTS
Round section? ENTPAT
Library search? LIB
Library search?
*^C

.R LINK.SAV
*FIP/Z:FIP/A/U:FIP=INIFIPLAB:DKIRSTS.STB/X/B10117000/U101000/I/C
*IN:RSTS
Round section? FIPPAT
Library search? DPL
Library search?
*^C

.R LINK.SAV
*RSX/Z:RSX/A/U:RSX=INIRSX:DKIENT.STB/X/B10117000/U101000/C
*INIRSTS
Round section? RSXPAT
*^C

.R LINK.SAV
*OVR/Z:OVR/A/U:OVR=INIOVRLAB:DKIFIP,STB/X/B101000/C
*INIRSTS
*^C

.R SILUS.SAV
*SY0:[0,1]SYS.SIL=TTI=RSTS,TER/M,DSK/M,ENT/M,FIP/M/C
*RSX/M/C
*XNDVR/M/C
*OVR/M/C
*IN:DBI,DEFALT
Directory of SIL SYS   .SIL on 08-Mar-79 at 14:47

 Name   Ident    Load    Size  Transfer Total

RSTS    07.006  000000  072000  000001   15K
TER     07.005  120000  017000  000001   17K
DSK     07.005  120000  005000  000001   20K
```

Figure SYS-2. System Generation (Page 5 of 7)


```
[illegible]  07.005  120000  013000  000001   24K
[illegible]  07.005  120000  011000  000001   18K
RSX     07.005  120000  007000  000001   30K
XNP     07.005  120000  001000  000001   30K
DVR     07.005  002000  044003  000001   43K
DBI     07.005  140000  012520  142522   46K
DEFALT  07.005  001000  002000  000001   46K

Resident monitor size is 30K (with 180 small buffers)
535 small buffers can be added to this system (716 total)
16 small buffers can be added without a size change (196 total)



*^C

.R PIP.SAV
*BASIC.SAV=BASIC.STB/DE:NOVARM
*^C

.R LINK.SAV
*BASIC/Z:SUSBAS/A/U:BASIC=INIRTS,DKI:ERR.BIB/X/HI0177776/U104000/C
*IN:MA2F/C
*IN:XL2F/C
*IN:XT2F/C
*IN:IO/C
*IN:FU/C
*IN:MX/C
*IN:SM/C
*IN:VE
Round section? PA
*^C

.R SILUS.SAV
*SUSBAS.RTB=TTI=BASIC
Directory of SIL SUSBAS.RTB on 08-Mar-79 at 14:47

 Name   Ident    Load    Size  Transfer Total

BASIC   07.005  104000  073776  000001   15K



*^C

.DEASSIGN  IN

.R PIP.SAV
*SY0:[0,1]0.0/MODE:16,=SUSBAS.RTB
*SUSBAS.R1B/DE
*^C

.R LOGOUT
Confirm Y
Saved all disk files( 12232 blocks in use
Job 11 User 1,2 logged off KB4 at 08-Mar-79 14:47
2 other users still logged in under this account
System RSTS V7.0-06 Mass FT (17Feb)
Run time was 5 minutes 37.6 seconds
Elapsed time was 10 minutes
Good afternoon
```

Figure SYS-2. System Generation (Page 6 of 7)

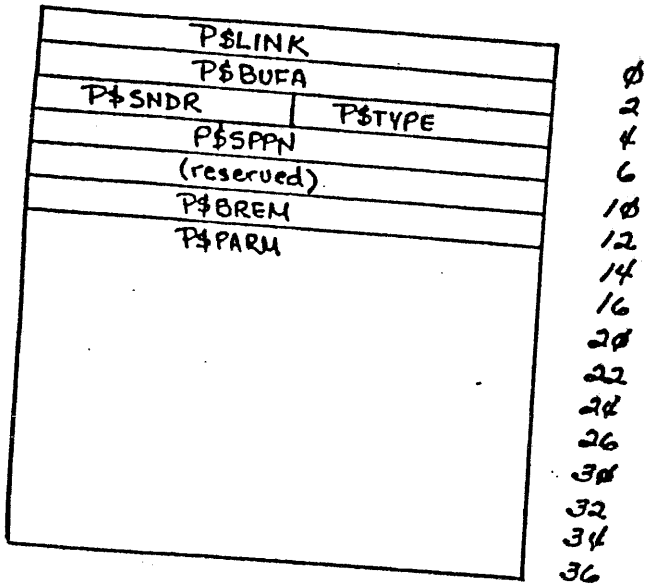


Figure MON-41. Pending Message Block (PMD)

P$LINK	Link to next block
P$BUFA	"Contorted" buffer address (pointer to buffer containing message data)
P$TYPE	Message type code (DECNET/E)
P$SNDR	Job number times two of sender
P$SPPN	Project Programmer number of sender
P$BREM	Number of bytes remaining in data message
P$PARM	(20 bytes) Parameters

"Contorted" Buffer Address

- If low five bits are zero then a small buffer
- If in XBUF, address has been rotated left seven bits so that low bits are not zero

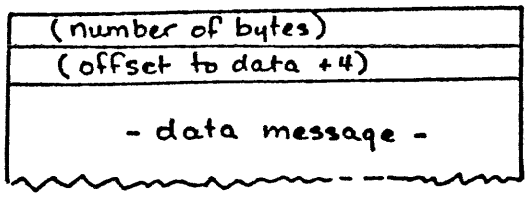


Figure MON-42. Buffer Containing Message (Long Messages Only)

(number of bytes) Length of data message

(offset to data + 4) Pointer to next character to be "received"

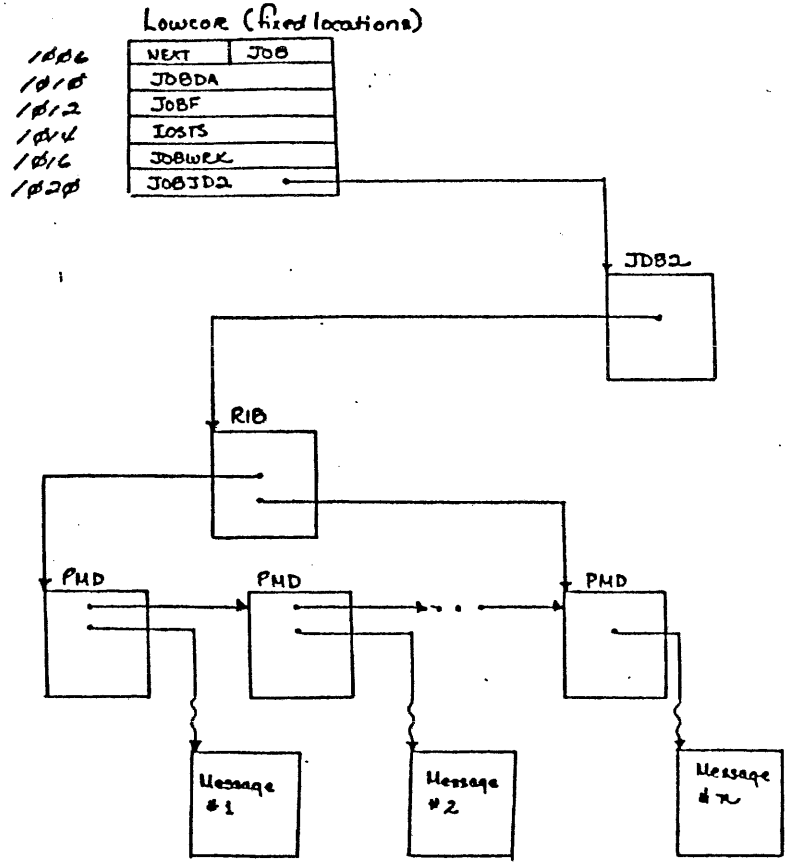


Figure MON-43. Monitor Tracking of Message Send/Receive for "JOB"

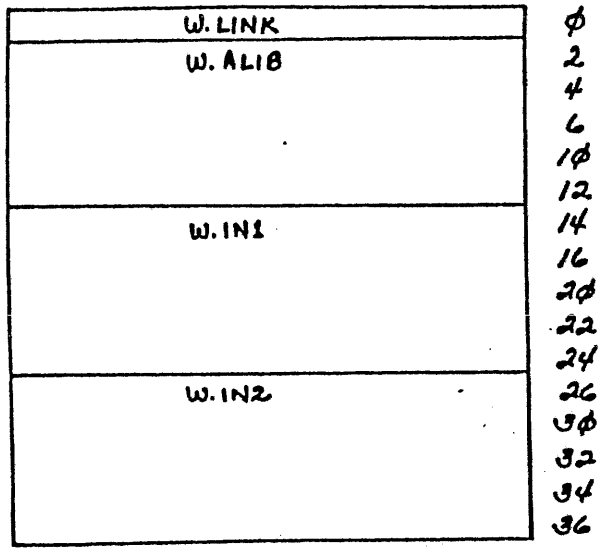


Figure MON-44. Window Descriptor Block (WDB)

W.LINK	Link to next WDB @ offset + 2
W.ALIB	Addresses of LIB blocks for attached libraries
W.IN1	Address window #1
W.IN2	Address window #2

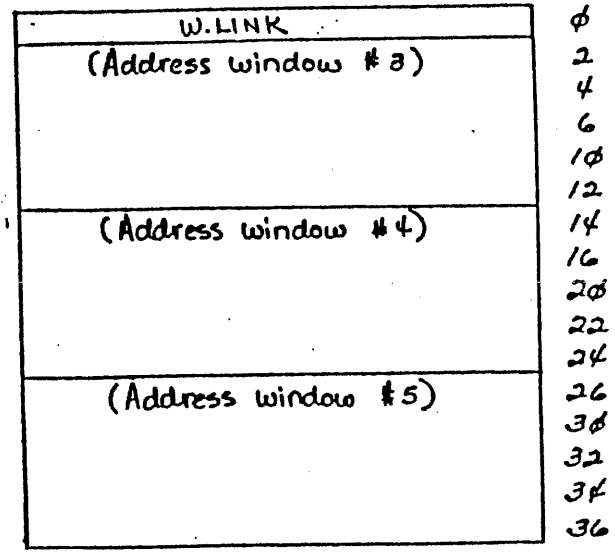


Figure MON-45. Window Descriptor Block #2 (WDB)

- Exists only if more than two APRs are mapping a resident library (or libraries)

```
Batch job completed.

Ready
```

Figure SYS-2. System Generation (Page 7 of 7)

RSTS/E UPDATE

BKP

Backup Programs

Backup Programs

CONTENTS

Introduction..... BKP-1

PIP Program..... BKP-2
General..... BKP-2
New Features..... BKP-2
New Switches and Options..... BKP-3
Multi-Volume Example..... BKP-7
Notes on PIP..... BKP-8

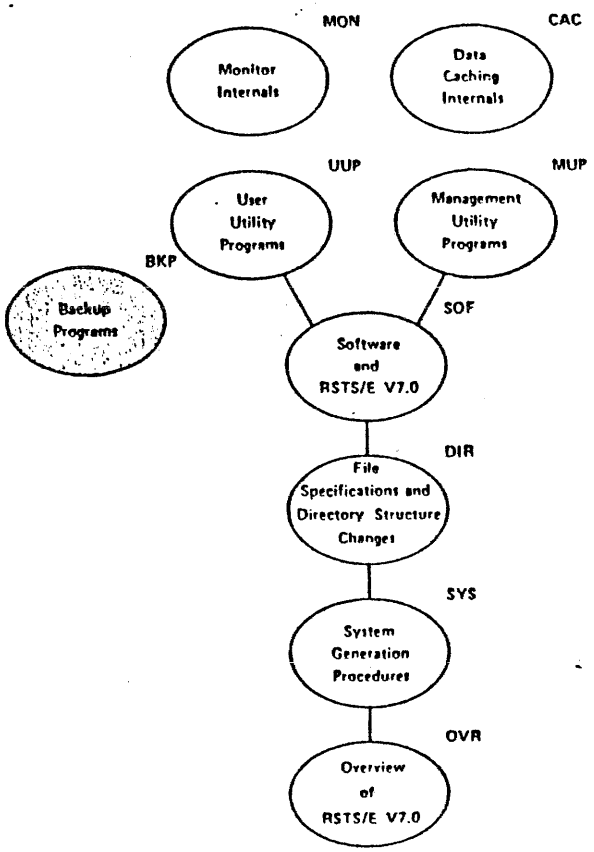
BACKUP Package..... BKP-12
New Features..... BKP-12
Example BACKUP in Interactive Mode..... BKP-13
Example of LOADINDEX Mode..... BKP-17
Example of LIST Mode..... BKP-19
Example of BACKUP Batch Control File..... BKP-22
Example of BACKUP Batch Log..... BKP-23

SAVE/RESTORE Program..... BKP-28
Features..... BKP-29
Rules for Using Media..... BKP-29
SAVE/RESTORE Options..... BKP-31
Example of IDENTIFY and SAVE Functions..... BKP-32
Example of RESTORE Function..... BKP-34
Example of IMAGE Function..... BKP-36
Additional SAVE/RESTORE Documentation..... BKP-38
Definitions of SAVE/RESTORE Terms..... BKP-39
Running SAVE/RESTORE..... BKP-40
SAVE/RESTORE Functions..... BKP-41
SAVE/RESTORE Options..... BKP-43
Dialogue..... BKP-45
SAVE/RESTORE Volumes..... BKP-46
Device Specifications..... BKP-47
Checking the Input Volume..... BKP-48
Checking the Output Volume..... BKP-49

General Comparison of Backup Programs..... BKP-53

Transferring RMS Relative and Indexed Files to
Other non-RSTS/E Operating Systems..... BKP-55

COURSE MAP


MON
Monitor Internals
CAC
Data Caching Internals
UUP
User Utility Programs
MUP
Management Utility Programs
BKP
Backup Programs
SOF
Software and RSTS/E V7.0
DIR
File Specifications and Directory Structure Changes
SYS
System Generation Procedures
OVR
Overview of RSTS/E V7.0

Backup Programs

INTRODUCTION

Since backup is a major concern for all system managers, RSTS/E supplies several backup utility programs which handle the individual needs of each RSTS/E site.

The new or enhanced RSTS/E V7.0 utilities:

- Backup and restore
  - large files
  - placed files
  - contiguous files
  - attributed files
  - entire disks
- Produce bootable backup media with bad block handling

Decisions must be made concerning which utility to use under a given set of circumstances.

In this module you will learn:

- The features of the new utility SAVE/RESTORE,
- The enhancements to PIP and the BACKUP package, and
- When to use SAVE/RESTORE, PIP, BACKUP and RMSBCK

PIP PROGRAM

General

- Has expanded from 8Kw to 12Kw,
- Has replaced PIPSML and PIPEXT,
- Accepts the /POSITION:n switch to create a placed file as long as the switch is adjacent to the filename,
- Is the only way to selectively back up large files which cannot fit on a single tape, and
- Zeroes large files faster than mode 128.

New features

- Multi-volume ANSI magnetic tape support, and
- Specification of density and parity when zeroing a magnetic tape.

New Switches and Options

File Access Switches

Switch	Format	Meaning
/ACCESS	/AC	For disk input only, this switch causes PIP to change the file's last access date to the current date. If you do not specify this switch, PIP preserves the last access date.
/ASCII	/AS	Does a formatted ASCII transfer, causing PIP to ignore nulls, parity bits, and rubout characters.
/BSIZE:n		Used to specify a non-standard blocksize for file transfers to magtape. "n" is the blocksize in bytes to be output. "n" must be a positive even integer from 18 to 4096. For ANSI format magtapes intended for interchange with foreign operating systems, the blocksize should not exceed 2048. For interchange with RT11 operating systems, the default of 512 should be used.
/NEwfile	/NE	For disk output only, this switch creates a new file with the current date of creation and access.
/NOAttributes	/NOA	This switch causes PIP to transfer the file without writing attributes to the output file.
/PR:NOWARN		This switch causes PIP to ignore the operation if an output file already exists and to suppress the printing of the error message.

Switch	Format	Meaning
/RETAIN		For disk output, this switch preserves the file's creation, access dates, and creation time. This is the normal default for PIP, but it may be changed by an optional patch.
/RTS:name		This switch sets the output file's run-time system name to that specified in name. If no output file is specified, PIP renames the input file's run-time system.
/CREATION	/CRE	This switch uses the creation date for limit checking with the date switches (/AF, /BE, /ON, /SI, /TO). This is the normal default for PIP, but it may be changed by an optional patch.
/DLA		This switch uses the date of last access for limit checking with the date switches.

Attributed File Translation Switch

/RMS /ANSI	/RMS:FTN /ANSI:FTN	For input files, translates FORTRAN carriage control to formatted ASCII. For non-disk output files, translation is automatic.
	/ANSI:PRN /RMS:PRN	For input disk files, translates RMS print files to formatted ASCII.

File Operation Switches - General

/IN	/IN	Same as V6C.
	/IN:S	Prints the filename, extension, filesize, protection code, creation date, and last access date.

File Operation Switches Deletion

Switch	Format	Meaning
/DE	/DE	Same as V6C.
	/DE:NO[WARN]	Causes PIP to suppress the printing of an error message if the file to be deleted does not exist.

File Operation Switches - Listing

/DI:option /LI:option	:AL	Prints the filename, extension, and the number of blocks allocated to the file.
	:AT :SA	Prints the filename, extension and symbolic representation of file attributes.
	:FU	Full listing; print all information except file attributes.
	:OA	Prints the filename, extension, and octal representation of file attributes.
	:RT	Prints the filename, extension, and associated run-time system.
	:SZ	Prints the filename, extension and file size (same as :SI).
	:WI	Prints filenames only across the page or screen.
/SLOW	:SL	Same as /DI:S .

File Operation Switches - Zeroing

Switch	Format	Meaning
/DENsity	/ZE/DEN:n	For magtape only, this switch sets the tape density prior to the zero operation.
/PARity	/ZE/PAR:ODD /ZE/PAR:EVEN	For magtape only, this switch sets the tape parity prior to the zero operation.

Multi-Volume Example

Consider the following example which transfers the contents of a user account to three magnetic tape volumes:

```
RUN $PIP
*MM0:*.*=[20,254]*.*/BL

%END OF ANSI MAGTAPE OUTPUT VOLUME HAS
%BEEN REACHED

%PLEASE TYPE THE DEVICE NAME AND UNIT
%NUMBER OF THE DRIVE WHERE THE NEXT
%VOLUME MAY BE FOUND

                                    ?MM1:TAPE2

%END OF ANSI MAGTAPE OUTPUT VOLUME HAS
%BEEN REACHED

%PLEASE TYPE THE DEVICE NAME AND UNIT
%NUMBER OF THE DRIVE WHERE THE NEXT
%VOLUME MAY BE FOUND

                                    ?MM2:TAPE3
*
```

In this example, you specify the block mode transfer of all files in account [20,254] on the system disk to the tape mounted on MM0:. When PIP reaches the end of the output tape, it prints the dialogue that requests the next volume. In response to the prompt, type the device name and unit number (MM1:) and the optional volume ID (TAPE2). PIP checks the output volume ID, if specified, and that the tape is ANSI labeled and initialized. If the ID is valid, and the tape is ANSI and free of data, then PIP continues with the transfer. This process continues until all of the specified files on the input device are transferred.

When the transfer is complete, PIP prints its asterisk prompt.

Notes on PIP

1. Appending files with attributes

   o Do not merge attributed files with PIP.

2. RMS Relative and Indexed Files

   a. Use the /BL switch for transfer from disk to ANSI magtape.

   b. Use no switch for transfer from ANSI magtape to disk.

> **Note**
>
> The above procedure is for RSTS/E to RSTS/E transfers only.

   c. No switches are needed for disk-to-disk transfers.

3. RMS Sequential Files

   a. No switches are needed for file transfers to and from ANSI magtape.

   b. No switches are needed for disk-to-disk transfers.

4. /ANSI and /RMS switches

   a. /ANSI:FA or /RMS:FA on the input file specification:

      Purpose - Translate RMS Sequential files to stream ASCII files

      o Takes an RMS sequential file with fixed or variable length records as input.

      o Strips off the byte count (if necessary), squeezes out null and rubout characters, and appends <CR><LF> to each record.

      o Creates a stream ASCII output file without attributes.

> **Note**
>
> Do not use this switch if the RMS sequential file contains binary data.

   b. /ANSI:FA or /RMS:FA on the output file specification:

      Purpose - Translate stream ASCII files to RMS sequential files with variable length records.

      o Takes a stream ASCII file as input.

      o Places a byte count at the beginning of each record and strips off <CR><LF>.

      o Creates a sequential file with variable length records.

   c. /ANSI:FTN or /RMS:FTN on the input file specification:

      Purpose - Translate files with FORTRAN carriage control characters to stream ASCII files.

      o Takes a file with FORTRAN carriage control attributes as input.

      o Strips off the FORTRAN carriage control characters from each record and appends <CR><LF>.

      o Creates a stream ASCII file without attributes.

> **Note**
>
> Specifying this switch on an output file specification produces a ?Bad Switch Value error.

d. /ANSI:FB or /RMS:FB on the input file specification:

Purpose - Transfer RSX type object modules to DOS labelled magnetic tape.

- o Takes RSX type object modules as input.
- o Translates the file into RT11 object module format.

e. /ANSI:FB or /RMS:FB on the output file specification:

Purpose - Transfer RSX type object modules from DOS magnetic tape to disk.

- o Translates the file into RSX object module format.

f. /ANSI:PRN or /RMS:PRN on the input file specification:

Purpose - Translate RMS print format files to stream ASCII files.

> **Note**
>
> Currently VAX uses this format for its log files. If these files are transferred to RSTS/E by magnetic tape or through DECnet/E, the /ANSI:PRN should be used since the SPOOL program does not recognize this format.

- o Strips carriage control information from records and inserts appropriate control characters.

> **Note**
>
> Specifying this switch on the output file specification produces a ?Bad Switch Value error.

g. /ANSI:IM or /RMS:IM on the input file specification:

Purpose - Print map files created by the task builder at terminals.

> **Note**
>
> /ANSI:FA on the input file also performs this operation.

- o Takes an RMS sequential file as input.
- o Strips off the byte count at the beginning of each record (if necessary)

> **Note**
>
> Specifying this switch on the output file specification produces a ?Bad Switch Value error.

BACKUP PACKAGE

New Features

- Copying of the index from the BACKUP set to a disk,
- Listing a directory index,
- Ability to run under BATCH,
- Caching of the workfile,
- Sequential caching of the files being backed up, and
- Residing in an account other than [1,2].


```
PUB [1,10]BACKUP
BACKUP  V7.0-05 RSTS V7.0-05 The ARK

BAC[KUP], RES[TORE], LOA[DINDEX] OR LIS[T] ? BACKUP
WORK FILE NAME<_ST:[  1,235]B20FEB.J25> ?
LISTING FILE<_KB:> ?
FROM DISK<_ST:> ?
FROM FILES<[  1,235]*.*> ? [1,235]*.*/EXC:(*.BAK,*.J25,BCKBAC.LST)
TO DEVICE<_MT:> ?
BEGIN AT<[*,*]*.*> ?
DELETE FILES<NONE> ?
COMPARE FILES<NONE> ?
*


                    Backup Run Listing

Backup from '_ST    :' to '_MT    :
Run started on 20-Feb-79 at 09:14 AM
Work-File is _ST    :[1,235]B20FEB.J25
Transfer  : [1,235]??????.???/EXC:([1,235]??????.BAK,[1,235]??????.J25,[1,235]BCKBAC.LST)
Begin at  : [*,*]*.*
Delete    : <none>
Compare   : <none>
*
PHASE   : LIST COMPLETE
ERRORS  : 0

ELAPSED TIME : 1 SECONDS
CPU TIME     : .4 SECONDS
KCTS         : 68


*
PHASE   : LOOKUP/SELECT COMPLETE
```

Figure BKP-1. BACKUP in Interactive Mode (Page 1 of 4)

```
VOLUME # : 1
ACCOUNTS : 1
FILES    : 44
BLOCKS   : 378
ERRORS   : 0


ELAPSED TIME : 23 SECONDS
CPU TIME     : 4.7 SECONDS
RCTS         : 705

.
PLEASE ENTER BACKUP SET NAME<B30FEB> -
PLEASE ENTER EXPIRATION DATE<28-Feb-80> -
PLEASE ENTER DENSITY IN BPI<800> -
PLEASE ENTER THE PARITY<ODD> -
MOUNT    DEVICE:          MT    :
            ID:          $30FEB
          SEQ#:          1
       DENSITY:          800 BPI
        PARITY:          ODD
       IDENTIFICATION WILL BE FINAL UPON SUCCESSFUL MOUNT
DEVICE? MM0:
%THIS VOLUME HAS NO BACKUP LABEL%
MOUNT IT ANYWAY<NO> ? Y

.

PHASE   : MOUNT / DISMOUNT COMPLETE
ERRORS  : 0


ELAPSED TIME : 51 SECONDS
CPU TIME     : .8 SECONDS
RCTS         : 128

.

PHASE   : TRANSFER COMPLETE
VOLUME # : 1
ACCOUNTS : 1
FILES    : 44
BLOCKS   : 378
ERRORS   : 0

CURRENT VOLUME  : _MM0    :

ELAPSED TIME : 23 SECONDS
CPU TIME     : 4 SECONDS
RCTS         : 836

.

PHASE   : INDEX DUMP COMPLETE
ERRORS  : 0

CURRENT VOLUME  : _MM0    :

ELAPSED TIME : 6 SECONDS
CPU TIME     : .6 SECONDS
RCTS         : 94
```

Figure BKP-1. BACKUP in Interactive Mode (Page 2 of 4)


```
.
DISMOUNT DEVICE:          MM0    :
            ID:          $30FEB
          SEQ#:          1(INDEX)
       DENSITY:          800 BPI
        PARITY:          ODD
EXPIRATION DATE:         28-Feb-80
       PLEASE LABEL THIS VOLUME!

.

PHASE   : MOUNT / DISMOUNT COMPLETE
ERRORS  : 0


ELAPSED TIME : 3 SECONDS
CPU TIME     : .3 SECONDS
RCTS         : 48

.



Backup Set Name : B30FEB              Backup Device : _MM    :
Volume Sequence # : 1(Index)          Owner : [1,235]
Creation Date : 28-Feb-79             Expiration Date : 28-Feb-80
Density : 800 BPI                     Parity : ODD


  Account : [1,235]                   Quota : 5000
  Clustersize : 16
                                  Creation          Access
  Name  Ext  Size  Prot    Date      Time      Date      Clu  PTS   TSCD

                        (SOURCE DISK = DB1:)
NFD   .DMP     39    60  09-Nov-78 02:59 PM  09-Nov-78     0 BAS4F   T
DMPFIL.BAS      3    60  09-Nov-78 03:32 PM  09-Nov-78     0 ...BSR*T
NFD1  .DMP     52    60  09-Nov-78 03:30 PM  09-Nov-78     0 BASIC2  T
SYS   .LST      3    60  16-Nov-78 01:16 PM  16-Nov-78     0 ...BSR*T
SYSO  .LST      3    60  16-Nov-78 01:23 PM  16-Nov-78     0 ...BSR*T
SEQ   .CBL      3    60  29-Nov-78 11:03 AM  29-Nov-78     0 ...BSC*T
SEQ2  .ODL      2    60  29-Nov-78 12:15 PM  29-Nov-78     0 ...BSC*T
CONT  .DAT    10C    60  07-Dec-78 08:57 AM  07-Dec-78     0 BAS4F   T
PLACE .BAS      1    60  07-Dec-78 09:47 AM  07-Dec-78     0 ...BSR*T
IIO34 .B2S      1    60  29-Nov-78 09:14 AM  29-Nov-78     0 B2YCOM  T
IIO24 .ODL      1    60  15-Dec-78 02:11 PM  15-Dec-78     0 B2COM   T
ONLY  .MAS     11    60  15-Dec-78 02:19 PM  15-Dec-78     0 BASIC2*T
DMP   .BAS      2    60  01-Jun-78 11:47 AM  01-Jun-78     0 BAS4F   T
LOOP  .BAS      1    60  12-Jul-78 10:41 AM  12-Jul-78     0 BAS4F   T
ERTIC .BAS      2    60  05-Sep-78 04:50 PM  05-Sep-78     0 BAS4F   T
TCOMP .BAS      1    60  24-Oct-78 03:31 PM  24-Oct-78     0 BAS4F   T
CCONT .BAS      1    60  21-Nov-78 02:36 PM  21-Nov-78     0 BAS4F   T
MOMII .BAS      1    60  21-Nov-78 02:36 PM  21-Nov-78     0 BAS4F   T
CCPOS .BAS      1    60  21-Nov-78 02:38 PM  21-Nov-78     0 BAS4F   T
CHAIN .BAS      1    60  21-Nov-78 02:39 PM  21-Nov-78     0 BAS4F   T
DEF   .BAS      1    60  21-Nov-78 02:39 PM  21-Nov-78     0 BAS4F   T
MOD128.BAS      1    60  27-Nov-78 02:22 PM  27-Nov-78     0 BAS4F   T
VIRT1 .BAS      1    60  27-Nov-78 02:30 PM  27-Nov-78     0 ...BSR*T
SEQ   .BAS      1    60  27-Nov-78 03:10 PM  27-Nov-78     0 BAS4F   T
VIRTUL.BAS      1    60  28-Nov-78 10:40 AM  28-Nov-78     0 BAS4F   T
UPDCRT.BAS      1    60  28-Nov-78 12:09 PM  28-Nov-78     0 BAS4F   T
UPDACC.BAS      1    60  28-Nov-78 12:11 PM  28-Nov-78     0 BAS4F   T
```

Figure BKP-1. BACKUP in Interactive Mode (Page 3 of 4)

```
UPDERT.BAS     1   60  28-Nov-78 12:18 PM  28-Nov-78    0 BAS4F  T
UPDRED.BAS     1   60  28-Nov-78 02:20 PM  28-Nov-78    0 BAS4F  T
TEMT1 .BAS     1   60  28-Nov-78 02:34 PM  28-Nov-78    0 ...RSX*T
TEMT  .BAS     1   60  28-Nov-78 02:47 PM  28-Nov-78    0 ...RSX*T
TEMT2 .BAS     1   60  05-Dec-78 04:11 PM  05-Dec-78    0 ...RSX*T
FMSC  .BAS     1   60  06-Dec-78 10:20 AM  06-Dec-78    0 ...RSX*T
MCMCMT.BAS     1   60  07-Dec-78 09:17 AM  07-Dec-78    0 BAS4F  T
TTT44 .BAS     1   60  15-Dec-78 03:27 PM  15-Dec-78    0 BAS4F  T
EVLMFS.BAS     1   60  12-Dec-78 05:10 PM  12-Dec-78    0 BAS4F  T
EVLMFS.BAC    15C 124  22-Dec-78 12:55 PM  22-Dec-78    0 BAS4F  T
EVLMFS.CRF    10   60  22-Dec-78 01:12 PM  22-Dec-78    0 BASIC2 T
AWALTE.DMP    10   60  02-Jan-79 09:52 AM  02-Jan-79    0 BASIC2 T
BSTB  .DMP     6   60  15-Jan-79 03:43 PM  15-Jan-79    0 RT11   T
BSTARL.IDX     4   60  17-Jan-79 09:58 AM  17-Jan-79    0 RT11   T
BSTAPL.DMP     4   60  17-Jan-79 10:09 AM  17-Jan-79    0 RT11   T
SYSGEN.LOG    40   60  01-Feb-79 03:26 PM  01-Feb-79    0 ...RSX*T
SYSGEN.CTL     7   60  06-Feb-79 10:32 AM  06-Feb-79    0 BAS4F  T

  * Attributes associated with this file

  Account total of 270 blocks in 44 files on account [1,235]

Volume total of 270 blocks in 44 files in 1 accounts on volume _MM    :


*****************************************************************

RUN total of 270 blocks in 44 files on 1 accounts on 1 Volumes
RUN total of 0 errors
RUN terminated on 20-Feb-79 at 09:16 AM
*****************************************************************

*

PHASE   : LIST COMPLETE
VOLUME # : 1
ACCOUNTS : 1
FILES    : 44
BLOCKS   : 270
ERRORS   : 0

ELAPSED TIME : 141 SECONDS
CPU TIME     : 2.5 SECONDS
KCTS         : 435


*
Ready
```

Figure BKP-1. BACKUP in Interactive Mode (Page 4 of 4)


```
RUN[1,10]BACKUP
BACKUP  X7.0-05 RSTS V7.0-05 The ARK

BAC[KUP], RES[TORE], LOA[DINDEX] OR LIS[T] ? LOADINDEX

WORK FILE NAME<_BY:(  1,235)B20FEB.J25> ?

LISTING FILE<_KB:> ?

FROM DEVICE<_MT:> ?

TO FILE<_BY:(  1,235)BIND20.IND> ?
*


                    Backup Run Listing

Loadindex from '_MT    :

Run started on 20-Feb-79 at 09:21 AM

Work-File is _BY    :[1,235]B20FEB.J25

To file  : _BY    :[1,235]BIND20.IND

*

PHASE   : LIST COMPLETE
ERRORS   : 0

ELAPSED TIME : 0 SECONDS
CPU TIME     : .3 SECONDS
KCTS         : 44


*
PLEASE ENTER BACKUP SET NAME<B20FEB> -
PLEASE ENTER DENSITY IN BPI<800> -
PLEASE ENTER THE PARITY<ODD> -
MOUNT    DEVICE:          MT    :
            ID:          B20FEB
           SEQ:          INDEX
       DENSITY:          800 BPI
        PARITY:          ODD
       PLEASE MOUNT VOLUME WRITE LOCKED!
DEVICE? MM0:
*

PHASE   : MOUNT / DISMOUNT COMPLETE
ERRORS   : 0
```

Figure BKP-2. LOADINDEX Mode (Page 1 of 2)

```
ELAPSED TIME : 48 SECONDS
CPU TIME     : .8 SECONDS
RCTS         : 144

.

PHASE   : INDEX LOAD COMPLETE
ERRORS  : 0

CURRENT VOLUME  : _MM0   :

ELAPSED TIME : 1 SECONDS
CPU TIME     : .5 SECONDS
RCTS         : 76

.
DISMOUNT DEVICE:        _MM0   :
.

PHASE   : MOUNT / DISMOUNT COMPLETE
ERRORS  : 0

ELAPSED TIME : 3 SECONDS
CPU TIME     : .1 SECONDS
RCTS         : 16

.
Ready
```

Figure BKP-2. LOADINDEX Mode (Page 2 of 2)


```
RUN [1,10]BACKUP
BACKUP  X7.0-05 RSTS V7.0-05 The ARR

BAC[KUP], RES[TORE], LOA[D]INDEX] OR LIS[T] ? LIST

WORK FILE NAMEC_BY:[  1,235]B20FEB.325> ?

LISTING FILEC_KB:> ?

INDEX FILE<PRIMARY> ? BIND20.IND
.



                        Backup Run Listing

List

Run started on 20-Feb-79 at 09:24 AM

Work-File is _BY    :[1,235]B20FEB.325

.

PHASE   : LIST COMPLETE
ERRORS  : 0

ELAPSED TIME : 1 SECONDS
CPU TIME     : .2 SECONDS
RCTS         : 30

.

PHASE   : INDEX LOAD COMPLETE
ERRORS  : 0

ELAPSED TIME : 3 SECONDS
CPU TIME     : .7 SECONDS
RCTS         : 97

.


Backup Set Name : B20FEB          Backup Device : _MM0   :
Volume Sequence # : 1(Index)      Owner : [1,235]
Creation Date : 20-Feb-79         Expiration Date : 20-Feb-80
Density :  800 BPI                Parity : ODD
```

Figure BKP-3. LIST Mode (Page 1 of 3)

```
  Account : [1,235]                Quota : 5000
  Clustersize : 10

                                Creation          Access
 Name   Ext   Size   Prot      Date     Time       Date     Clu  PTS   TSCO
WFD    .DMP    39    60  09-Nov-78 02:59 PM  09-Nov-78    0 BAS4F  T
DMPFIL.BAS     3    60  09-Nov-78 03:32 PM  09-Nov-78    0 ...PSX*T
WFDI   .DMP    52    60  09-Nov-78 03:38 PM  09-Nov-78    0 BASIC2 T
STS    .LST     3    60  16-Nov-78 01:16 PM  16-Nov-78    0 ...PSX*T
STSO   .LST     3    60  16-Nov-78 01:23 PM  16-Nov-78    0 ...PSX*T
SEQ    .CBL     3    60  29-Nov-78 11:03 AM  29-Nov-78    0 ...PSX*T
SEQ2   .ODL     2    60  29-Nov-78 12:13 PM  29-Nov-78    0 ...PSX*T
COMT   .DAT    16C   60  07-Dec-78 08:57 AM  07-Dec-78    0 BAS4F  T
PLACE  .BAS     1    60  07-Dec-78 09:47 AM  07-Dec-78    0 ...PSX*T
IIO2A  .BIN     1    60  29-Nov-78 09:14 AM  29-Nov-78    0 BP1COM T
IIO2A  .ODL     1    60  15-Dec-78 02:33 PM  15-Dec-78    0 BP2COM T
ONLY   .PAS    11    60  15-Dec-78 02:19 PM  15-Dec-78    0 BASIC2*T
DMP    .BAS     2    60  01-Jun-78 11:47 AM  01-Jun-78    0 BAS4F  T
LOOP   .BAS     1    60  12-Jul-78 10:41 AM  12-Jul-78    0 BAS4F  T
SPTIC  .BAS     2    60  05-Sep-78 04:59 PM  05-Sep-78    0 BAS4F  T
TCOMP  .BAS     1    60  24-Oct-78 03:31 PM  24-Oct-78    0 BAS4F  T
CCOMT  .BAS     1    60  21-Nov-78 02:36 PM  21-Nov-78    0 BAS4F  T
MCMII  .BAS     1    60  21-Nov-78 02:36 PM  21-Nov-78    0 BAS4F  T
CCPOS  .BAS     1    60  21-Nov-78 02:38 PM  21-Nov-78    0 BAS4F  T
CHAIN  .BAS     1    60  21-Nov-78 02:39 PM  21-Nov-78    0 BAS4F  T
DEF    .BAS     1    60  21-Nov-78 02:39 PM  21-Nov-78    0 BAS4F  T
MSD128.BAS     3    60  27-Nov-78 02:22 PM  27-Nov-78    0 BAS4F  T
VIRT1  .BAS     1    60  27-Nov-78 02:38 PM  27-Nov-78    0 ...PSX*T
SEQ    .BAS     1    60  27-Nov-78 03:10 PM  27-Nov-78    0 BAS4F  T
VIRTUL.BAS     1    60  28-Nov-78 10:48 AM  28-Nov-78    0 BAS4F  T
UPDCRT.BAS     1    60  28-Nov-78 12:09 PM  28-Nov-78    0 BAS4F  T
UPCACC.BAS     1    60  28-Nov-78 12:11 PM  28-Nov-78    0 BAS4F  T
UPCEXT.BAS     1    60  28-Nov-78 12:16 PM  28-Nov-78    0 BAS4F  T
UPCPTO.BAS     1    60  28-Nov-78 02:20 PM  28-Nov-78    0 BAS4F  T
TEMT1  .BAS     1    60  28-Nov-78 02:24 PM  28-Nov-78    0 ...PSX*T
TEMT   .BAS     1    60  28-Nov-78 02:47 PM  28-Nov-78    0 ...PSX*T
TEMT2  .BAS     1    60  05-Dec-78 04:11 PM  05-Dec-78    0 ...PSX*T
FMSC   .BAS     1    60  06-Dec-78 10:20 AM  06-Dec-78    0 ...RSX*T
MOMCMT.BAS     1    60  07-Dec-78 09:17 AM  07-Dec-78    0 BAS4F  T
TIT44  .BAS     1    60  15-Dec-78 03:27 PM  15-Dec-78    0 BAS4F  T
EVLMFS.BAS    14    40  12-Dec-78 05:10 PM  12-Dec-78    0 BAS4F  T
EVLMFS.BAC   15C   124  22-Dec-78 12:55 PM  22-Dec-78    0 BAS4F  T
EVLMFS.CMF    10    60  22-Dec-78 01:12 PM  22-Dec-78    0 MASIC2 T
ANALYS.DMP    10    60  02-Jan-79 09:52 AM  02-Jan-79    0 BASIC2 T
RSTS   .DMP     6    60  15-Jan-79 03:43 PM  15-Jan-79    0 RT11   T
RSTARE.IDX      4    60  17-Jan-79 09:58 AM  17-Jan-79    0 RT11   T
RSTARE.DMP      4    60  17-Jan-79 10:09 AM  17-Jan-79    0 RT11   T
BTSCEN.LOG     40    60  01-Feb-79 03:26 PM  01-Feb-79    0 ...PBX*T
BTSCEN.CTL      7    60  06-Feb-79 10:32 AM  06-Feb-79    0 BAS4F  T

   * Attributes associated with this file

  Account total of 270 blocks in 44 files on account [1,235]

Volume total of 270 blocks in 44 files in 1 accounts on volume _MM0   :
```

Figure BKP-3. LIST Mode (Page 2 of 3)


```
RUN total of 270 blocks in 44 files on 1 accounts on 1 Volumes
RUN total of 0 errors
RUN terminated on 20-Feb-79 at 09:25 AM

.

PHASE   : LIST COMPLETE
VOLUME 0 : 1
ACCOUNTS : 1
FILES    : 44
BLOCKS   : 270
ERRORS   : 0

ELAPSED TIME : 58 SECONDS
CPU TIME     : 3 SECONDS
KCTS         : 450


.
Ready
```

Figure BKP-3. LIST Mode (Page 3 of 3)

```
$JOB
$RUN [1,10]BACKUP
$DATA
BACKUP


[1,235]*.*/EXC:(*.BAK,*.J25)




MM0:
YES
$EOD
$EOJ
```

Figure BKP-4. Batch Control File


```
------------$JOB
09:30:01 AM  HELLO

09:30:03 AM  RSTS V7.0-05 The ARK  Job 25  KB1  20-Feb-79  09:30 AM
09:30:03 AM  [1/235
09:30:04 AM  Password:
09:30:04 AM  1 other user is logged in under this account



09:30:04 AM  Ready

------------$RUN [1,10]BACKUP
09:30:04 AM  RUN [1,10]BACKUP
09:30:05 AM  BACKUP  X7.0-05 RSTS V7.0-05 The APR

------------$DATA
09:30:06 AM  BAC[KUP], RES[TORE], LOA[DBINDER] OR LIS[T] ? BACKUP

09:30:07 AM  WORK FILE NAME<_SY:[  1,235]B20FEB.J25> ?

09:30:07 AM  LISTING FILE<_KB:> ?

09:30:08 AM  FROM DISK<_SY:> ?

09:30:09 AM  FROM FILES<[  1,235]*.*> ? [1,235]*.*/EXC:(*.BAK,*.J25)

09:30:10 AM  TO DEVICE<_MT:> ?

09:30:11 AM  BEGIN AT<[*,*]*.*> ?

09:30:12 AM  DELETE FILES<NONE> ?

09:30:13 AM  COMPARE FILES<NONE> ?
09:30:15 AM  *



09:30:15 AM                              Backup Run Listing

09:30:15 AM  Backup from '_SY    :' to '_MT    :

09:30:15 AM  Run started on 20-Feb-79 at 09:30 AM

09:30:15 AM  Work-File is _SY    :[1,235]B20FEB.J25

09:30:15 AM  Transfer  :
09:30:15 AM                [1,235]??????.???/EXC:([1,235]??????.BAK,[1,235]??????.J25)

09:30:15 AM  Begin at  : [*,*]*.*

09:30:15 AM  Delete    : <none>

09:30:15 AM  Compare   : <none>

09:30:15 AM  *
```

Figure BKP-5. Batch Log (Page 1 of 5)

```
09:38:19 AM  PHASE   : LIST COMPLETE
09:38:19 AM  ERRORS  : 0

09:38:19 AM  ELAPSED TIME : 0 SECONDS
09:38:19 AM  CPU TIME     : .4 SECONDS
09:38:19 AM  KCTS         : 68




09:38:33 AM  *

09:38:33 AM  PHASE   : LOOKUP/SELECT COMPLETE
09:38:33 AM  VOLUME 0 : 1
09:38:33 AM  ACCOUNTS : 1
09:38:33 AM  FILES    : 58
09:38:33 AM  BLOCKS   : 322
09:38:33 AM  ERRORS   : 0

09:38:33 AM  ELAPSED TIME : 13 SECONDS
09:38:33 AM  CPU TIME     : 4.6 SECONDS
09:38:33 AM  KCTS         : 784

09:38:36 AM  *
09:38:36 AM  PLEASE ENTER BACKUP SET NAME<B20FEB> -
09:38:37 AM  PLEASE ENTER EXPIRATION DATE<20-Feb-80> -
09:38:38 AM  PLEASE ENTER DENSITY IN BPI<800> -
09:38:39 AM  PLEASE ENTER THE PARITY<ODD> -
09:38:40 AM  MOUNT    DEVICE:          MT    :
09:38:40 AM                ID:          B20FEB
09:38:40 AM              SEQ#:          1
09:38:40 AM           DENSITY:          800 BPI
09:38:40 AM            PARITY:          ODD
09:38:40 AM          IDENTIFICATION WILL BE FINAL UPON SUCCESSFUL MOUNT
09:38:40 AM  DEVICE? MM0:
09:38:41 AM  %EXPIRATION DATE HAS NOT YET ARRIVED!
09:38:41 AM                ID:          B20FEB
09:38:41 AM              SEQ#:          1
09:38:41 AM           DENSITY:          800 BPI
09:38:41 AM            PARITY:          ODD
09:38:41 AM  EXPIRATION DATE:           20-Feb-80
09:38:41 AM  MOUNT IT ANYWAY<NO>?YES

09:38:44 AM  *

09:38:44 AM  PHASE   : MOUNT / DISMOUNT COMPLETE
09:38:44 AM  ERRORS  : 0

09:38:44 AM  ELAPSED TIME : 8 SECONDS
09:38:44 AM  CPU TIME     : 1.2 SECONDS
09:38:44 AM  KCTS         : 388

09:39:01 AM  *
09:39:01 AM  DATA UNRELIABLE - FILE OPENED BY ANOTHER USER IN FILE _DB1 :[1,235]BACKU
09:39:01 AM  P.LOG
09:39:01 AM  (ON TRANSFER)
```

Figure BKP-5. Batch Log (Page 2 of 5)


```
09:39:02 AM  *
09:39:02 AM  LENGTH CHANGE IN FILE _DB1 :[1,235]BACKUP.LOG
09:39:02 AM  (ON TRANSFER)
09:39:02 AM  *
09:39:02 AM  LENGTH AT LOOKUP =  3   LENGTH AT TRANSFER =  5
09:39:03 AM  *

09:39:03 AM  PHASE   : TRANSFER COMPLETE
09:39:03 AM  VOLUME 0 : 1
09:39:03 AM  ACCOUNTS : 1
09:39:03 AM  FILES    : 58
09:39:03 AM  BLOCKS   : 336
09:39:03 AM  ERRORS   : 2

09:39:03 AM  CURRENT VOLUME  : _MM0   :

09:39:03 AM  ELAPSED TIME : 19 SECONDS
09:39:03 AM  CPU TIME     : 6.7 SECONDS
09:39:03 AM  KCTS         : 748

09:39:09 AM  *

09:39:09 AM  PHASE   : INDEX DUMP COMPLETE
09:39:09 AM  ERRORS  : 0

09:39:09 AM  CURRENT VOLUME  : _MM0   :

09:39:09 AM  ELAPSED TIME : 5 SECONDS
09:39:09 AM  CPU TIME     : .7 SECONDS
09:39:09 AM  KCTS         : 110

09:39:14 AM  *
09:39:14 AM  DISMOUNT DEVICE:          MM0   :
09:39:14 AM                ID:          B20FEB
09:39:14 AM              SEQ#:          1(INDEX)
09:39:14 AM           DENSITY:          800 BPI
09:39:14 AM            PARITY:          ODD
09:39:14 AM  EXPIRATION DATE:           20-Feb-80
09:39:14 AM          PLEASE LABEL THIS VOLUME!

09:39:14 AM  *

09:39:14 AM  PHASE   : MOUNT / DISMOUNT COMPLETE
09:39:14 AM  ERRORS  : 0

09:39:14 AM  ELAPSED TIME : 3 SECONDS
09:39:14 AM  CPU TIME     : .3 SECONDS
09:39:14 AM  KCTS         : 48

09:39:17 AM  *



09:39:17 AM  Backup Set Name : B20FEB          Backup Device : _MM    :
09:39:17 AM  Volume Sequence # : 1(Index)      Owner : [1,235]
09:39:17 AM  Creation Date : 20-Feb-79         Expiration Date : 20-Feb-80
09:39:17 AM  Density :  800 BPI                Parity : ODD


09:39:17 AM    Account : [1,235]               Quota : 5P00
09:39:17 AM    Clustersize : 16
09:39:17 AM                                 Creation          Access
```

Figure BKP-5. Batch Log (Page 3 of 5)

```
09:39:17 AM   Name  Ext  Size  Prot    Date     Time       Date     Use  Min   Typ
09:39:17 AM                        (SOURCE DISK = DB1:)
09:39:17 AM  RFC    .CMP    39   60  09-Nov-78 03:59 PM  09-Nov-78     0 BAS4P  T
09:39:17 AM  DMPFIL.BAS     3   60  09-Nov-78 03:32 PM  09-Nov-78     0 ...PSX*T
09:39:17 AM  RFC1   .CMP    52   60  09-Nov-78 03:38 PM  09-Nov-78     0 BASIC2 T
09:39:17 AM  ST8    .LST     3   60  16-Nov-78 01:16 PM  16-Nov-78     0 ...PSX*T
09:39:17 AM  SYSO   .LST     3   60  16-Nov-78 01:23 PM  16-Nov-78     0 ...PSX*T
09:39:17 AM  SEQ    .COL     3   60  29-Nov-78 11:03 AM  29-Nov-78     0 ...PSX*T
09:39:18 AM  SEQ2   .OOL     2   60  29-Nov-78 12:15 PM  29-Nov-78     0 ...PSX*T
09:39:18 AM  CONT   .DAT   16C   60  07-Dec-78 08:57 AM  07-Dec-78     0 BAS4P  T
09:39:18 AM  PLACE  .BAS     1   60  07-Dec-78 09:47 AM  07-Dec-78     0 ...PSX*T
09:39:18 AM  IIO24  .B2S     1   60  29-Nov-78 09:14 AM  29-Nov-78     0 BP1COM T
09:39:18 AM  IIO24  .OOL     1   60  15-Dec-78 02:11 PM  15-Dec-78     0 BP2COM T
09:39:18 AM  ONLY   .MAS    11   60  15-Dec-78 02:19 PM  15-Dec-78     0 BASIC2*T
09:39:20 AM  CMP    .BAS     3   60  01-Jun-78 11:47 AM  01-Jun-78     0 BAS4P  T
09:39:20 AM  LOOP   .BAS     1   60  12-Jul-78 10:41 AM  12-Jul-78     0 BAS4P  T
09:39:20 AM  SPTIC  .BAS     2   60  05-Sep-78 04:50 PM  05-Sep-78     0 BAS4P  T
09:39:20 AM  TCOMP  .BAS     1   60  26-Oct-78 03:31 PM  26-Oct-78     0 BAS4P  T
09:39:20 AM  CCONT  .BAS     1   60  21-Nov-78 02:36 PM  21-Nov-78     0 BAS4P  T
09:39:20 AM  MONII  .BAS     1   60  21-Nov-78 02:36 PM  21-Nov-78     0 BAS4P  T
09:39:20 AM  CCPOS  .BAS     1   60  21-Nov-78 02:38 PM  21-Nov-78     0 BAS4P  T
09:39:20 AM  CMAIN  .BAS     1   60  21-Nov-78 02:39 PM  21-Nov-78     0 BAS4P  T
09:39:20 AM  DEF    .BAS     1   60  21-Nov-78 02:39 PM  21-Nov-78     0 BAS4P  T
09:39:20 AM  MCD120.BAS     1   60  27-Nov-78 02:22 PM  27-Nov-78     0 BAS4P  T
09:39:20 AM  VIRT1  .BAS     1   60  27-Nov-78 02:30 PM  27-Nov-78     0 ...PSX*T
09:39:20 AM  SEQ    .BAS     1   60  27-Nov-78 03:10 PM  27-Nov-78     0 BAS4P  T
09:39:20 AM  VIPTUL.BAS     1   60  28-Nov-78 10:49 AM  28-Nov-78     0 BAS4P  T
09:39:20 AM  DPDCRT.BAS     1   60  28-Nov-78 12:09 PM  28-Nov-78     0 BAS4P  T
09:39:20 AM  UPDACC.BAS     1   60  28-Nov-78 12:11 PM  28-Nov-78     0 BAS4P  T
09:39:20 AM  UPDERT.BAS     1   60  28-Nov-78 12:36 PM  28-Nov-78     0 BAS4P  T
09:39:20 AM  CPDPED.BAS     1   60  28-Nov-78 02:20 PM  28-Nov-78     0 BAS4P  T
09:39:21 AM  TEMTI  .BAS     1   60  28-Nov-78 02:34 PM  28-Nov-78     0 ...PSX*T
09:39:21 AM  TEMT   .BAS     1   60  28-Nov-78 02:47 PM  28-Nov-78     0 ...PSX*T
09:39:21 AM  TEMT2  .BAS     1   60  05-Dec-78 04:11 PM  05-Dec-78     0 ...REX*T
09:39:21 AM  FMSC   .BAS     1   60  06-Dec-78 10:20 AM  06-Dec-78     0 ...PSX*T
09:39:21 AM  MCMCMT.BAS     1   60  07-Dec-78 09:17 AM  07-Dec-78     0 BAS4P  T
09:39:22 AM  TTT66  .BAS     1   60  15-Dec-78 03:27 PM  15-Dec-78     0 BAS4P  T
09:39:22 AM  EVLMFS.BAS    14   60  12-Dec-78 05:10 PM  12-Dec-78     0 BAS4P  T
09:39:22 AM  EVLMFS.BAC   15C  124  22-Dec-78 12:55 PM  22-Dec-78     0 BAS4P  T
09:39:22 AM  EVLMFS.CRF    10   60  22-Dec-78 01:13 PM  22-Dec-78     0 BASIC2 T
09:39:23 AM  ANALYS.DMP    10   60  02-Jan-79 09:52 AM  02-Jan-79     0 BASIC2 T
09:39:22 AM  RSTS   .DMP     6   60  15-Jan-79 03:43 PM  15-Jan-79     0 RTIL   T
09:39:22 AM  RSTARL.IDX     4   60  17-Jan-79 09:58 AM  17-Jan-79     0 RTIL   T
09:39:23 AM  RSTARL.DMP     4   60  17-Jan-79 10:09 AM  17-Jan-79     0 RTIL   T
09:39:22 AM  SYSGEN.LOG    48   60  01-Feb-79 03:26 PM  01-Feb-79     0 ...REX*T
09:39:22 AM  SYSGEN.CTL     7   60  06-Feb-79 10:32 AM  06-Feb-79     0 BAS4P  T
09:39:22 AM  BCFBAC.LST    11   60  20-Feb-79 09:12 AM  20-Feb-79     0 BAS4P  T
09:39:22 AM  BCFLOA.LST     3   60  20-Feb-79 09:20 AM  20-Feb-79     0 BAS4P  T
09:39:22 AM  BIND20.IND    16   60  20-Feb-79 09:22 AM  20-Feb-79     0 BAS4P  T
09:39:22 AM  BCFLIB.LST    10   60  20-Feb-79 09:24 AM  20-Feb-79     0 BAS4P  T
09:39:22 AM  BACKUP.CTL     1   60  20-Feb-79 09:31 AM  20-Feb-79     0 BAS4P  T
09:39:22 AM  BACKUP.LOG     5   60  20-Feb-79 09:38 AM  20-Feb-79     0 BAS4P  T

09:39:22 AM                Errors :
09:39:23 AM       DATA UNRELIABLE - FILE OPENED BY ANOTHER USER on FILE
09:39:23 AM       LENGTH CHANGE on FILE
09:39:24 AM           Total of 2 errors encountered on FILE

09:39:24 AM     * Attributes associated with this file

09:39:24 AM    Account total of 326 blocks in 50 files on account [1,2351
09:39:24 AM        Total of 2 errors encountered on ACCOUNT
```

Figure BKP-5. Batch Log (Page 4 of 5)


```
09:39:24 AM  Volume total of 326 blocks in 50 files in 1 accounts on volume _RM    :
09:39:24 AM     Total of 2 errors encountered on VOLUME



09:39:24 AM  ********************************************************************************

09:39:24 AM  RUN total of 326 blocks in 50 files on 1 accounts on 1 Volumes
09:39:24 AM  RUN total of 2 errors
09:39:24 AM  RUN terminated on 20-Feb-79 at 09:39 AM

09:39:24 AM  ********************************************************************************

09:39:24 AM  *

09:39:24 AM  PHASE    : LIST COMPLETE
09:39:24 AM  VOLUME 0 : 1
09:39:24 AM  ACCOUNTS : 1
09:39:24 AM  FILES    : 50
09:39:24 AM  BLOCKS   : 326
09:39:24 AM  ERRORS   : 0


09:39:24 AM  ELAPSED TIME : 0 SECONDS
09:39:24 AM  CPU TIME     : 3.4 SECONDS
09:39:24 AM  BCTS         : 510




09:39:25 AM  *
09:39:25 AM  Ready

------------$EOD
------------$EOJ
09:39:25 AM  BYE
09:39:26 AM  Confirm: YES
09:39:27 AM  Saved all disk files; 637 blocks in use, 4368 free
09:39:27 AM  Job 25 User 1,235 logged off KB1 at 20-Feb-79 09:39 AM
09:39:27 AM  1 other user still logged in under this account
09:39:27 AM  System RSTS V7.0-05 The ARK
09:39:27 AM  Run time was 23.5 seconds
09:39:27 AM  Elapsed time was 1 minute
09:39:27 AM  Good morning
```

Figure BKP-5. Batch Log (Page 5 of 5)

SAVE/RESTORE PROGRAM

Features

- Disk to disk image copying with bad block handling,
- Bootable save set of disk-to-tape or disk-to-disk,
- Runnable off-line in the INIT.SYS environment, and
- Runnable on-line under the RT11 Run-Time System.

Rules for Using Media

1. The IMAGE Function
   a. IMAGE copies can only be made from disk to disk.
   b. IMAGE can only copy between like disks (disks of the same size).
2. The SAVE and RESTORE functions.
   a. Volumes of a single save set must be all disks or all tapes.
   b. A save set may consist of different types of disks.
   c. Tapes in a single save set must all be written at the same density.
3. Disks
   a. Must be formatted.
   b. Must be DSKINTed if SAVRES is being run on-line.
   c. Will automatically DSKINT if SAVRES is running off-line.
   d. The following blocks must not be bad:
      - The first 16 blocks for a save set disk, and
      - The first device cluster and the next n blocks, where n is the MFD clustersize for an IMAGE copy disk.
   e. Must not have more than 161 bad clusters.
   f. Cannot be logically mounted.

4. Tapes

   a. Must be long enough to hold the:

      - Save set label and header information
      - Bootstrap, and
      - INIT.SYS .

   b. A bad block is considered to be the end of a volume.

   c. The default density is the lowest allowed on the first magtape drive used to produce a save set.

SAVE/RESTORE Options

Table BKP-1. SAVE/RESTORE Options

Option	Valid for Functions			
	IMAGE	SAVE	RESTORE	IDENTIFY
*EXPIRATION EXPIRATION: date NOEXPIRATION		X		
VERIFY *NOVERIFY	X	X	X	
*STATS NOSTATS	X	X	X	X
*ERROR NOERROR	X	X	X	X

* Default Condition (the default expiration date is one year from the date of the SAVE operation).

```
RUN $SAVRES

SAV/RES Function:HELP

The valid SAV/RES functions are:
    SAVE        Make a safe copy of a RSTS/E file-structured disk.
    RESTORE     Restore a RSTS/E file-structured disk from a SAVE Set.
    IMAGE       Copy a volume image from a RSTS/E file-structured disk
                to a like disk.
    IDENTIFY    List the characteristics of a SAVE Set or RSTS/E disk.
    HELP        Print this list of acceptable responses.
Only the first two characters need be typed.

SAV/RES Function: IDENTIFY

From Device? DM0:

          Device:       DM0:
        Pack ID:        SYSPAR
Pack ClusterSize:       8
   Creation date:       Wednesday, 07-Feb-79

SAV/RES Function: SAVE

From RSTS disk? DM0:

*** Pack ID/default Save Set Name is SYSPAR

To Device? MM0:SAVEST

*** Save Set Name is SAVEST

Expiration Date <28-Feb-80>?

Verify (Yes or No) <NO>?

Proceed (Yes or No)? Y

*** Initializing first SAVE volume

*** Begin SAVE from DM0: to MM0: at 11:00 AM

 Discount Device:       MM0:
        Set Name:       SAVEST
          Seq #:        1
        Density:        1600
  Creation date:        Tuesday, 20-Feb-79
 Expiration date:       Wednesday, 28-Feb-80

       Please label this volume!

--- SAVE operation completed at 11:12 AM

                Summary Report

SAVE of DM0:SYSPAR to Save Set SAVEST

    Input Device:       DM0:
        Pack ID:        SYSPAR
Pack Clustersize:       8
   Creation date:       Wednesday, 07-Feb-79

   Output Device:       Magtape
        Set Name:       SAVEST
  # of volumes:         1
        Density:        1600
```

Figure BKP-6. IDENTIFY and SAVE Functions (Page 1 of 2)


```
   Creation date:       Tuesday, 20-Feb-79
 Expiration date:       Wednesday, 28-Feb-80

SAVE started on Tuesday, 20-Feb-79, at 11:00 AM

                Run Statistics

Transfer Totals:

        Total of 14936 blocks transferred

Error Totals:

        Total of    0 new bad blocks encountered on source.

Timing Totals:

        Total elapsed time:   0 hrs..   6 mins..   7 secs.
        Total wait    time:   0 hrs..   0 mins..   0 secs.
        Total process time:   0 hrs..   6 mins..   7 secs.

SAV/RES Function: ^Z

Ready
```

Figure BKP-6. IDENTIFY and SAVE Functions (Page 2 of 2)

```
RUN $SAVRES

SAV/RES Function: RESTORE

From Device? MM0:

*** Save Set Name/default Pack ID is SAVEST

To RSTS DM: Disk? DM0:SYSPAK

*** The volume on DM0: is a RSTS disk
*** Pack ID is SYSPAK

*** Pack will be reinitialized
Mount it anyway <NO>? Y

*** Pack ID is SYSPAK

Verify (Yes or No) <NO>?

Proceed (Yes or No)? Y

*** Begin RESTORE from MM0: to DM0: at 11:17 AM

 Dismount Device:       MM0:
        Set Name:       SAVEST
          Seq #:        1
         Density:       1600
   Creation date:       Tuesday, 20-Feb-79
 Expiration date:       Wednesday, 20-Feb-80

 Dismount Device:       DM0:
         Pack ID:       SYSPAK
 Pack Clustersize:      8
   Creation date:       Wednesday, 07-Feb-79

       Please label this volume!

--- RESTORE operation completed at 11:21 AM
```

Figure BKP-7. RESTORE Function (Page 1 of 2)


```
                Summary Report

RESTORE of Save Set SAVEST to DM0:SYSPAK

     Input Device:      Magtape
         Set Name:      SAVEST
     # of volumes:      1
          Density:      1600
    Creation date:      Tuesday, 20-Feb-79
  Expiration date:      Wednesday, 20-Feb-80

    Output Device:      DM0:
          Pack ID:      SYSPAK
  Pack Clustersize:     8
    Creation date:      Wednesday, 07-Feb-79

RESTORE started on Tuesday, 20-Feb-79, at 11:17 AM

                Run Statistics

Transfer Totals:

        Total of 14824 blocks transferred

Error Totals:

        Total of    0 new bad blocks encountered on destination.

        0 files structurally altered.

Timing Totals:

        Total elapsed time:   0 hrs.,   4 mins.,   6 secs.
        Total wait    time:   0 hrs.,   0 mins.,   0 secs.
        Total process time:   0 hrs.,   4 mins.,   6 secs.

SAV/RES Function: ^Z

Ready
```

Figure BKP-7. RESTORE Function (Page 2 of 2)

```
RUN $SAVRES

SAV/RES Funtion: IMAGE/NOERROR

From RSTS disk? DB0:

*** Input Pack ID/default Output Pack ID is SOURCE

To RSTS DB: Disk? DB1:DESTIN

*** The volume on DB1: is a RSTS disk
*** Pack ID is OUTPUT

*** Pack will be reinitialized
Mount it anyway <NO>? YES

*** Output Pack ID is DESTIN

Verify (Yes or No)? YES

Proceed (Yes or NO)? YES

*** Begin IMAGE copy from DB0: to DB1: at 11:02 AM

*** Begin VERIFY pass from DB0: to DB1: at 11:04 AM

*** 0 differences found

   Dismount Device:   DB1:
          Pack ID:    DESTIN
  Pack Clustersize:   8
     Creation date:   Monday, 22-Jan-79

      Please label this volume!

--- IMAGE copy operation completed at 11:07 AM
```

Figure BKP-8. IMAGE Function (Page 1 of 2)


```
                Summary Report

IMAGE copy of DB0:SOURCE to DB1:DESTIN

      Input Device:   DB0:
            Pack ID:  SOURCE
  Pack Clustersize:   8
     Creation date:   Monday, 22-Jan-79

     Output Device:   DB1:
            Pack ID:  DESTIN
  Pack Clustersize:   8
     Creation date:   Monday, 22-Jan-79

IMAGE copy started on Monday, 22-Jan-79, at 11:02 AM

                Run Statistics

Transfer Totals:

     Total of 24400 blocks transferred

Error Totals:

     Total of     0 bad compares.
     Total of     0 new bad blocks encountered on source.
     Total of     0 new bad blocks encountered on destination.

     0 files structurally altered.

Timing Totals:

     Total elapsed time:   0 hrs.,   5 mins.,  22 secs.
     Total wait    time:   0 hrs.,   0 mins.,   0 secs.
     Total process time:   0 hrs.,   5 mins.,  22 secs.

SAV/RES Function: ^Z

Ready
```

Figure BKP-8. IMAGE Function (Page 2 of 2)

Additional SAVE/RESTORE Documentation

The RSTS/E SAVE/RESTORE system program is a disk backup and copy utility that provides you with four operational functions. SAVE, RESTORE, and IMAGE perform the backup, recreate, and copy functions respectively, while IDENTIFY extracts label information from a SAVE/RESTORE volume or RSTS/E disk. This documentation describes these four functions, how they are invoked and how and when they should be used.

When to Use SAVE/RESTORE

SAVE/RESTORE and BACKUP each possess similar but different system functions. SAVE/RESTORE was developed to provide a non-selective, fast volume backup, and copy capability that requires few operator responses during the operational dialogue. Unlike BACKUP, SAVE/RESTORE processes entire volumes only, and does not allow selective file transfers, or file deletes. SAVE/RESTORE does not require operator intervention when it encounters bad blocks. For these reasons, use the SAVE/RESTORE package when you need to:

1. Create a fast, reliable copy of an entire RSTS/E disk,
2. Backup files that are larger than 65,535 blocks, and
3. Make a fast copy to a disk that contains bad blocks which may not be reflected in the BADB.SYS file.

Definitions of SAVE/RESTORE Terms

The SAVE/RESTORE terms defined here are used throughout this documentation.

SAVE Format The format of the output written by a SAVE operation and read by a RESTORE operation.

SAVE Set The set of magtapes or disks created by a SAVE operation. A SAVE Set must be composed entirely of disks or tapes, not a combination of the two device types. However, different drive types may be mixed within one set, for example, two RK05's and one RK06.

SAVE Volume One of the magnetic tapes or disks of a SAVE Set.

SAVE Set Name One to six alphanumeric characters used to identify a SAVE set. By default, the SAVE Set Name is the same as the Pack ID from which it was created. You can specify another name for the SAVE Set rather than accept the default.

LIKE Disks Like disks are units of the same device size. For example, SAVE/RESTORE considers the following devices LIKE:

- An RP06 is like another RP06
- An RM02 is like an RM03
- An RP04 is like an RP05
- An RK05J is like one unit of an RK05F

As a further clarification, if you copy an RM02 to a SAVE Set, that set can later be RESTOREd to either an RM03 or another RM02.

Running SAVE/RESTORE

You can invoke the SAVE/RESTORE program off-line while you are running the system initialization code (INIT.SYS). To do this, type SAVRES on the console terminal in response to the INIT "Option:" prompt. After you press the RETURN key, SAVE/RESTORE requests the current date and time then responds with its own "SAV/RES Function:" prompt. The following example illustrates the proper response to the INIT "Option:" prompt and the appearance of the SAVE/RESTORE program prompt.

```
Option: SAVRES

DD-MM-YY? 12-JAN-79
HH:MM? 12:15

SAV/RES Function:
```

To invoke SAVE/RESTORE on-line, you must type RUN $SAVRES, then press the RETURN key. After you issue these commands, RSTS/E prints the SAVE/RESTORE program prompt, "SAV/RES Function:". The example that follows represents the steps you take to initiate SAVE/RESTORE on-line.

```
RUN $SAVRES <CR>
SAV/RES Function:
```

Once the "SAV/RES Function:" prompt appears, either as a result of invoking SAVE/RESTORE off-line or on-line, you are under the control of SAVE/RESTORE and are ready for backup processing.

Next, select from the following table one of the following four operational functions: SA[VE], RE[STORE], IM[AGE], or ID[ENTIFY]. Unless you need to exit SAVE/RESTORE (by using <LF> or ^Z) or need further help (by typing HE[LP] or <CR>), type one of the four operational functions on the console terminal in response to the "SAV/RES Function:" prompt. The following example illustrates this procedure:

```
SAV/RES Function: SAVE
```


SAVE/RESTORE Functions

SA[VE]	Creates a copy of a RSTS/E file-structured disk. The SA[VE] function backs up to disk or tape.
RE[STORE]	Recreates a RSTS/E file-structured disk from a SAVE Set.
IM[AGE]	Copies a RSTS/E file-structured disk to a LIKE disk.
ID[entify]	Prints label information and other volume characteristics of a SAVE volume or a RSTS/E file-structured disk.
HE[LP]	Prints a table of SAVE/RESTORE functions.
<LF>	Press the LINE FEED key if you want to exit the SAVE/RESTORE program environment. If you invoke SAVE/RESTORE with a RUN command, pressing <LF> returns you to your default Run-Time System. If you are running SAVE/RESTORE off-line with INIT.SYS, pressing <LF> returns you to the "Option:" prompt issued by the INIT program.
<CR>	Press the RETURN key and the "Type HELP for help" message appears on your terminal.
^C	Type ^C to produce the same result as pressing the LINE FEED key.
^Z	Type ^Z to produce the same result as pressing the LINE FEED key.
other	Produces the same result as pressing the RETURN key.

Press the RETURN key when you finish entering the selected function. This procedure invokes the SAVE/RESTORE dialogue.

Note that the function response in the previous example represents only one of three different ways you can answer the "SAV/RES Function:" prompt. The response to the prompt determines the type of dialogue to be carried out as well as the operation to be performed. A description and an example of each type of response appears in the following table.

Table BKP-2. Response Types for the "SAV/RES Function:" Prompt

Response Type	Example	Description
Function (by itself)	SAV/RES Function:SAVE	If only a function is given, the full dialogue follows
Function plus Option(s)	SAV/RES Function: SAVE/EXP:10-JUL-79	If the function is followed by options, dialogue questions will follow only for the devices to be used and the unspecified options.
Full Function Command Line	SAV/RES Function: IMAGE DM1:=DM0: MYPAK/VER/NOERR	If a complete command line is given, no dialogue questions will be asked.

During the run, SAVE/RESTORE informs you of its operational status by printing messages on your terminal. Some of the messages are intended for your information; others notify you of error conditions that might require your attention. This latter group consists of errors that can occur during the dialogue, mount, transfer, or verification phases of SAVE/RESTORE.

When the program encounters an error during a dialogue or mount operation, it instructs you to rectify the problem, then asks you to re-enter the appropriate command. These errors never affect the integrity of the data being transferred.

On the other hand, errors that occur while SAVE/RESTORE is transferring or verifying data can jeopardize this data. In this case, SAVE/RESTORE resolves the problem, aborts the run, or requests whether to continue or abort the program. In error situations of this type, the program gives you as much information as it can to help protect your data. That is, SAVE/RESTORE lets you know what it is doing by issuing informational messages as well as error messages to your terminal.

Finally, SAVE/RESTORE prints summary information about SAVE, RESTORE, or IMAGE operations. This report describes the operation performed, device information, run start date and time statistics in addition to run total statistics. The number of errors encountered during the operation are listed as part of the run total statistics. The program prints this report at the end of a run.

SAVE/RESTORE Options

You can include SAVE/RESTORE options (switches) in a function response or in a single line command. The options that appear in the following table are of two types. The first two pairs of options correspond to SAVE/RESTORE dialogue questions and have option defaults. If neither of a pair of options is attached to a function response, an appropriate question will appear in the dialogue. However, if one of the pairs is specified, the corresponding question will not appear in the dialogue. If neither of the pair is specified in a full function command line, the default is assumed.

The last two options in the table have no corresponding dialogue questions. The /ERROR option is a default setting which means that SAVE/RESTORE will print non-fatal error messages and continue processing. Consequently, if you want to have all errors abort the program, you must attach the /NOERROR option to the function response or the full function command line. The /STATS option is a default setting which means SAVE/RESTORE will automatically print a summary report at the end of each run. Consequently, if you want to suppress the summary report, you must attach the /NOSTATS option to your function response.

Table BKP-3. SAVE/RESTORE Options (page 1 of 2)

Option	Default	Description
EX[PIRATION][:date] NOEX[PIRATION]	EXPIRATION	This option is legal only for a SAVE operation. The date argument that you insert is the date after which you may overwrite the destination volume. You are warned by SAVE/RESTORE of any attempt to overwrite the volume(s) any time prior to the specified date. If you specify /EX[PIRATION] with no date argument, SAVE/RESTORE establishes the expiration date as one year after the current date. If you specify /NOEX[PIRATION], SAVE/RESTORE uses the current date as the expiration date.
VE[RIFY] NOVE[RIFY]	NOVERIFY	This option is legal only for a SAVE, RESTORE, or IMAGE operation. If you specify the VE[RIFY] option, SAVE/RESTORE compares the volume(s) to ensure that the transfer was accurate.
ST[ATS] NOST[ATS]	STATS	SAVE / RESTORE prints a summary report at the end of a run, unless you specify the /NOST[ATS] switch with either a function response or a single line command.

Table BKP-3. SAVE/RESTORE Option (page 2 of 2)

Option	Default	Description
NOER[ROR] ER[ROR]	ERROR	If you omit the /NOERROR option and a non-fatal error occurs during a SAVE/RESTORE run, an error message appears and processing continues. If you specify NOER[ROR] and any run-time error occurs, the program terminates after printing an error message.

Dialogue

The SAVE/RESTORE dialogue begins after you invoke the SAVE/RESTORE program and enter a function response to the "SAV/RES Function:" prompt. If you respond with SAVE, RESTORE, IMAGE, or IDENTIFY, the program initiates a set of dialogue questions. All questions in the dialogue have a short form or a long form. The short form appears automatically; the long form prints only if you press the RETURN key in response to the short form question. The long form provides information that may help you with an answer.

Some of the dialogue questions have default values; these are printed in angle brackets next to the question. You may select the default value by pressing the LINE FEED key or by typing the proper default response. If the LINE FEED key is pressed in response to a question with no default value, the question is repeated.

There are two valid responses, ^Z and ^C, in the SAVE/RESTORE package that can alter the existing operating status. For example, if you type ^Z at any point in the dialogue, SAVE/RESTORE returns you to the previous question. If you want to abort the run at any time, type ^C. After the abort, you will return to the "Option:" prompt if you are running off-line under the control of the INIT.SYS

program or to your default Run-Time System prompt if you invoked SAVE/RESTORE on-line with a RUN command. SAVE/RESTORE automatically returns to the "SAV/RES Function:" prompt after the execution of a specified function. This occurs when you are either operating on-line or off-line. You may exit from the INIT environment version of SAVE/RESTORE to the "Option:" prompt by typing LINE FEED or ^Z to the "SAV/RES Function:" prompt.

SAVE/RESTORE Volumes

You can perform SAVE/RESTORE operations on two classes of volumes: disks and tapes. The SAVE Set, a set of tapes or disks created from a SAVE operation, cannot include both disks and tapes. However, the SAVE Set may include one or more volumes of the same class. In other words, a tape volume SAVE Set could consist of two MM devices and one MT device while one RP04 and two RK05 disks constitute a legal disk SAVE Set. SAVE/RESTORE, to repeat, does not allow any disk/tape combination SAVE Sets.

A legal SAVE/RESTORE disk device is any disk supported by RSTS/E. Legal disk devices must meet the following requirements:

1. Must be formatted; that is, contain sector and track information.
2. All disks used by the on-line version of SAVE/RESTORE must have a valid RSTS/E file structure; that is, the disk device must be formatted and DSKINTed prior to its use.
3. The first 16 blocks on a disk must not contain any bad blocks.
4. RSTS/E output disk can contain no more than 161 bad clusters (the RSTS/E limit).

Magnetic tape is the only valid SAVE/RESTORE tape medium; DECtape or cassette are not supported. The statements here define what constitutes a legal tape in the SAVE/RESTORE package.

1. The magnetic tape must be long enough to contain the SAVE Set label and header information, a bootstrap, and INIT.SYS which is on the first volume only.
2. A bad block will be considered to be the end of the tape volume during the SAVE operation.
3. The default density is the lowest density allowed by the drive on which the first volume is mounted. Subsequent volumes cannot be written at any other density. Therefore, ensure that the density used on the first volume is compatible with all other tape drives to be used.

Device Specifications

Device specifications are required as responses to several dialogue questions. These specifications must be in the following general format.

<device>[:[<id>]][/<option(s)>]

where <device> is a two character device type (for disk either "Dx" or "Rx" is acceptable) followed by a single digit unit number in the range 0 to 7. The <id> represents either a SAVE Set name or a disk pack identification and the <options> are selected from the following table.

Table BKP-4. SAVE/RESTORE Device Specifications
(page 1 of 2)

Options	Description
SCR[atch]	The SCR[ATCH] option is legal only on an output volume device specification. If you use the SCR[ATCH] option, SAVE/RESTORE by-passes most volume label checking. SAVE/RESTORE always checks the label to see if an output volume contains a SAVE Set written at the current density and ensures that it is not a volume from the SAVE Set currently being written. It also checks destination disks for bad block information.

Table BKP-4. SAVE/RESTORE Device Specifications
(page 2 of 2)

Options	Description
DEN[SITY]:[800] [1600]	The DEN[SITY] option is legal only for tape. With this option you can specify the density at which the tape is to be written or read. If SAVE/RESTORE cannot read the tape at the specified density, it will try to read the tape at the other legal density setting.

Checking the Input Volume

SAVE/RESTORE checks the input volume prior to any actual data transfer and warns you of any problem that it foresees in saving, restoring, or copying the volume. All of the error messages that are printed at this point (at least concerning the input volume) are warning messages only; you can recover from them by following appropriate procedures. The error messages that result from a problem with the input volume are listed here and described separately below.

- %%% Input disk has only nn% free clusters.
  Mount it anyway <NO>?
- %%% This is not the correct volume.
- %%% This volume has no label.
- %%% Input disk should be "CLEANed".

There must be at least as many good pack clusters on an output RSTS/E disk as there are clusters to be transferred from the original input disk. If SAVE/RESTORE determines that the input disk has very few clusters that are not allocated, it warns you that you may encounter problems copying or restoring the disk.

%%% Input disk has only nn% free clusters.
Mount it anyway <NO>?

Type NO or press the LINE FEED key to return to the previous device prompt. If you type YES, you are instructing SAVE/RESTORE to proceed with the SAVE or IMAGE operation. In the case of an IMAGE copy, the program also

checks the output volume to see if the transfer can be made. No further checking takes place in the case of the SAVE operation. At this point, you may want to free up space on the input disk by deleting unnecessary files. This process will increase your chances of eventually completing a successful RESTORE.

When you specify a Pack ID or a SAVE Set name with the input device specification, SAVE/RESTORE compares the specified name with the name already written on the volume. If the names do not match, the following warning message is printed and the input device prompt is asked again:

%%% This is not the correct volume.

A dismount request follows the above warning message. It lists useful information from the volume label but does so only if it is a SAVE Set or a RSTS/E file-structured disk.

If the input volume is neither a RSTS/E disk nor a SAVE Set, the terminal displays the following warning message and the input device prompt is asked again:

%%% This volume has no label

If you remove a disk from a drive without logically dismounting it and subsequently attempt to use it as the input for a SAVE or IMAGE copy operation, SAVE/RESTORE will print the message that follows:

%%% Input disk should be "CLEANed".

To perform the cleaning operation simply invoke the ONLCLN program if running on-line.

Checking the Output Volume

SAVE/RESTORE checks the output volume before SAVEing, RESTOREing, or IMAGE copying a RSTS/E disk. This procedure ensures that the SAVE/RESTORE operations will proceed to completion with the least chance of error. The messages that SAVE/RESTORE issues as a result of checking the output

volume, pertain to the expiration date, the label, and the available data space on the destination device. Some of the messages are printed only for your information; the rest warn you of serious output device problems. The paragraphs that follow exhibit the messages, describe why they appear, and explain how to respond to them.

When the expiration date of an output volume, labelled as a SAVE Set, has passed, SAVE/RESTORE prints this series of messages:

```
*** The volume on dev: is SAVE Set <xxxxxx>
      Density: nnn
Creation date: day-of-week, dd-mmm-yy
Expiration date: day-of-week, dd-mmm-yy
```

and proceeds to the next dialogue question. If you decide not to use the expired date volume, type ^Z to return to the "To Device?" output dialogue question.

If the output volume is labelled as a SAVE Set and its expiration date has not passed, the following series of messages are printed:

```
*** The volume on dev: is SAVE Set <xxxxxx>
      Density: nnn
Creation date: day-of-week, dd-mmm-yy
Expiration date: day-of-week, dd-mmm-yy
%%% Its expiration date has not passed
Mount it anyway <NO>?
```

If you type NO or press the LINE FEED key in response to the mount question, SAVE/RESTORE returns to the previous device prompt. SAVE/RESTORE will proceed to write over the existing SAVE Set if you type YES. If you press the RETURN key, the following message appears:

```
*** This is your last chance to prevent the
*** loss of any data on the output volume.
Mount it anyway <NO>?
```


SAVE/RESTORE notifies you with the following message that the output volume you are using is labelled as a RSTS/E disk:

```
*** The volume on dev: is a RSTS disk
*** Pack ID is <xxxxxx>
*** Pack will be reinitialized
Mount it anyway <NO>?
```

If you type NO, or press the LINE FEED key, you direct SAVE/RESTORE to return to the previous output prompt. A YES response, at this point, indicates that you wish to use the present volume.

If you RESTORE or IMAGE copy a disk, SAVE/RESTORE checks to see whether the total number of clusters on the disk minus the number of clusters with known bad blocks is greater than or equal to the number of allocated clusters on the original source disk. If SAVE/RESTORE determines that there is not enough space and the transfer cannot be made, it prints:

```
%%% Too many bad blocks on output disk
```

and then returns to the previous device prompt. If, however, the program determines that there are very few free blocks (<10%), it prints the following warning:

```
%%% Only nnt of the output disk clusters are available for relocation.
Mount it anyway <NO>?
```

You must decide at this point whether the number of free clusters is large enough to make possible a successful SAVE/RESTORE operation. If you decide it is not, type NO or press LINE FEED to return to the output prompt. Type YES to direct SAVE/RESTORE to proceed with the operation if you think there are enough free clusters available. Whenever this warning appears, DIGITAL recommends that, when it is possible, you use another disk for output. Follow this procedure since there exists a higher than usual possibility that the operation will not succeed. Note that DIGITAL does not guarantee that the operation will succeed even if this message does not appear.

SAVE/RESTORE normally transfers each allocated pack cluster from the original source disk to the same cluster on the output RSTS/E disk. If the output cluster is found to be bad, SAVE/RESTORE will try to relocate the data. If the cluster size of the item being moved is larger than the pack cluster size, there must be enough contiguous free pack clusters on the disk to accommodate the entire entity cluster. If there is not SAVE/RESTORE will abort the operation.

If the specified output is not a RSTS/E or a SAVE Set disk it prints the new SAVE Set name and proceeds without further notice. The program does not check the output volume label if you specified the /SCR[ATCH] option, except to determine whether it is a volume of the SAVE Set that is currently being created. It always attempts to recover bad block information from a RSTS file structured and SAVE disk.

GENERAL COMPARISON OF BACKUP PROGRAMS

Table BKP-5. Comparison of Backup Programs

	BACKUP	SAVE/ RESTORE	PIP	RMSBCK/ RMSRST
Copy Large Files	No	Yes	Yes	Yes (5)
Maintain Position of Placed Files	No	Yes(1)	Yes(2)	No
Selective Backup and Restore	Yes	No	Yes	Yes
Backup and Restore Public Structure On-line	Yes	No	Yes(3)	Yes
Backup and Restore Private Structure On-line	Yes	Yes	Yes(3)	Yes
Compensates for Bad Blocks on the Output Medium	Yes	Yes(4)	No	No
Creates Bootable Volumes	No	Yes	No	No
Runs Off-line	No	Yes	No	No
Restores Files to Different Accounts then Backed up From	No	No	Yes	Yes

(1) SAVE/RESTORE maintains the position of placed files upon recovery as long as bad blocks on the output disk do not interfere with the placement of the first cluster of the file.

(2) PIP maintains the position of a file if the user specifies that the output file is to be positioned. If PIP cannot position a file at the user specified block, it will attempt to create the file between the specified block and the end of the disk. If it still cannot find a space to create the file, it will start at the beginning of the disk and attempt to find space for the file. If the file is located at a lower numbered disk block than requested, it is not marked in the directory as a placed file. If it cannot be created at all, the "?No room for user on device" error is generated.

(3) PIP does not store project,programmer numbers in ANSI magtape labels. Therefore, it is difficult to recover files to multiple accounts from ANSI formatted magtapes. DOS magtape labels do store project,programmer numbers, but do not store RMS-11 attributes. DOS format also does not support multi-volume magtape files and therefore cannot support large files that do not fit on one volume.

(4) Bad blocks on the output disk cause contiguous files to be made non-contiguous.

(5) RMSBCK can copy large files as long as they can fit on the current output volume. There is no multi-volume capability allowing files to cross a volume boundary.

TRANSFERRING RMS RELATIVE AND INDEXED FILES TO OTHER NON-RSTS/E OPERATING SYSTEMS

1. File or files fit on one magtape

   a. Use RMSBCK on the RSTS/E system.

   b. Use RMSRST on the other system.

2. File requires more than one magtape

   a. Use RMSCNV to do a disk to disk conversion to an RMS Sequential file.

   b. Use PIP to transfer the sequential file to ANSI magtape with no switches.

   c. On the other system

      1. Relative file - use RMSDEF and RMSCNV

      2. Indexed file - use RMSDEF and RMSIFL

3. Another possible consideration, for example:

   a. If you have an 86,000-block indexed file (400,000 82-byte records), the back-up (RMSBCK) copy of the file requires at least two magnetic tape volumes.

   b. However, if you convert (RMSCNV) the file to tape with 4096-character blocks, you need only one 2400-foot tape volume.

   c. You can then restore the file quickly with RMSIFL.

RSTS/E V7.0 SWS UPDATE

MUP

Management Utility Programs

Management Utility Programs

CONTENTS

Introduction..... MUP-1

Device Testing Package (DEVTST)..... MUP-3
DEVTST Documentation..... MUP-3
Introduction to DEVTST..... MUP-4
Conventions..... MUP-5
DEVTST Operating Procedures..... MUP-6
DEVTST Dialogue Errors..... MUP-10

CPEXER..... MUP-11
CPEXER Tests..... MUP-12
CPEXER Dialogue..... MUP-13
CPEXER Errors..... MUP-13

CPUTST..... MUP-15
CPUTST Dialogue..... MUP-15
CPUTST Errors..... MUP-15

DSKEXR..... MUP-17
DSKEXR Dialogue..... MUP-17
DSKEXR Errors..... MUP-17

DSKSEK..... MUP-19
DSKSEK Dialogue..... MUP-19
DSKSEK Errors..... MUP-20

DTEXER..... MUP-21
DTEXER Dialogue..... MUP-21
DTEXER Errors..... MUP-21

DXEXER..... MUP-22
DXEXER Dialogue..... MUP-23
DXEXER Errors..... MUP-23

KBEXER..... MUP-24
KBEXER Dialogue..... MUP-24
KBEXER Errors..... MUP-25

LPEXER..... MUP-25
LPEXER Dialogue..... MUP-25
LPEXER Errors..... MUP-26

MTEXER..... MUP-26
MTEXER Dialogue..... MUP-27
MTEXER Errors..... MUP-27

PPEXER..... MUP-28
PPEXER Dialogue..... MUP-28
PPEXER Errors..... MUP-28

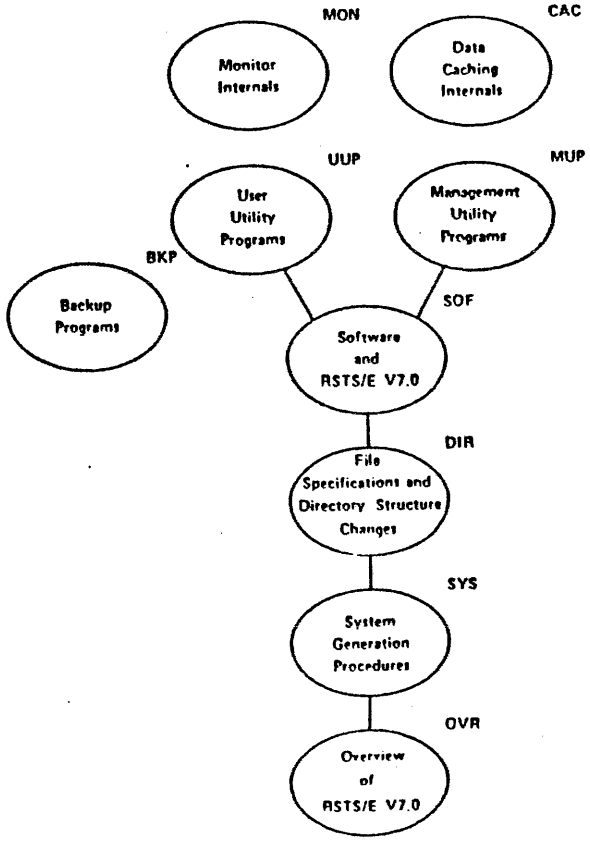
PREXER..... MUP-29
PREXER Dialogue..... MUP-29
PREXER Errors..... MUP-29

ON-LINE CLEAN Program - ONCLN..... MUP-31

Enhanced Management Utility Programs..... MUP-32
Spooling Package..... MUP-32
SYSTAT Program..... MUP-35
VT50PY (VT5DPY) Program..... MUP-41
TTYSET Program..... MUP-42
Error Logging Package..... MUP-43
UTILTY Program..... MUP-48
ANALYS Program..... MUP-51
SHUTUP Program..... MUP-58
LOGIN Program..... MUP-59
MONEY Program..... MUP-59
DSKINT Program..... MUP-59
REACT Program..... MUP-59
ODT Program..... MUP-59

Programs That Use Data Caching (If Enabled)..... MUP-60

COURSE MAP



Management Utility Programs

INTRODUCTION

Utility programs supplied with RSTS/E, which relate to system manager functions, allow the manager to:

- Anticipate possible hardware problems and take appropriate action.
- Track system activity during time sharing sessions.
- Gather vital information in case of a system crash.

With RSTS/E V7.0 new management utilities have been supplied while others have been enhanced. With the exception of system generation and back up related programs, this module covers these new and enhanced programs.

This page is for notes.

Management Utility Programs

DEVICE TESTING PACKAGE (DEVTST)

1. Supersedes the UETP package
2. Includes exercisers for:
   - o CPU
   - o Disk
   - o TU56 DECtape
   - o Terminal
   - o Floppy disk
   - o Line printer
   - o Magnetic tape
   - o Paper tape reader
   - o Paper tape punch
3. Executes under BASIC-PLUS, BASIC-PLUS-2, or RSX Run-Time Systems.
4. Limitations:
   - o CPU exercisers require trigonometric functions and matrix operations.
   - o No control programs exist

DEVTST Documentation

The DEVTST package exercises hardware on the RSTS/E V7.0 system during normal time-sharing operations. DEVTST is used to stimulate hardware errors on the system.

Introduction to DEVTST

DEVTST is a package of programs that provides the customer, System Manager or Digital Field Service personnel with tests that verify the reliability of RSTS/E supported hardware. Execution of any DEVTST program is limited to users with privileged accounts. Operational data, conventions, complete device testing information and error messages are explained in this documentation. After execution of individual DEVTST exercisers, users should check the System Error Log for additional errors.

The DEVTST exercisers are tests that can be initiated whenever hardware trouble is suspected. These device exercisers generate excess activity in an attempt to make the hardware fail during operation.

The DEVTST package should be used in conjunction with the RSTS/E Error Logging package. The Error Logging facility provides the means for determining detailed information on hardware errors. In addition, the DEVTST programs return error messages which may be useful in determining the failure; however, these messages are not as specific as the information provided by the System Error Log.

Conventions

The following conventions are used throughout the DEVTST package dialogue:

< >	Indicates default answer.
( )	Shows all possible replies.
[ ]	Gives an optional reply; for example ASC[II], means enter either ASC, ASCI or ASCII.
<ALT MODE> or <ESC>	This key moves the program back one question. If you type this key as a response to the first question, DEVTST will return to the RSTS/E command level.
CTRL/Z	The CTRL/Z combination causes the program to stop execution and return to RSTS/E command level. CTRL/Z is detected only when the system checks for new input.
CTRL/C	Typing CTRL/C causes the program to stop execution immediately and return to a command level.
RET	Use the RET key to terminate the input line.
Detach <NO>	All of the device exercisers have the ability to detach; this allows you to run an exerciser while keeping your terminal free for other operations.
-1	Negative one, when used as a value for number of iterations, pages or lines, tells the program to run indefinitely, or until you enter CTRL/C.

DEVTST Operating Procedures

To run a DEVTST program contained within the system library account [1,2], type:

```
RUN $program name
```

For example:

```
RUN $KBEXER
```

To run a DEVTST program from another privileged account, type:

```
RUN [account number]program name
```

For example:

```
RUN [1,215]KBEXER
```

Where program name is one of the exercisers in Table MUP-1.

Table MUP-1. DEVTST Programs

Program Name	Hardware Tested
CPEXER	CPU
CPUTST	CPU
DSKEXR	FILE STRUCTURED DISK
DSKSEK	NON-FILE STRUCTURED DISK
DTEXER	DECTAPE
DXEXER	FLOPPY DISK
KBEXER	TERMINAL
LPEXER	LINE PRINTER
MTEXER	MAGNETIC TAPE
PPEXER	PAPER TAPE PUNCH
PREXER	PAPER TAPE READER

A complete list of RSTS/E supported hardware, including designators and the associated devices, appears in Table MUP-2.

Table MUP-2. RSTS/E Device Designators (Page 1 of 2)

Device Designator	Device
DF:, DS:, DK:, DL:,DM:, DP:, DR:, DB:, or SY:	RSTS/E public disk structure as a whole.
SY*:	System disk (the unit which was bootstrapped).
DF*:	RF11 disk (all platters).
DS0: to DS7:	RH11/RS03/RS04 fixed head disk units 0 through 7.
DK0: to DK7:	RK11/RK05 disk cartridge units 0 through 7.
DL0: to DL3:	RL01/RL02 disk cartridge units 0 through 3.
DM0: to DM7:	RK611/RK06/RK07 disk cartridge units 0 through 7.
DP0: to DP7:	RP11/RP02/RP03 disk pack units 0 through 7.
DR0: to DR7:	RH11/RM02/RM03 disk units 0 through 7.
DB0: to DB7:	RH11/RP04/RP05/RP06 disk pack units 0 through 7.

Table MUP-2. RSTS/E Device Designators (Page 2 of 2)

Device Designator	Device
PR:	High speed paper tape reader.
PP:	High speed paper tape punch.
CR:	CR11 punched or CM11 mark sense card reader.
CD:	CD11 punched card reader.
MT0: to MT7:	TE10/TU10/TS03 magtape units 0 through 7.
MM0: to MM7:	TE16/TU16/TU45 magtape units 0 through 7.
MS0: to MS7:	TS04 magtape units 0 through 7
LP0: to LP7:	Line printer units 0 through 7.
DT0: to DT7:	TU56 DECtape units 0 through 7.
DD0: to DD7:	TU58 DECtape II units 0 through 7.

DEVTST Dialogue Errors

The following error messages may occur in DEVTST programs:

Message	Meaning
%Warning: (Device Name) is a logical device: (Device Name) will be used.	DEVTST determines that the device name given is a logical name for another device. DEVTST performs a logical device translation and the translated device name is used.
?Illegal device: (Device Name) specified.	The device name given by the user is either incorrect or is a logical device name that cannot be translated (for example, incorrect characters or the wrong type of device). Retry the operation with the correct device name.
?Open failure on (Device Name): (Followed by an error message).	This error message can mean one of five things: 1. The device specified does not exist. 2. No volume was mounted on that device. 3. Another user has ASSIGNed the device. 4. The volume on the device has not been formatted. 5. The device is not on-line or the device is not write-enabled. To resolve: retry the operation using the correct option as determined by the error message received.
?Illegal number of iterations selected.	Retry the operation using the correct number of iterations.

CPEXER

The central processor exerciser is designed to put a heavy load on the central processing unit. CPEXER runs compute bound for short bursts and then sleeps for five seconds so that average processor loading does not appreciably degrade the performance of the system. CPEXER was specifically designed to test the FPP option, but also verifies general CPU integrity and the PDP11/40 EIS operation. There is also a test of the PEEK function for kernel addresses 0 through 22000.

The tests performed by CPEXER are described in the next section. In most of the tests, results are compared to known correct values. Two heavily compute-bound tests are included to verify consistent results of duplicate calculations. Finally, tests are included that verify FPP divide by zero trap, integer divide by zero trap, integer conversion error and EIS operation.

CPEXER Tests

There are thirteen CPEXER tests included in the CPEXER program:

1. SIN(X) Uses SIN(X) extended function.
2. SIN(X) Uses polynomial approximation to SINE function.
3. LOG(X) Uses LOG(X) extended function.
4. EXP(X) Uses EXP(X) extended function.
5. SQR(X) Uses SQR(X) extended function.
6. SQR(X) Uses Newton-Raphson method to determine the square root.
7. LOG(EXP(X)) Compute-bound test.
8. ATN(TAN(X)) Compute-bound test.
9. A=1.0/0.0 Verify FPP divide by zero trap.
10. A%=1%/0% Verify integer divide by zero trap.
11. A%=60000 Verify integer conversion error.
12. INT(40.5621*100+0.5)/100=40.66 Verify EIS operation.
13. PEEK(X) Verify operation of PEEK function.

CPEXER Dialogue

The following is an example of the CPEXER dialogue:

```
RUN $1,2151 CPEXER
CPEXER  V7.0-1  RSTS V07.0-07 The ARK

How many minutes to run <15>? 1

Detach <No>? NO

11.4 seconds of CPU time used in 1 pass.

CPEXER finished at 11:01 AM

Ready
```

> **Note**
>
> The CPEXER program will build correctly only if the version of BASIC-PLUS or BASIC-PLUS-2 includes TRIG and LOG. These mathematical functions must be included during System Generation.

CPEXER Errors

This program reports four types of errors:

1. Instructions failed to execute
2. Errors failed to trap
3. Computation errors
4. Computation consistency checks failed

Check the System Error Log for reported errors. These errors may indicate hardware problems.

When type 1 errors (instruction errors) occur, CPEXER prints a two line message: the first line gives the test that was being performed and the exact error message and the second line indicates that the test is continuing. For example:

```
?CPEXER PEEK failed at memory location m.
-( Error Message)
Continuing.....
```

Where: (Error Message) is the RSTS/E error message.

m is the memory location at which the PEEK function failed.

Examples of type 2 errors (trap failure) are:

```
?CPEXER Floating point divide by 0 didn't trap
Continuing.....

?CPEXER Integer divide by 0 didn't trap
Continuing.....
```

Type 3 errors (computation errors) give the type of test (SIN, PLYNOM, LOG, EXP, SQR or NEWTON SQR), error in the computed value, and the location of the possible error. For example:

```
?CPEXER - SIN test failed for the following cases:
?CPEXER - Computed value of SIN( 40    ) was .643, should be .64
?CPEXER CPU,EIS,FIS,FPP,MUL,DIV or SHIFT  Error
Continuing....
```

Type 4 errors (consistency failure) in the CPEXER are illustrated by the examples shown below:

```
?CPEXER - EXP/LOG Consistency check failed for the following cases:
?CPEXER - EXP(LOG(...( 1    ))))))))))) <> EXP(LOG(...(1 ))))))))))
?CPEXER - EXP(LOG(...( 4    ))))))))))) <> EXP(LOG(...(4 ))))))))))
?CPEXER CPU,EIS,FIS,FPP,MUL,DIV or SHIFT Error
Continuing ....
```



```
?CPEXER - ATN/TAN Consistency check failed for the following cases:
?CPEXER - ATN(TAN(...(-1.5    ))))))))))) <> ATN(TAN(...(-1.5))))))))))
?CPEXER CPU,EIS,FIS,FPP,MUL,DIV, or SHIFT Error
Continuing....
```

CPUTST

The CPUTST dialogue first asks how many minutes you want the test to run. The program then checks for expiration of this request time at eight strategic points during each run sequence. CPUTST then clears and/or sets three 2-dimensional arrays to specified values. Matrix A is then transposed and the results placed in Matrix B. The transposition is checked by the statement A(J,I) = B(I,J) while I and J are being varied. Any error is then reported; if none are encountered, the program returns to the matrix clear instruction. The program repeats this process until the run time has expired, at which time it prints the CPU time in seconds of expended job time.

CPUTST Dialogue

The following is an example of CPUTST dialogue:

```
Run [1,215]CPUTST
CPUTST V7.0-1 RSTS V07.0 0-07 The ARK

How many minutes to run <15>?1

Detach <no>? NO

6.9 seconds of CPU time used

CPUTST Finished at 10:59 AM

Ready
```

CPUTST Errors

This device exerciser reports errors as shown in the following examples:

?CPUTST-Error in matrix transposition routine.

?(Error message) at line mm in CPUTST V7.0-07.

Where: (Error message) is the RSTS/E error message.

mm is the line number in CPUTST where the error occurred.

Check the System Error Log for reported errors. These errors may indicate hardware problems.

DSKEXR

This disk exerciser tests the normal operation of all file structured disks. It is possible to run several copies of DSKEXR in order to test several drives simultaneously or to put a heavier load on any single drive. The disk must be physically and logically mounted. DSKEXR begins by asking questions to determine the disk drive to be tested and the number of test iterations to be performed. After this dialogue, the exerciser opens and extends a temporary file to a predetermined size which is dependent upon disk size and the number of free blocks. A pattern buffer is then loaded with one of four patterns (all 1's, all 0's or a pattern of alternating 1's and 0's) and the file is written. Each block is then read and compared for each of the patterns. Upon completion of all iterations for each drive specified, a status report is printed.

> **Note**
>
> This write-operation does not use current disk files to read and write. The temporary file will always be deleted after use, regardless of how the DSKEXR program is terminated. Be sure to turn off data caching before you run this test.

DSKEXR Dialogue

The following is an example of DSKEXR dialogue:

```
RUN [1,215]DSKEXR
DSKEXR V7.0-1 RSTS V07.0-07 The ARK

Disk drive to test?  DB1:

Number of test iterations <8>? 1

Detach <No>? NO

No errors detected on _DB1:

Ready
```

DSKEXR Errors

Errors detected by DSKEXR are displayed as shown below:

?(Error message) error at line mm in DSKEXR V7.0-07.

?nn errors detected on DBn.

Where: (Error message) is the RSTS/E error message.

mm is the line number where the error occurred in the DSKEXR program.

nn is the number of errors detected by DSKEXR.

When errors are detected:

1. Insure that the volume is physically mounted and write-enabled.

2. Insure that the volume is logically mounted. This may be verified by the use of the SYSTAT program. If the disk is not mounted, use the mount command of the UTILTY program.

> **Caution**
>
> A. Packs mounted as unlocked public disks will be available to other system users for creation and storage of their files. Subsequent removal of a public disk may result in the elimination of these user files from the system. For this reason, DIGITAL recommends the use of a private pack if a non-system disk is to be tested.
>
> B. Never mount a potentially bad disk on another drive until you are sure that the drive and pack are free of physical damage.

3. Persistent errors may be due to bad blocks on the pack. Any potentially bad blocks will be flagged in the System Error Log following unsuccessful access. To prevent further access to defective areas of the pack, use the BADS suboption of the REFRESH option of INIT.

DSKSEK

This disk exerciser tests the normal operation of non-file structured disks by randomly reading blocks from the disk. It is possible to use several copies of DSKSEK in order to test several drives simultaneously or to put a heavier load on any single drive. If the disk being tested is a mounted RSTS/E file structured disk, DSKSEK will not read any known bad blocks. If an error occurs on a read operation, it will be reported by DSKSEK and logged into the System Error Log.

> **Note**
>
> This is a read only operation. Turn off data caching before you run this test.

DSKSEK Dialogue

The following is an example of DSKSEK dialogue:

```
RUN [1,215]DSKSEK
DSKSEK V70-1  RSTS V07.0-07  The ARK

Disk drive to test? DB0:

Number of test iterations <30>? 1

Detach <No>? NO

No bad block detections in 1000 gets from _DB0:

Ready
```

DSKSEK Errors

Errors detected by DSKSEK are displayed as follows:

?(Error message) error at line mm in DSKSEK V7.0-07.

Where: (Error message) is the RSTS/E error message.

mm is the line number where the error occurred in the DSKSEK program

When errors are detected:

1. Check to see if the volume on the testing device is physically mounted.
2. If it is, mount another scratch volume on the drive and retry operation.

> **Caution**
>
> Never mount a potentially bad disk on another drive until you are sure that the drive and pack are free of physical damage.

3. If the test continues to fail, check the System Error Log to see if bad blocks are recorded.
4. If bad blocks are recorded, add them to the bad block list on the disk using INIT'S REFRESH/BADS suboption. This can only be done if the disk has been initialized (DSKINTed) as a RSTS/E disk.
5. If the problems continue, get a complete listing of the System Error Log (using the ERRDIS program).

DTEXER

The DECtape exerciser tests the normal operation of the TC11 DECtape control and TU56 DECtape drives. DTEXER begins by opening a file on the drive being tested and fills the file with floating point numbers. The program writes 420 blocks (of a possible 578 blocks) on the tape. Numbers written on the tape are read and checked, keeping a count of incorrect values. If at the end of the test the error count is not zero, DTEXER prints the error count.

> **Caution**
>
> Any information previously contained on the DECtape used for this test will be destroyed.

DTEXER Dialogue

The following is an example of DTEXER dialogue:

```
RUN [1,215]DTEXER
DTEXER V7.0-1  RSTS V07.0-07 The ARK

Warning:  This exerciser will destroy data on the tested
          DECtape.

Dectape drive to test? DT0:

Number of iterations <2>? 1

Detach <No>? NO
No errors detected on _DT0:

DTEXER Finished at 11:00 AM

Ready
```

DTEXER Errors

This device exerciser shows the number of errors found during operation as shown in this example:

?mm errors on DTn during iteration nn.

Where: mm is the number of errors detected by the DTEXER program.

DTn is the name of device being tested.

nn is the number of the iteration during which the error occurred.

When errors are detected:

1. Check to see if the DECtape on the testing device is physically mounted and write-enabled.
2. If it is, mount another scratch DECtape on the drive and retry the operation.
3. Insure that the tape path and heads are free of oxide.

DXEXER

This exerciser tests the normal operation of the floppy controller and the specified floppy drives. DXEXER starts by opening a file 420 blocks long consisting of floating point numbers. These are written on the drive being tested, then read back and verified. The program then displays the number of words written and read, and the number of errors detected.

Caution
Any information previously contained on the floppy disk used for this test will be destroyed.

DXEXER Dialogue

The following is an example of DXEXER dialogue:

```
RUN [1,215]DXEXER
DXEXER  V7.0-1 RSTS V07.0-07 The ARK

Warning:  This exerciser will destroy data on the tested
          floppy disk.

Floppy drive to test? DX0:

Number of iterations <20>? 2

Detach <No>? NO

Floppy disk DXEXER I/O data

Device     Reads     Words     Writes     Words
_DX0:      300       76.8K     300        76.8K

DXEXER Finished at 10:53 AM
```

DXEXER Errors

DXEXER shows the number of errors detected during operation as shown below:

?mm errors on DXn during iteration nn.

Where: mm is the number of errors detected by the DXEXER program.

DXn is the name of device being tested.

nn is the number of the iteration during which the error occurred.

When errors are detected:

1. Check to see if the floppy disk on the testing device is physically mounted and the door closed.
2. If it is, mount another scratch floppy disk on the drive and retry the operation.

KBEXER

The keyboard exerciser (KBEXER) is used to test local and remote terminals connected to the RSTS/E system. The four terminal tests are as follows:

1. The Space Test verifies that the terminal carriage will return reliably from any position.
2. The ASCII Pattern Test verifies that the terminal will print the standard ASCII character set in all print positions.
3. The Worst Case Test forces the head on ASR33 terminals to rotate a half revolution in every print position.
4. The Repeat Test repeats everything typed, one line at a time.

> **Note**
>
> The tests can be terminated at any time by typing CTRL/C. HELP can be obtained by typing HELP in answer to any dialogue question.

KBEXER Dialogue

The following is an example of KBEXER dialogue:

```
RUN [1,215]KBEXER
KBEXER V7.0-1 RSTS V07.0-07 The ARK

Keyboard to test <_KB68:>?

Test (SPA[CE], ASC[II], WOR[ST], REP[EAT], HEL[P]) ? ASC

Number of test iterations <8>? 1
```



```
Detach <No>? NO

     *** ROTATING ASCII CHARACTERS TEST ***

#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
[/]^_!"#$%&'
                    .
                    .
                    .
KBEXER Finished at 11:44 AM

Ready
```

KBEXER Errors

Errors detected by KBEXER appear as errors in the keyboard test patterns. Visually check the patterns for errors.

LPEXER

This exerciser prints a rotating pattern of ASCII characters on a specified number of pages and is used to test any line printer.

LPEXER Dialogue

The following is an example of LPEXER dialogue:

```
RUN [1,215]LPEXER
LPEXER V7.0-1 RSTS V07.0-07 The ARK

Line printer to test? LP0:

Number of pages to output <4>? 1

Detach <No>? No

LPEXER finished at 11:39 AM

Ready
```

LPEXER Errors

Problems found on the line printer appear as errors in the print-out of ASCII characters. Visually check the line printer listing and if an error message is printed by LPEXER, check to make sure that the line printer selected is on-line and not out of paper.

MTEXER

The magtape exerciser checks normal operation of the TM11 magtape control or RH11/TM02/TM03 controller and the specified number of 7-track or 9-track TE10, TE16, TU10, TS03, TU16, TU45 or TS04 drives. MTEXER allows you to select the drive to be tested, the length of tape to be written and the number of iterations to be performed. For each iteration, MTEXER zeroes the tape, opens a file and writes data until the specified length of tape has been used. MTEXER then rewinds the tape, opens the file for input, reads and verifies the data. If errors are detected, MTEXER prints a count of the number of bytes found to be incorrect; MTEXER then proceeds to the next iteration.

The data pattern used is a worst case NRZI pattern for 9-track drives; it is not worst case for 7-track recording. The pattern is loaded into a 512 byte buffer and, successively greater numbers of records are written to the magtape until the requested length of tape has been written. Furthermore, the pattern base varies with X so that the contents of the pattern buffer also varies on successive iterations.

> **Caution**
>
> This exerciser will overwrite data contained on the magtape used for the test.

MTEXER Dialogue

The following is an example of MTEXER dialogue:

```
RUN [1,215]MTEXER
MTEXER V7.0-1 RSTS V07.0-07 The ARK

Magtape drive to test? MT0:

Density (800 or 1600) <800>? 1600

Number of feet to test <100>? 25

Number of iterations <2>? 1

Detach <No>? NO

Magtape MTEXER I/O Data

Device      Reads       Words      Writes      Words
 MT0:
96          24,576 K    96         24,576 K

No errors detected on _MT0:

Ready
```

MTEXER Errors

Magtape errors are reported in these formats:

```
?Magtape select error at line nn in MTEXER V7.0-07.

?MTEXER-Selection error occurred.
```

Where: nn is the line number in MTEXER at which the error occurred.

When errors are detected:

1. Check to see if the volume on the testing device is physically mounted, write-enabled and assigned to you.

2. If it is, mount another scratch volume on the drive and retry the operation.

3. Insure that the tape path and heads are free of oxide.

PPEXER

This program tests the normal operation of the paper tape punch by punching random patterns on the paper tape for input into the PREXER.

PPEXER Dialogue

The following is an example of PPEXER dialogue:

```
RUN [1,215]PPEXER
PPEXER V7.0-1  RSTS V07.0-07 The ARK

This test will punch a binary count pattern tape. The tape
will be equivalent to MAINDEC-00-D2G4 entitled S p e c i a l
Binary Count Pattern Tape.

Detach <No>?

The generated paper tape may be used to test the paper tape
reader

PPEXER finished at 11:23 AM

Ready
```

PPEXER Errors

Problems in the paper tape punch will result in an incorrectly generated paper tape which is detected by PREXER when the tape is read. If PPEXER prints an error message, check to see that the paper tape punch is on-line and not out of paper tape.

PREXER

This program tests the normal operation of the paper tape reader by reading the paper tape punched by the PPEXER program.

PREXER Dialogue

The following is an example of PREXER dialogue:

```
RUN [1,215]PREXER
PREXER V7.0-1 RSTS V07.0-07 The ARK

Load MAINDEC-00-D2G4-PT or the binary count pattern
tape generated by the punch exerciser into the reader with
the leader under the read station.

Detach <No>? NO
No errors detected on paper tape.

PREXER finished at 11:29 AM

Ready
```

PREXER Errors

PREXER errors are reported in this format:

```
?nn errors detected on paper tape.
```

Where: nn is the number of errors detected by the PREXER program.

> **Note**
>
> If the paper tape was loaded backwards and the reader is operating correctly, the following error message is printed: ?15360 errors detected on paper tape.

Any other errors detected by the PREXER may indicate a hardware problem.

ON-LINE CLEAN PROGRAM - ONLCLN

1. Provides the same 'CLEAN' as the REFRESH option of INIT.
2. Executes under the RT11 Run-Time System.

   Example dialogue:

   ```
   RUN $ONLCLN

   Disk? DM0:

   Disk is being cleaned - wait...

   Disk? ^C

   Ready
   ```

> **Note**
>
> Disk must be physically, but not logically mounted.

ENHANCED MANAGEMENT UTILITY PROGRAMS

Spooling Package

1. Programs and their related files can reside in an account other than [1,2].

   - OPSER Program

     Reports the keyboard number from which a message to the operator has been sent

     For example:

```
MESSAGE          65 : 07-Mar-79 09:46 AM  JOB:20  KB68:  PLEASE[100,73
     PLEASE MOUNT MAGTAPE LABELLED 'TESTS' ON MM1:
```

Figure MUP-1. OPSER Message

   - QUEMAN Program

     New interrupt commands

Table MUP-3. New Interrupt Commands

Command	Option	Meaning
DIS[ABLE]	QUE[UING]<text>	QUE stops sending messages to QUEMAN and stops listing the queue.
	SPO[OLING] <text>	QUEMAN stops sending job requests to all spoolers.
	All <text>	Both QUEUING and SPOOLING are stopped.
ENA[BLE]	QUE[UING] <text>	QUE resumes sending messages to QUEMAN and enables listing the queue.
	SPO[OLING] <text>	QUEMAN resumes sending job requests to the spoolers.
	All <text>	Both QUEUING and SPOOLING are enabled.

- QUE program

  New commands

  F FLUSH A QUEUE*
  S SHORT QUEUE LIST

  CCL command

  Recognizes the CCL command SUBMIT <filename> and spools the file for batch processing.

* Privileged operation

SYSTAT Program

```
RUN $SYSTAT
SYSTAT  X7.0-05 RSTS V7.0-04 Timesharing
Output Status to?

RSTS V7.0-04 Timesharing status at 14-Nov-78, 03:00 PM Up: 31:00:37

Job   Who     Where    What    Size    State    Run-Time   Pri/LR   RTS
 1    [OPR]   Det      EPRCPT   5/31K  SR D60     0:23.8     -/5    BAS4F
 2    [OPR]   Det      OPSRUN  15/31K  SL         1:44.4    -0/4    BASIC2
 3    [OPR]   Det      QUMRUN  16/31K  SL D40     0:22.9    -0/4    BAS4F
 4     1,247  KB66     DSKDMP   8/31K  RN          5J.3   -24/2    BAS4F
 5    [OPR]   Det      BATRUN  15/31K  SL         28.7      -0/4    BAS4F
 6    [OPR]   Det      BATIOL  13/31K  SL D61      0.7      -0/3    BAS4F
 7    [OPR]   Det      BATIOL  13/31K  SL D59      4.3      -0/3    BAS4F
 8    [OPR]   Det      MCU     14/31K  SR D58      9.9      -0/3    BAS4F
 9    [OPR]   Det      MPKOVR   8/31K  SR         24.3      -0/2    BAS4F
10    [OPR]   Det      DTM.2R   2/31K  SL        17:21.7    120/10   ...DCL
11     1,224  KB44     SYSTAT  10/31K  *C         3:26.8     0/3     BAS4F
12    [OPR]   Det      SPLIOL  16/31K  SL D62      0.1      -0/4    BAS4F
13     2,215  KB10     VTEDIT  17/31K  KB         7:57.6     0/4     TECO
14     2,201  KB43     NONAME   1/31K  *C          16.9      0/1     DF2COM
15     1,250  KB33     ...DP2  20/31K  *C A16     33:22.1     0/6     DF2COM
16     1,200  KB22     NONAME   2/31K  *C A13      45.6      0/1     BAS4F
17     1,240  KB24     SYSTAT  10/31K  *C         2:20.5     0/2     BAS4F
18     1,231  Det      STANEW   4/31K  SL          13.7      -0/3    BAS4F
19     1,219  KB35     QUE     15/31K  *C A02       5.0      0/4     BAS4F
20     1,242  KB31     VTEDIT  20/31K  KB A08     2:09.0     0/5     TECO
21     1,222  KB25     NONAME   2/31K  *C A06       18.4     0/3     BAS4F
22 D   2,245  KB23     NETWRK   8/31K  SL         5:17.0     -0/2    BAS4F
23     2,220  KB20     VTEDIT  17/31K  KB A07     1:05.0     0/4     TECO
24     1,243  KB40     ...TKB  14/31K  KB A03     2:11.9     0/3     ...RSX
25     2,237  KB47     VTEDIT  17/31K  KB A11       7.0      0/4     TECO
26     1,225  KB30     VTEDIT  20/31K  KB         7:51.3     0/5     TECO
27     1,222  KB26     TTYSET  10/31K  *C A04     6:10.7     0/3     BAS4F
28     1,247  KB46     DSKDMP  13/31K  *C A05    10:54.2     0/3     BAS4F
29     1,237  KB17     EVLNFS   4/31K  *C        1:50.6     0/3     BAS4F
30     2,210  KB27     NONAME   2/31K  *C A01      31.0     0/1     BAS4F
31     2,230  KB32     NONAME   2/31K  *C A12      23.9     0/1     BAS4F
32     2,236  KB21     NONAME   2/31K  *C A15       0.9     -0/6    BAS4F
33     1,248  PB35     SYMBLD   6/31K  RN         5:52.9    -40/2    BAS4F
34     1,245  KB34     SYSTAT  10/31K  *C A10      29.5     0/3     BAS4F
35     1,244  KB26     SYSTAT  10/31K  *C         15.1      0/3     BAS4F
36    [SELF]  KB7.D    SYSTAT  10/31K  RN Lck      7.3      -0/3    BAS4F
37     1,253  PJJ9     NFT.V2  16/31K  RN          2.1      -7/4    RMSV15

Busy Devices:
Device  Job     Why
PK0      5      AS*INIT
PK1      9      INIT
XM0     NSP     AS*INIT
XM1     NSP     AS*INIT
XM2     NSP     AS*INIT
MM0      35     AS*ANSI
MM2      35     AS*ANSI

Disk Structures:
Disk    Open    Free     Cluster Errors  Name    Comments
DS0      13     608        8      9      SWAP0   Pri, DLW
DS1       1                1      0             Pri, NFS
DB4       1                1      0             Pub, Priv, NFS
CB0       6     5120       8      0      VT6C    Pri, Priv, DLW
DB1      37     3068       8      0      ALB     Pub, DLW
LB2       3     152096     8      0      VP      Pri, DLW
```

Figure MUP-2. SYSTAT (page 1 of 2)

```
Small   Large   Jobs     Hung TTY's      Errors
 41       1    37/63         0            388

Run-Time Systems:
 Name    Ext       Size   Users   Comments
BAS4F    BAC     16(16)K   26     Perm, Addr:46, KBM, CSZ
HISTOG           28(4)K    0     Perm, Addr:268, KBM, CSZ, EMT:255
BASIC    BAC     16(16)K   0     Non-Res, KBM, CSZ
TECO     TEC      8(24)K   5     Temp, Addr:296, KBM
RT11     SAV      4(28)K   1     Temp, Addr:216, KBM, CSZ, EMT:255
RSX      TSK      3(28)K   0     Non-Res, KBM
RMS11    TSK      4(28)K   0     Non-Res
COBOL    TSK      4(28)K   0     Non-Res
BASIC2   TSK     16(16)K   1     Temp, Addr:163
BP2COM   TSK      4(28)K   2     Temp, Addr:179, KBM
BASICY   TSK     16(16)K   0     Non-Res
BPYCOM   TSK      4(28)K   0     Non-Res, KBM
RMSV15   TSK      4(28)K   1     Temp, Addr:184

Resident Libraries:
 Name     Prot      Acct      Size    Users   Comments
RMSRES   < 42>   [  0,1  ]    23K      0     Non-Res, Addr:66

Message Receivers:
 Name    Job    Msgs    Max     Senders
ERRLOG    1       0      40     Priv
OPSER     2       0      30     Local
QUEMAN    3       0      60     Local
BA0SPL    5       0       5     Priv
BA1SPL    6       0       5     Priv
BA2SPL    7       0       5     Priv
NCU       8       0       5     Priv, Network
NWPK09    9       0       5     Network
LP0SPL   12       0       5     Priv
NWTT22   22       0       5     Network
NFT.37   37       0       5

Ready
```

Figure MUP-2. SYSTAT (page 2 of 2)

- Three new privileged switches

  /O[:DEV] - list open files (large file systems only)
  /W[:DEV] - list open files and jobs accessing them (large file systems only)
  /C - list memory allocation

Table MUP-4. Status Codes

Code	Meaning
Pla	File is placed
Upd	File is opened in update mode
Nok	File cannot be renamed or deleted
Ufd	File is a UFD
Ctg	File is contiguous
Mdl	File is marked as deletable

Table MUP-5. Job Access Codes

Code	Meaning
Rd	Job has read access
Rr	File opened read regardless
Wr	Job has write access
Tent	File is a tentative file
Ca	File opened with data caching
Sq	File opened with sequential data caching
Up	File opened in update mode
Spup	File opened in special update mode

```
Open Files:

DS0: -- Null list

DM1: -- None

DB0: -- None

DB1: -- File                    Op/RR    Size    Clu  Status
DB1:[  1,235]SYSO   .LST<60>    1/0       0       8
DB1:[  1,247]DISKID.BAS<60>     1/0      20       8
DB1:[  2,201]BUFMAP.B2S<60>     1/0       2       8
DB1:[  1,250]                   1/0      48      16   NoK, UFD
DB1:[  2,201]TEMP14.TMP<60>     1/0      22       8
DB1:[  2,201]TMP214.TMP<60>     1/0     128       8
DB1:[150,34 ]WRK    .TMP<40>    1/0       0       8
DB1:[150,34 ]CREF   .TMP<40>    1/0      25       8
DB1:[  1,2  ]MACRO  .SAV<104>   0/1      57       8
DB1:[  1,10 ]BATC06.WRK<60>     1/0      13       8
DB1:[  1,10 ]SPL004.WRK<60>     1/0       9       8
DB1:[  1,10 ]CHARS  .QUE<60>    1/0       2       8
DB1:[  1,10 ]BATC05.WRK<60>     1/0      13       8
DB1:[  1,10 ]BATC07.WRK<60>     1/0      11       8
DB1:[  1,10 ]OPSLG1.LOG<60>     1/0       6       8
DB1:[  1,2  ]OPSER0.WRK<60>     1/0       2       8
DB1:[  1,2  ]OPSER1.WRK<40>     3/0      21       8
DB1:[  1,2  ]QUEUE  .WRK<60>    1/0      23       8
DB1:[  1,2  ]QUEUE  .SYS<40>    2/0      48       8
DB1:[  1,2  ]TEMP03.QUE<60>     1/0     105       8

DB2: -- File                    Op/RR    Size    Clu  Status
DB2:[  1,2  ]BASIC2.TSK<104>    0/1     465       8  Ctg
```

Figure MUP-3. SYSTAT /O


```
Open Files and jobs accessing them:

DS0: -- Null list

DM1: -- None

DB0: -- None

DB1: -- File                    Op/RR    Size    Clu  Status
DB1:[150,34 ]CREF   .TMP<40>    1/0       0       8
                  17   Rd, Wr, Tent

DB1:[  1,2  ]MACRO  .SAV<104>   0/1      57       8
                  17   Rd, RR

DB1:[  2,201]EVTRQS.B2S<60>     1/0      16       8
                  14   Rd

DB1:[  1,10 ]BATC06.WRK<60>     1/0      13       8
                   6   Rd, Wr

DB1:[  1,10 ]SPL004.WRK<60>     1/0       9       8
                   4   Rd, Wr

DB1:[  1,10 ]CHARS  .QUE<60>    1/0       2       8
                   4   Rd

DB1:[  1,10 ]BATC05.WRK<60>     1/0      13       8
                   5   Rd, Wr

DB1:[  1,10 ]BATC07.WRK<60>     1/0      11       8
                   7   Rd, Wr

DB1:[  1,10 ]OPSLG1.LOG<60>     1/0       6       8
                   2   Rd, Wr

DB1:[  1,2  ]OPSER0.WRK<60>     1/0       2       8
                   2   Rd, Wr

DB1:[  1,2  ]OPSER1.WRK<40>     4/0      21       8
                  19   Rd
                   3   Rd
                  26   Rd
                   2   Rd, Wr

DB1:[  1,2  ]QUEUE  .WRK<60>    1/0      23       8
                   3   Rd, Wr

DB1:[  1,2  ]QUEUE  .SYS<40>    3/0      48       8
                  19   Rd
                  26   Rd
                   3   Rd, Wr


DB2: -- File                    Op/RR    Size    Clu  Status
DB2:[150,62 ]WRITE  .MAC<40>    1/0       5       8
                  12   Rd
```

Figure MUP-4. SYSTAT /W

```
SY/C

Memory allocation table:
 Start    End   Length   Permanent    Temporary
   0K -    46K (  47K) MONITOR
  47K -    62K (  16K) BAS4F  RTS
  63K -   142K (  80K) ** XBUF **
 143K -   150K (   8K) TECO   RTS
 151K -   178K (  28K) HISTOG RTS
 179K -   179K (   1K)   (User)
 180K -   202K (  23K)   (User)     RMSRES LIB
 203K -   204K (   2K)   (User)     COBOL  RTS
 205K -   206K (   2K)   (User)     COBOL  RTS, FOO    RTS
 207K -   207K (   1K)   (User)     FOO    RTS
 208K -   511K ( 304K)   (User)
 512K -  *** END ***

Ready
```

Figure MUP-4A. SYSTAT/C

VT50PY (VT5DPY) Program

1. New command for resident libraries

Command	Meaning
H	Displays resident library data.
-H	H preceded by a minus sign removes the data from the screen.

2. Memory display for resident libraries

   -n- where n is the library's position in the resident library list.

TTYSET Program

1. Macros for new terminals include:

   LA34
   LA38
   LA120
   VT100

Error Logging Package

1. Recognizes and reports errors for new devices.
2. Programs and their related files can reside in an account other than [1,2].
3. Always reports the block number of the disk block that caused an error.

```
**********************************************************************
DR  RH11/RM02-03 Seq #16 Occurred on 17-Nov-78 at 08:12:48 AM

User Description:
-----------------
Job Number                16
KB Number                 33
Account                   [1,250]
Program Name              PACKID
User Job Physical Addr.   01100000
User Job Size             4K
Control Parameters        200
RTS Name                  BAS4F
RTS Physical Address      00270000

Detailed Description:
---------------------
I/O Status                000
Timeout Indicator         000
Offset Position           Previous
Overlapped Seek Ind.      000
Unit Size in DC's         100220
Device Cluster Size       4
Pack Cluster Size         4
Logical Name
Physical Name             DR1:
Logical Block Number      4

DSQ                       000400  174044  072344  000007
                          000001  002171  106266  176000
                          000000  013700  000000  000034
                          000000  002002  127600  000400

*CB                       002400  102044  000001  000005
                          013734  000000  000000  000000
                          000000  000000  000000  000000
                          001007  001975  000004  000000
```

Figure MUP-5. ERRDIS (page 1 of 3)


```
CSR Address               176300

Contents of Registers
RMCS1       160200        MCPE Bus Err TRE Xfer Err
RMWC        000000
RMBA        000000
RMDA        000000
RMCS2       010101
RMDS        000000
RMER1       000000
RMAS        000000
RMLA        000000
RMMR1       000000
RMD1        000000
RMSN        000000
RMOF        000000
RMDC        000000
RMHR        000000
RMMR2       000000
RMER2       000000
RMEC1       000000
RMEC2       000000
RMBAE       000000
RMCS3       000000
```

Figure MUP-5. ERRDIS (page 2 of 3)

```
**********************************************************************
XM  DMC11 NPR Inter. Seq #613 Occurred on 16-Nov-78 at 02:59:59 PM

Detailed Description:
---------------------
Timeout Indicator       000000
Physical Name           XM1:

DDB                     000010  000403  000000  040001
                        102400  001035  000000  027640
                        023006  000000  000000  000000
                        073532  000004  027540  000200
                        000000  000000  000000  000400
                        000000  000000  000000  000000

DDB Extension           000000  027640  014200  021040
                        000000  027650  000000  027654
                        000000  027660  000000  027664
                        000000  000000  002004  000000

CSR Address             160160

Contents of Registers
SEL0      100000
SEL2      046701
SEL4      000000
SEL6      000200
```

Figure MUP-5. ERRDIS (page 3 of 3)

Table MUP-6. ERRDIS Mnemonics

Mnemonic	Description
UI	Undefined Interrupt
DR	RH11/RM02-03
DX	RX11/RX211
DD	TU58
TS	TS11/TS04
UM	Unexpected message*

* A message not received from the monitor.

UTILTY Program

1. New Commands include:

Command	Meaning
SEIZE DEV:	Assign "DEV:" even if it is currently assigned to another job, unless a file is open on the device.
ADD LIBRARY LIBNAME	Add "LIBNAME" as a Resident Library.
/ADDR:n	Load Library in memory at address n, which is a 1K section of memory.
/STAY	The Library should be permanently resident.
/1USER	The Resident Library is to handle only 1 user, rather than be shared code.
/RW	The Library should be mapped Read/Write, rather than Read-Only.
/NOLOGERR	Errors occurring within the Library should not be logged to the system error log.
/REMOVE	The Library should be immediately removed from memory when all jobs using it detach from the library.

> **Note**
>
> The /ADDR: switch must be specified with the ADD LIBRARY command.

REMOVE LIBRARY LIBNAME	Remove "LIBNAME" as a Resident Library.

LOAD LIBRARY LIBNAME	Load Resident Library "LIBNAME" in memory.
/ADDR:n	Load Library in memory at address n which is a 1K section of memory.
/STAY	The library should be permanently resident.
UNLOAD LIBRARY LIBNAME	Unload Resident Library "LIBNAME" from memory.
LIST CACHE	List current cache settings.
ENABLE CACHE	Enable directory caching.
/ALL	Cache all data transfers regardless of OPEN mode.
/FILE	Enable data caching based on file and OPEN mode.
/NOFILE	Disable data caching.
/BUFF	Small buffers may be used for caching.
/NOBUFF	Small buffers may not be used for caching.
/CL:n	Cache Cluster Size
/LIMIT:n	Total # of Cache Clusters which will be used.
/DIR:n	Total # of Cache Clusters which can be used for directories.
/DATA:n	Total # of Cache Clusters which can be used for data.
DISABLE CACHE	Disable all caching.
RUNBURST n Q	Change job n's runburst to Q.
SIZE n Q	Change job n's maximum size to Q
FLAG filename	Modify file's directory based on the following switches.
/CACHE	Flag file to be cached automatically when the file is opened.

/NOCACHE	Do not automatically cache the file when it is opened.
/SEQ	Set file's default caching mode to sequential.
/RAN	Set file's default caching mode to random.
/PLACE	Flag file to be a placed file.
/NOPLACE	Flag file as not being placed.
/NOCTG	Flag file as not being contiguous.

ANALYS Program

1. New switch on output file produces an annotated version of a memory dump.
2. Additional information in the dump file.
3. New device support
4. Resident library support

```
RUN [1,10]ANALYS
ANALYS X7.0-05 RSTS V7.0-04 Timesharing
Input <[0,1]CRASH.SYS>?
Output <KB:ANALYS.DMP>?ANALYS.DMP/WIDE
Crash Error Log Filename<[1,10]ERRCRS.FIL>?

Ready
```

```
Analysis of [0,1]CPASM.SIS/SIL:SYS:[0,1]RSTAB1.SIL
        Taken on 22-Nov-78 at 09:30 AM

System was configured with DECnet/E

System was configured for large files

Crash Dump Status From 16-Nov-78 at 11:32 AM, Upr 38:07:11


Job   Hpc    Where    What    Size     State     Run-Time   Pri/RB   RTS
 1    1,2    Det      EPPCP3   5/31K   RN D56      0:56.8     0/6   BASMP
 2    1,10   Det      CPSPSH  15/31K   SL          2:29.2    -8/4   BASIC2
 3    1,10   Det      CLMPLR  16/31K   SL         10:59.9    -8/4   BASMP
 4    1,10   Det      SPL:DL  16/31K   SL D57       0.1      -8/4   BASMP
 5    1,1C   Det      BA1ELL  13/31K   SL           0.1      -8/3   BASMP
 6    1,1C   Det      BA1ELL  13/31K   SL           0.1      -8/3   BASMP
 7    1,10   Det      BA1PUN  15/31K   SL           9.4      -8/4   BASMP
 8    1,10   Det      NCU     14/31K   SR D59      11.5      -8/3   BASMP
 9    1,10   Det      APEC:P   8/31K   SR          12.5      -8/2   BASMP
10    1,10   Det      DIM.IN   2/31K   SL       19:49.0    120/10   ...RSX
11    1,2#8  KB2#     NCNAME   2/31K   ^C          2:07.0     0/1   BASMP
13    1,224  PCJ7     B2CPEF  25/31K   DB Lck      2:01.8   -40/5   ...RSX
22    2,217  KB#7     YIEC:1  17/31K   KB A10      13.6       0/4   TECO
23    2,225  KB20     YIEC21  16/31K   KB A06      8:02.6     0/4   TECO
24    1,2#6  KB30     RT11     2/31K   ^C          42.5       0/1   RT11
32    1,242  KB31     NCNAME   2/31K   ^C A10      28.8       0/1   BASMP

Busy Devices:
Device  Job     Why
FPC     F       AS=IM1T
DM1     RSP     AS=IN1T
DM2     RSP     AS=IM3T

Disk Structures:
Disk    Open    Free     Cluster  Errors  Name     Comments
DSC:     15       0         8       0     SWAP0    Pri, DLW
DS1:      1                  1                      Pri, NFS
DM0:      0    17#1#         1       0     BAS1MM   Pri, R-O, DLW
DP2:      0     #618         4       0     VOT8     Pri, DLW
DB5:      0    5120          8       0     V06C     Pri, R-O, DLW
DS1:     23    12#2#         8       0     ARK      Pub, DLW
DB4:      4    15#8#0        8       0     VOT      Pri, DLW

User Files:

DSC: -- 9.11 List

DM0: -- None

DP0: -- None

DB5: -- None

DB1: -- File
```

Figure MUP-6. ANALYS (page 1 of 6)


```
                                  Cp/Rb   Size   Clu     FRE              NE          Status
[ 1,224]B2CREF.1HP<#0>            1/0      77     8  0022524#20540 0022524#20820
      12   Rc, Wr

[ 1,2  ]B2CREF.1SR<124>           0/1     110     8  0002C0#1#44C 0002C0#1#42C  C1#
      15   Rd, RA

[ 1,10 ]BA2J7  .COM<#0>           1/0       5    10  0005#111#640 0005#111#740
       7   Rd, Wr

[ 1,224]B1QCSP.LUG<60>            1/0       0     6  7777777778776 0022524#2#200
       7   Rd, Wr

[ 1,10 ]BA1C07.%RE<60>            1/0      13     8  0005#110#720 0005#111#100
       7   Rd, Wr

[ 1,10 ]BA1C05.%RR<60>            1/0      13     8  0005#106#200 0005#106#160
       5   Rd, Wr

[ 1,10 ]BA1C06.%RE<60>            1/0      13     8  0005#110#640 0005#103#020
       6   Rd, Wr

[ 1,10 ]CHARS  .OVE<60>           1/0       2     6  0005#102#400 0005#102#360
       4   Rd

[ 1,10 ]SPL004.%RE<60>            1/0       9     8  0005#10:#040 0005#111#740
       6   Rd, Wr

[ 1,10 ]LPSL01.LOG<60>            1/0       8     8  0005#11:#640 0005#11:#620
       2   Rd, Wr

[ 2,229]TECO23.1HP<60>            1/0       0     6  7777777778776 0016:#21#240
      29   Rd, Wr, Ident

[ 2,229]FILE   .HAN<60>           1/0       1     8  0016:#21#120 0016:#21#5C0
      25   Rd

[ 2,217]COALS  .DCC<60>           1/0       0     8  7777777778776 011740218240
      22   Rd, Wr, Ident

[ 1,2  ]MCUBUF.STS<60>            1/0       9     8  000600118140 000200558720
       8   Rd, Wr

[ 1,2  ]QUEUE  .%RE<60>           1/0      23     8  000600128240 000200518120
       3   Rd, Wr

[ 1,2  ]QUEUE  .SIS<40>           1/0      48     8  000600128260 0002[illegible]#160
       5   Rd, Wr

[ 1,2  ]TEMP01.QUE<60>            1/0     108     8  000600168340 0C0160058660
       5   Rd, Wr

[ 1,2  ]QPSER1.%RR<#0>            2/0      21     8  0006C0128220 000200578C60
       3   Rd
       2   Rd, Wr

[ 1,2  ]QPSER0.%RR<60>            1/0       3     8  000600128200 000200578020
       2   Rd, Wr


DB2: -- File                      Cp/Rb   Size   Clu     FRE              NE          Status
[151,52 ]BA1PUN.LST<60>           1/0     168     8  00043#0119256 00043#119216
      15   Rd, Wr

[150,52 ]BA1ELL.BAS<#2>           1/0     134     6  0022220[illegible]C 0022220[illegible]220
      13   Rd
```

Figure MUP-6. ANALYS (page 2 of 6)

```
Fault   Large   Jobs    Hung Jobs       Errors
 229      1      16 0;       0             227

Run-Time Systems:
 Name   Ref      Size     Users   Comments
 BASIC  BAC     16k'0:K   10     Perm, Addr:48, KBM, CSI
 FILESC         2k:0:K    0     Perm, Addr:268, KBM, CSI, EM1:255
 BASIC  BAC     16k'0:K   0     Non-Res, KBM, CSI
 DECX   DEC     8k'2:K    2     Perm, Addr:296, KBM
 MACRO  SAV     4k'2:K    1     Temp, Addr:357, KBM, CSI, EM1:255
 PIP    ISK     3k'0:K    0     Temp, Addr:207, KBM
 SYSFCB ISK     4k'E:K    0     Non-Res
 CCPCL  ISK     4k'0:K    0     Non-Res
 BASIC2 ISK     16k'0:K   1     Temp, Addr:210
 BPICOM ISK     4k'2:K    0     Non-Res, KBM
 BASICT ISK     16k'6:K   0     Non-Res
 BPICOM ISK     4k'2:K    0     Non-Res, KBM
 BPIFYS ISK     4k'6:K    0     Non-Res

Resident Libraries:
 Name    Prot     Acct      Size    Users   Comments
 FMS8S  < R2>   [ 0,1 ]    23K      0     Non-Res, Addr:1k

No Links were in use

Nodes:
 Name    Addr    Line    Links/Max  Seg  State  Comments
 RSX       0     None       0/32    541  Run    Adj, Ans
 NCAN      6     XM2:       0/32    541  Run    Adj, Crg, Rct
 -224     55     XM1:       0/32    542  Run    Adj, Org, Rct

Lines:
 Line   Node   Size  Quota  Count  Mode      State  DCR  IMG  OVR  Job
 XM1:   0320   541     4      4    FDX       On     0    0    0   MSP
 XM2:   NCAN   541     8      8    FDX       On     0    0    0   MSP

Message Receivers:
 Device  Job   Obj    Msgs/Max    Lines/Max    Access
 $$$LOG   1     0      1/40         0/0       Prv
 OPSER    2     0      0/30         0/0       Lel
 OLSPAN   3     0      0/80         0/0       Lel
 LP:SPL   4     0      0/5          0/0       Prv
 BA1SPL   5     0      0/5          0/0       Prv
 BA7SPL   6     0      0/5          0/0       Prv
 BA2SPL   7     0      0/5          0/0       Prv
 NCB      8    19      0/5          0/6       Net, Prv
 $$PAC9   9   249      0/5          0/1       Net
 MSP      MSP   0      0/0          0/32      None

Octal Dump of Status

ERROR CODE
000176/ 177777

SAVED R0 TO R5
000202/ 000017  075776  000005  176000  120106  000002

KERNEL SP
000214/ 002004

VIRTUAL PROGRAM COUNTER
000216/ 016000
```

Figure MUP-6. ANALYS (page 3 of 6)


```
PROCESSOR STATUS
004344/ 004040

NEXT 8. ITEMS ON KERNEL SP STACK
004352/ 000000  000000  000000  000000  000000  000000  000000  000000

USER SP
004732/ 145004

TOP 8. ITEMS ON USER SP STACK
004750/ 000043  000043  000043  000043  000043  000043  000043  000043

USER KEYWORD
004710/ 103200

USER FIRQB
004706/ 000000  003032  000062  000000  010603  070516  077430  000000
004666/ 000000  000000  000000  000000  000000  000000  000000  000000

USER JSB
004646/ 004434  000000  050024  000036  000115  000000  000000

                        USER              USER              KERNEL            KERNEL
                        ADDR.             DESC.             ADDR.             DESC.
                        REG.              REG.              REG.              REG.
004532/  APR0           016740            077406            000000            077506
004542/  APR1           017140            077406            000200            077506
004552/  APR2           017340            077406            000400            077506
004562/  APR3           017540            077406            000500            077506
004572/  APR4           017740            077406            001000            077402
004602/  APR5           020140            057406            001710            077402
004612/  APR6           000000            000000            017440            077006
004622/  APR7           000000            000000            177600            077506

         JCB     NEXT
001006\  000     000

         JOBDA   JOBF    IOSIS   JOBWRK  JOBJD2  JOBRIS  CPUTIM  JOBWDB
001010/  000000  105052  062406  000000  062406  000000  062406  000000

F1JOB
001302\  032

F1JBDA
001306/  020040

JOB1BL
043362/ 000000  002700  010140  011040  011600  012240  012240  014600
043402/ 016640  015140  016300  017240  000000  020040  000000  000000
043422/ 000000  000000  000000  000000  000000  000000  022700  026200
043442/ 034300  000000  000000  000000  000000  000000  000000  000000
043462/ 027500  000000  000000  000000  000000  000000  000000  000000
043502/ 000000  000000  000000  000000  000000  000000  000000  000000
043522/ 000000  000000  000000  000000  000000  000000  000000  000000
043542/ 000000  000000  000000  000000  000000  000000  000000  000000
```

Figure MUP-6. ANALYS (page 4 of 6)

[redacted]

		Load	Size	Transfer	Total
PDCS	[redacted]	[redacted]	140000	000001	24K
IER	[redacted]	[redacted]	021000	000001	25K
IER	[redacted]	[redacted]	010000	000001	31K
BMS	[redacted]	[redacted]	017000	000001	34K
PIP	[redacted]	[redacted]	031000	000001	40K
PIP	[redacted]	[redacted]	007000	000001	41K
RSP	[redacted]	[redacted]	021000	000001	46K
.SY	[redacted]	[redacted]	031000	000001	51K
SYS	[redacted]	[redacted]	012520	142522	52K
DEFALT	[redacted]	[redacted]	000000	000001	54K

Contents of patch space [redacted]

```
[redacted]
```

Core Dump of Monitor

```
[redacted]
```

Figure MUP-6. ANALYS (page 5 of 6)


```
[redacted]
```

Figure MUP-6. ANALYS (page 6 of 6)

SHUTUP Program

1. Additional information in printout

```
RUN $SHUTUP
SHUTUP   X07.0-1     RSTS V07.0-03 TIME SHARING

12:24 PM 22-Sep-78 ###### Set-up Dialogue Phase #######

Type 'ESC' ('ALT') to any query to backup one (1) step

'OPSER' not running

Minutes until system shutdown (0-99) <5>? 0

12:24 PM 22-Sep-78 ###### Warning Message Phase ######
Further LOGINs are now disabled

12:24 PM 22-Sep-78 ###### Initial Job Killing Phase ######
12:24 PM 22-Sep-78 ###### Remove RTS/RES LIB Phase ######
12:24 PM 22-Sep-78 ###### SWAP File Removal Phase ######
12:24 PM 22-Sep-78 ###### Disk DISMOUNT Phase ######
12:24 PM 22-Sep-78 ###### Final Shutdown Phase ######

Please wait for system to re-boot itself
```


LOGIN Program

1. Has a spawned entry point for use by the BUILD package.

MONEY Program

1. Reports accounts using more than 65535 blocks on the public structure as > = 65535 in the "blocks used" column.

DSKINT Program

1. Supports new disks.
2. Reports the current PACKID and Pack Cluster Size if disk already has a RSTS/E format.

REACT Program

1. Sets the protection code on ACCT.SYS to 128.

ODT Program

1. Supports large files

PROGRAMS THAT USE DATA CACHING (IF ENABLED)

1. BATCH
2. ANALYS

RSTS/E V7.0 SWS UPDATE

MON

Monitor Internals

Monitor Internals

CONTENTS

Introduction..... MON-1

Monitor Modules..... MON-3
RSTS..... MON-3
TER..... MON-3
DSK..... MON-3
EMT..... MON-3
FIP..... MON-3
RSX..... MON-4
OVR..... MON-4
ODT..... MON-4
DEFAULT..... MON-4

Monitor as a SIL on Disk..... MON-5

Exercise for SIL Format..... MON-11

Feedback for SIL Format..... MON-15

The Resident Monitor..... MON-17
Miscellaneous Code..... MON-23
Map (PATCHS Module) V6C Change..... MON-25
System Wide Logical Name Entries..... MON-32

Job Related Structures..... MON-34
JOBCLK Table - V6C Change..... MON-34
FCB List Table (FCBLST)..... MON-50
"Contorted" Buffer Address..... MON-55
Unchanged JOB Data Structures..... MON-52

Device Related Structures..... MON-65
New Handler Indexes..... MON-65
New Driver Indexes..... MON-65
Unit Count Table (UNTCNT)..... MON-73
Unit Options Table (UNTOPT)..... MON-75
Disk Indexes..... MON-77
DSKMAP Table..... MON-78
Device Name Table (DEVNAM)..... MON-81
Device Maximum Unit Number Table (DEVCNT)..... MON-83
Device Pointer Table (DEVPTR)..... MON-85

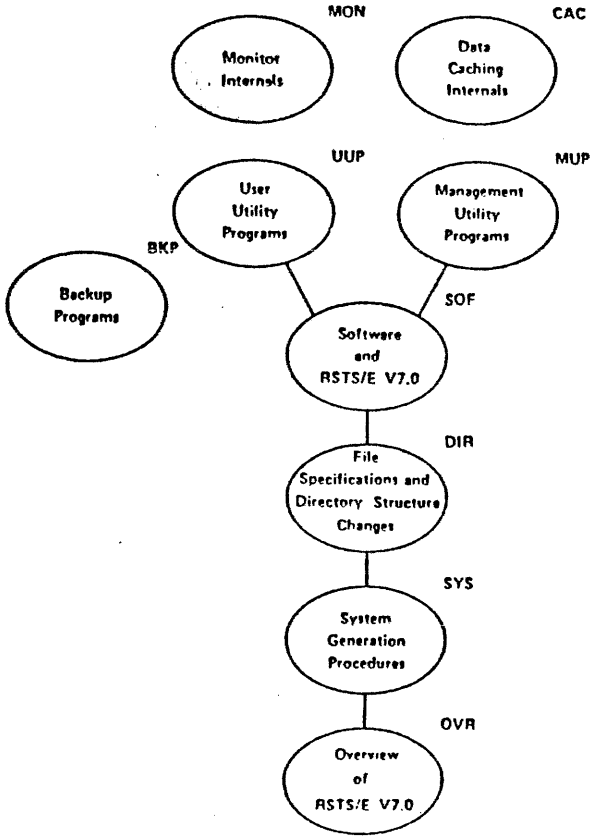
Device Table (DEVTBL)..... MON-88
Determining FIP Unit Numbers (FUN) for a System..... MON-91

Interfacing with Monitor from User Mode..... MON-94
The Null (Disappearing) Run-Time System..... MON-100

Exercise for Monitor..... MON-104

Feedback for Monitor..... MON-154

COURSE MAP



Monitor Internals

INTRODUCTION

The RSTS/E monitor performs a multitude of functions for its users. These functions include:

- Scheduling jobs to be run
- Performing all device operations
- Performing all file related operations
- Managing all resources on the system

In actuality, the monitor is responsible for the harmony and security which RSTS/E users have come to take for granted.

In this module you will learn about the changes to the monitor in RSTS/E V7.0. These changes involve not only the format of a RSTS/E monitor as it resides on disk prior to system start up, but also the operational and structural changes which affect all users during a time-sharing session.

This page is for notes.

MONITOR MODULES

RSTS

- Data structures for monitor, device and job management
- Non-disk interrupt routines
- Some monitor code

TER

- Terminal driver routines
- Mini SYSTAT routines

DSK

- Disk interrupt routines
- Disk driver code and error logging routines

EMT

- Monitor call handling routines
- Resident library handling and mapping routines
- Scheduling and swapping routines
- Memory management routines

FIP

- Directory handling routines
- File operation routines
- FIP function handling
- SAT handling routines

RSX

- RSX directive emulation routines

OVR

- Routines not made permanently resident during SYSGEN

ODT

- Monitor ODT code

DEFALT

- Default settings for the monitor

MONITOR AS A SIL ON DISK

During system generation the monitor modules are assembled and linked. Following this process the SILUS program is run to convert the linked monitor into a save image library (SIL) format. It is this SIL format which makes autopatching of the modules possible.

This section covers the format and information contained in a save image library.

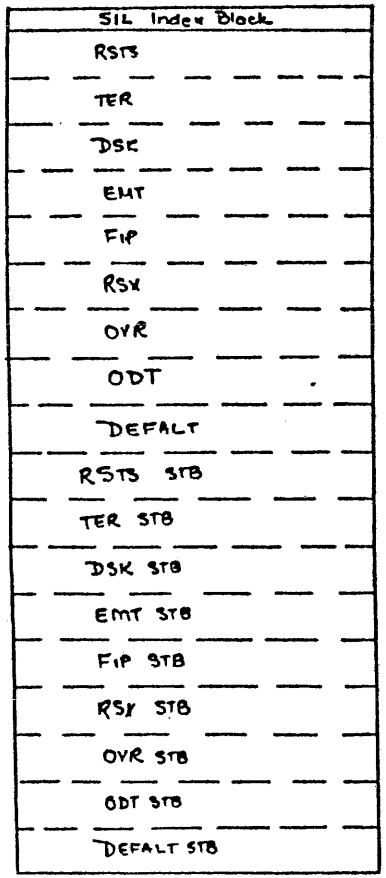


Figure MON-1. Monitor on Disk (A Save Image Library)

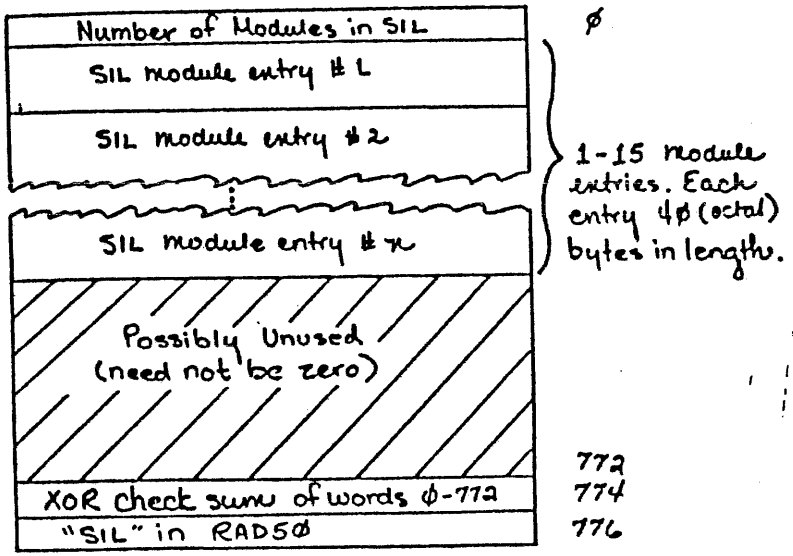


Figure MON-2. Index Block (First Block in a SIL)

SE.NAM	0
	2
SE.IDN	4
	6
SE.BLK	10
SE.STB	12
SE.STN	14
SE.LOD	16
SE.SIZ	20
SE.XFR	22
SE.SZD	24
SE.OVB	26
SE.OVN	30
Reserved (must be zero)	32
	34
SE.XXX	36

Figure MON-3. SIL Module Entry Format

SE.NAM Module name in RAD50

SE.IDN Module's .IDENT in RAD50

SE.BLK Block offset to module within SIL
- 1 First module usually
- 0 A SAV format SIL

SE.STB Block offset to module's symbol table
- 0 Module has no symbol table

SE.STN Number of symbols in module's symbol table
- 0 Module has no symbol table

SE.LOD Virtual address into which module 'loads'
Lowest address in the module's image

Low limit	Monitor SIL
Low limit	RTS
Lowest limit	Non-PIC LIB
000001	PIC LIB
000000	SAV format SIL and INIT.SYS

SE.SIZ Virtual address 'size' of module

The first illegal address of a module is SE.SIZ plus SE.LOD

High limit	Monitor SIL
177776	RTS
Highest address	Non-PIC LIB
Highest address	PIC LIB
Highest address	SAV format SIL

SE.XFR Virtual transfer address of module
Only SAV format SILs have transfer addresses.
- 000001 means no transfer address

SE.SZD Size of module on disk in blocks
For non-overlaid modules:
- SE.SZD = (SE.SIZ + 777)/1000

For overlaid modules (LIBs and overlaid RT11 programs in SAV format SILs)
- SE.SZD > SE.SIZ

SE.SZD = 0 SIL format of previous RSTS/E releases

SE.OVB Block offset to module's overlay descriptors

SE.OVN Number of overlay descriptors for module

SE.OVB, SE.OVN = 0 Monitors and RTS'

> **Note**
>
> Any module which starts on a 1000 byte multiple and is not overlaid does not need overlay descriptors. Any module which starts on a non-1000 byte multiple (e.g. INIT.SYS), which has data below its start (e.g. all RT11 programs), or which is overlaid (e.g. LIBs) require overlay descriptors

SE.XXX Reserved for SAV format SILs. Contents 'random' to all but the RT11 emulator.

Symbol in RAD5Ø (Must be unique and non-zero)	Ø
Symbol's overlay desc. no.	2
Symbol's value	4
	6

Figure MON-4. Symbol Table Entry Format

EXERCISE

1. Using the following octal dump of a SIL index block,

```
BLOCK NUMBER  00000
000/ 000012 071614 073300 140544 140122 000001 000377 002010
020/ 000000 140000 000001 000140 000000 000000 000000 000000
040/ 000000 076732 000000 140544 140122 000141 000420 002333
060/ 120000 021000 000001 000021 000000 000000 000000 000000
100/ 000000 016003 000000 140544 140122 000162 000444 002070
120/ 120000 010000 000001 000010 000000 000000 000000 000000
140/ 000000 020534 000000 140544 140122 000172 000465 002037
160/ 120000 017000 000001 000017 000000 000000 000000 000000
200/ 000000 023370 000000 140544 140122 000211 000506 002222
220/ 120000 027000 000001 000027 000000 000000 000000 000000
240/ 000000 071620 000000 140544 140122 000240 000531 002041
260/ 120000 007000 000001 000007 000000 000000 000000 000000
300/ 000000 055210 000000 140443 131716 000247 000552 002255
320/ 120000 020000 000001 000020 000000 000000 000000 000000
340/ 000000 060502 000000 140544 140122 000267 000575 002474
360/ 002000 073000 000001 000073 000000 000000 000000 000000
400/ 000000 057164 000000 140544 140122 000362 000622 000003
420/ 140000 012520 142522 000013 000000 000000 000000 000000
440/ 000000 014716 004064 140544 140122 000375 000623 000001
460/ 001000 002000 000001 000002 000000 000000 000000 000000
500/ 000000 000000 000000 000000 000000 000000 000000 000000
520/ 000000 000000 000000 000000 000000 000000 000000 000000
540/ 000000 000000 000000 000000 000000 000000 000000 000000
560/ 000000 000000 000000 000000 000000 000000 000000 000000
600/ 000000 000000 000000 000000 000000 000000 000000 000000
620/ 000000 000000 000000 000000 000000 000000 000000 000000
640/ 000000 000000 000000 000000 000000 000000 000000 000000
660/ 000000 000000 000000 000000 000000 000000 000000 000000
700/ 000000 000000 000000 000000 000000 000000 000000 000000
720/ 000000 000000 000000 000000 000000 000000 000000 000000
740/ 000000 000000 000000 000000 000000 000000 000000 000000
760/ 000000 000000 000000 000000 000000 000000 117416 074064
```

Figure MON-5.

a. How many modules are in the save image library?

_____

b. What is the name of the first module in the save image library?

_____

c. What is the module's .IDENT?

_____

d. What is the module's block offset within the SIL?

_____

e. What is the module's STB block offset (in octal)?

_____

f. How many symbols are in the symbol table (in octal)?

_____

g. What is the virtual address into which this module 'loads'?

_____

h. What is the virtual transfer address of this module?

_____

2. Using the following octal dump of a module's symbol table,

```
BLOCK NUMBER  00377
000/ 003237 071640 000000 000014 006254 012265 000000 000027
020/ 006254 012404 000000 000037 006254 015172 000000 000001
040/ 006254 021042 000000 000171 006254 024327 000000 000022
060/ 006254 053665 000000 000002 006254 062063 000000 000032
100/ 006254 075154 000000 000103 006456 021042 000000 000064
120/ 006565 130272 000000 000206 006571 130272 000000 000006
140/ 006601 130272 000000 000012 006601 130360 000000 000024
160/ 007575 071520 000000 000056 007716 127664 000000 000776
200/ 007716 127670 000000 001000 007716 130272 000000 000200
220/ 007716 130360 000000 000200 007716 130425 000000 001000
240/ 007716 130474 000000 000002 007716 130613 000000 000200
260/ 007716 131021 000000 001000 007716 131315 000000 001000
300/ 010360 071671 000000 000046 010363 076463 000000 000044
320/ 010363 107770 000000 000045 010377 135600 000000 000042
340/ 010400 157560 000000 000043 010402 000000 000000 000041
360/ 012263 124564 000000 177777 012263 125325 000000 177775
400/ 012263 125721 000000 000001 012404 127642 000000 000003
420/ 012404 127653 000000 000002 012404 127655 000000 000002
440/ 012404 127662 000000 000002 012404 127663 000000 000002
460/ 012404 127664 000000 000002 012404 127670 000000 000002
500/ 012404 130272 000000 000130 012404 130360 000000 000001
520/ 012404 130425 000000 000003 012404 130613 000000 000017
540/ 012404 131315 000000 000010 012453 021042 000000 000113
560/ 012516 073546 000000 000133 012525 023710 000000 000135
600/ 012525 062246 000000 000136 012525 062314 000000 000137
620/ 012525 062402 000000 000134 012762 045575 000000 000034
640/ 014474 021042 000000 000015 014574 074170 000000 000000
660/ 014657 131330 000000 176356 014663 127664 000000 000040
700/ 014663 127670 000000 000034 014663 130272 000000 000072
720/ 014663 130360 000000 000034 014663 130425 000000 000046
740/ 014663 130474 000000 000010 014663 130613 000000 000014
760/ 014663 131315 000000 000060 014734 042641 000000 000033
```

Figure MON-6

a. What is the name of the first symbol in the table?

_____

b. What is the value of the symbol (in octal)?

_____

FEEDBACK

1. Using the octal dump of a SIL index block,

a. How many modules are in the save image library?

10. or 12 (octal)

b. What is the name of the first module in the save image library?

RSTS

c. What is the module's .IDENT?

07.004

d. What is the module's block offset within the SIL?

1

e. What is the module's STB block offset (in octal)?

377

f. How many symbols are in the symbol table (in octal)?

2010

g. What is the virtual address into which this module 'loads'?

0

h. What is the virtual transfer address of this module?

1

2. Using the octal dump of a module's symbol table,

   a. What is the name of the first symbol in the table?

      ABORT

   b. What is the value of the symbol (in octal)?

      14

THE RESIDENT MONITOR

At system start up image copies of specific monitor SIL modules are loaded into memory and make up the 'resident monitor'. Each of these memory resident modules is called a monitor 'phase'.

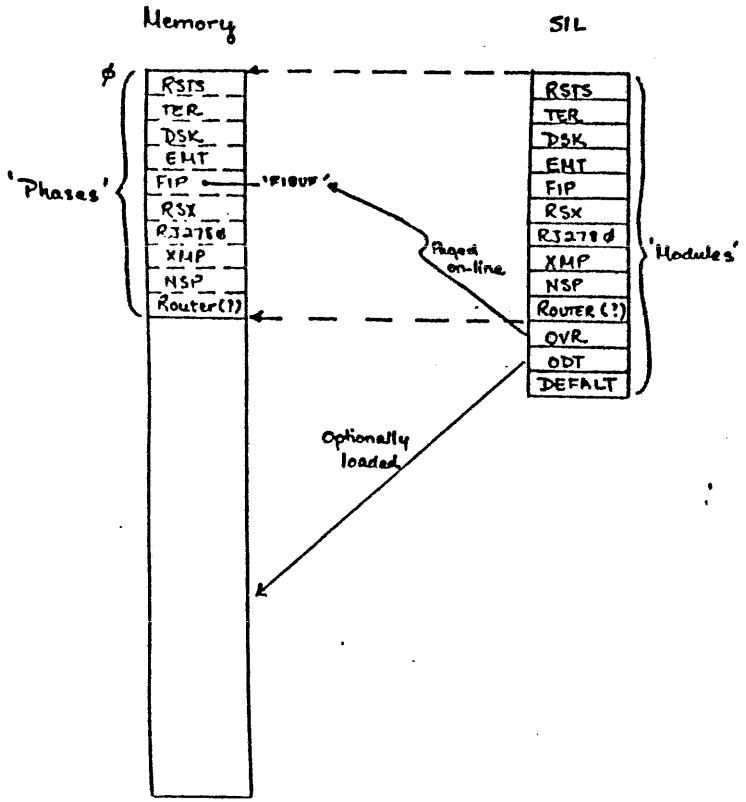


Figure MON-7. Resident Monitor

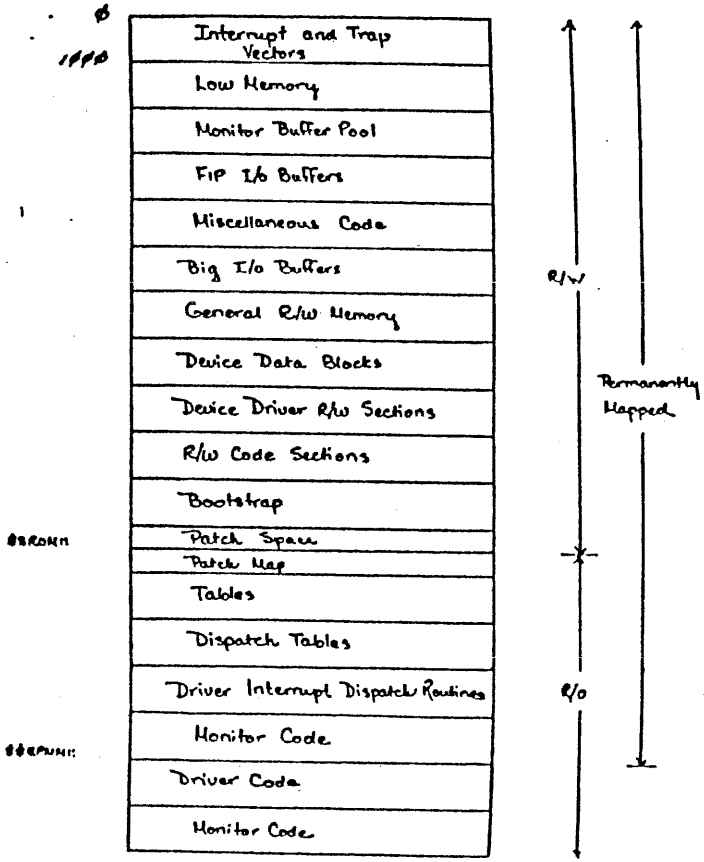


Figure MON-8. RSTS Phase

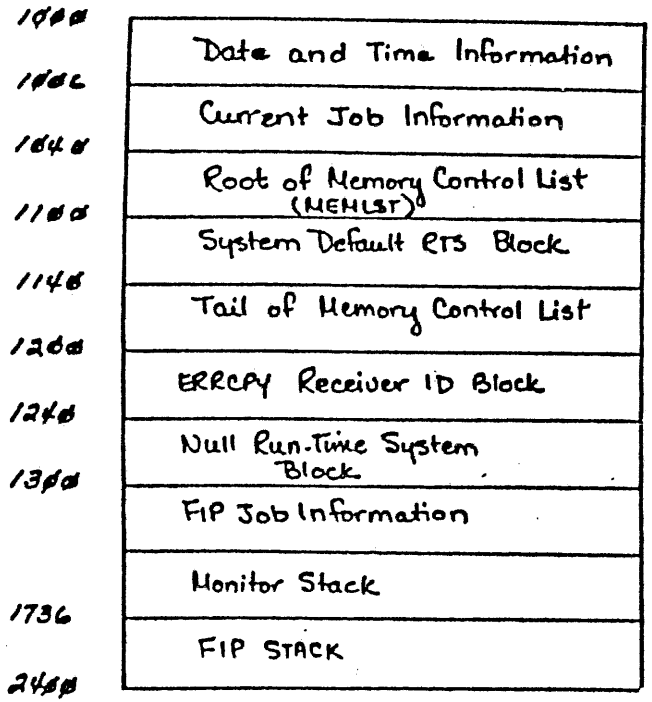


Figure MON-9. Low Memory (Fixed Locations)

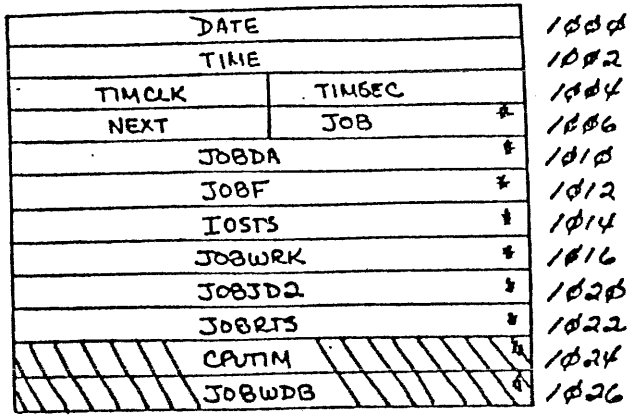


Figure MON-10. Monitor Low Memory Area

DATE	Today's date in internal form
TIME	Current time in internal form
TIMSEC	Seconds to next minute
TIMCLK	Ticks to next second
JOB	Job running now
NEXT	Next job to run
JOBDA	Pointer to current job data block (JDB)
JOBF	Pointer to current job flags (JDB @ JDFLAG)
IOSTS	Pointer to current I/O status (JDB @ JDIOST)
JOBWRK	Pointer to job's work block (WRK)
JOBJD2	Pointer to job's second job data block (JDB2)
JOBRTS	Pointer to job's run-time system block (RTS)
CPUTIM	Pointer to current CPU time bucket (@ J2TICK)
JOBWDB	Pointer to job's window descriptor block (WDB @ W.WIN1)

*Filled in by the scheduler

R.LINK = Ø		1240
R.NAME (... RSX)		1242
		1244
R.DEXT = Ø		1246
M.PPRV Ø		1250
M.PNXT Ø		1252
M.TSIZ Ø		1254
M.CTRL LCK	M.SIZE Ø	1256
M.PHYA Ø		1260
R.DATA Ø		1262
Ø		1264
R.FILE Ø		1266
Ø		1270
R.CNT		1272
R.MSIZ = 1	R.SIZE (SWPMAX)	1274
R.FLAG Ø		1276

Figure MON-11. Null Run-time System Descriptor Block

R.LINK No pointer to next RTS
R.NAME ...RSX in RAD50
R.DEXT No default extension
M.PPRV
M.PNXT No previous or next pointers in MEMLST
M.TSIZ No total size
M.SIZ No RTS size
M.CTRL Always locked in memory
M.PHYA No physical address
R.DATA Nothing on disk
R.FILE Nothing in UFD
R.CNT Access and residency counts
R.SIZE SWPMAX filled in by INIT
R.MSIZ Minimum legal job size
R.FLAG No flags

Miscellaneous Code

- 11/60 'MED' code
- Unibus Memory Parity Code
- Unibus Window Mapping Code
- Asynchronous FIP code
- Resident Library Subroutines

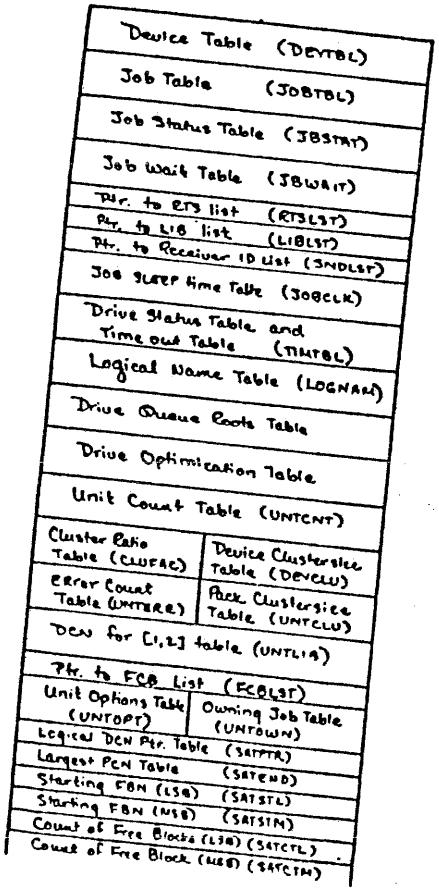


Figure MON-12. General R/W Memory

PATCH Map (PATCHS Module) - V6C change

- Bit map for mandatory and optional patches
- Sixteen words long
- Bit 'turned on' when patch is installed

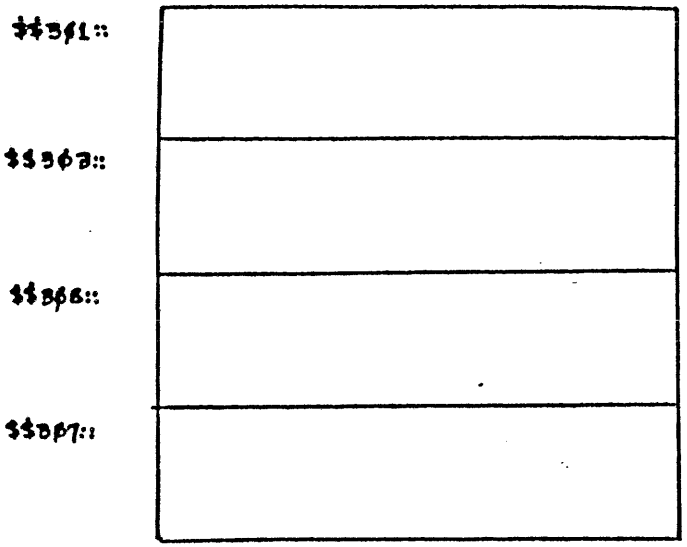


Figure MON-13. PATCHS Module Layout

$$301	(4 words)	Monitor patches 3.1.1 through 3.1.64
$$303	(4 words)	Terminal service patches 3.3.1 through 3.3.64
$$305	(4 words)	File processor patches 3.5.1 through 3.5.64
$$307	(4 words)	Device driver patches 3.7.1 through 3.7.64

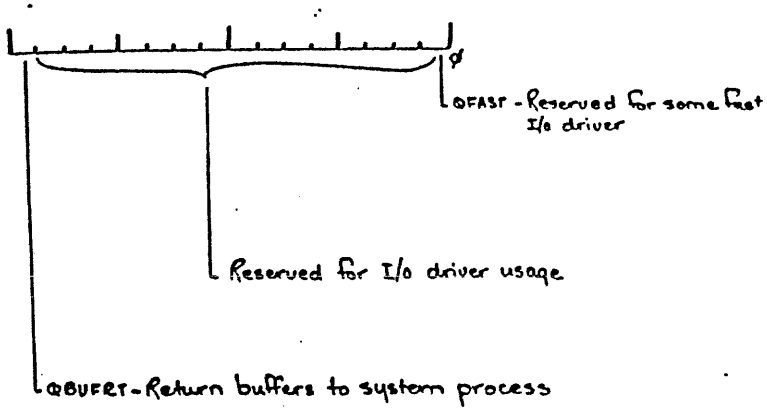


Figure MON-14. Level 3 Queue Bit Assignments (L3QUE)

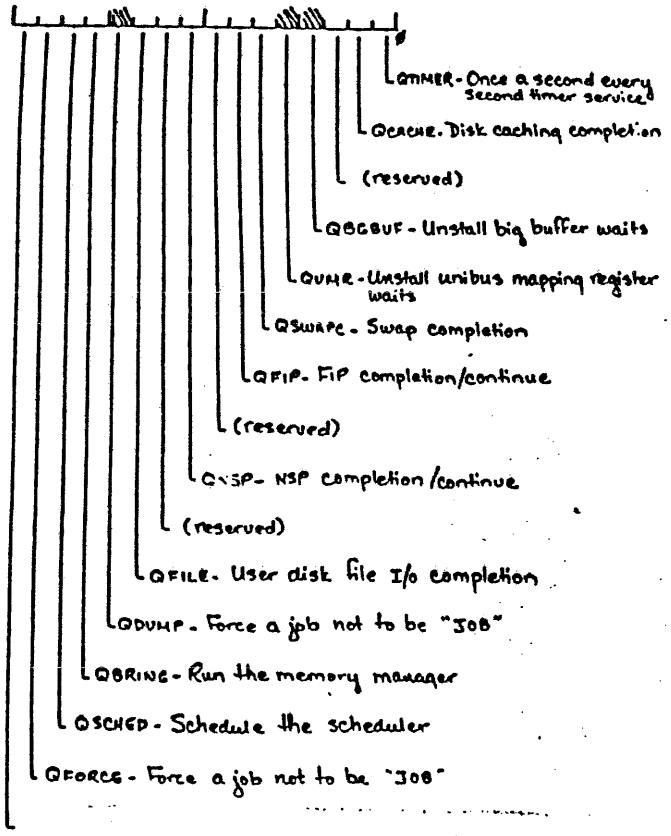


Figure MON-15. Level 3 Queue Bit Assignments (L3QUE2)

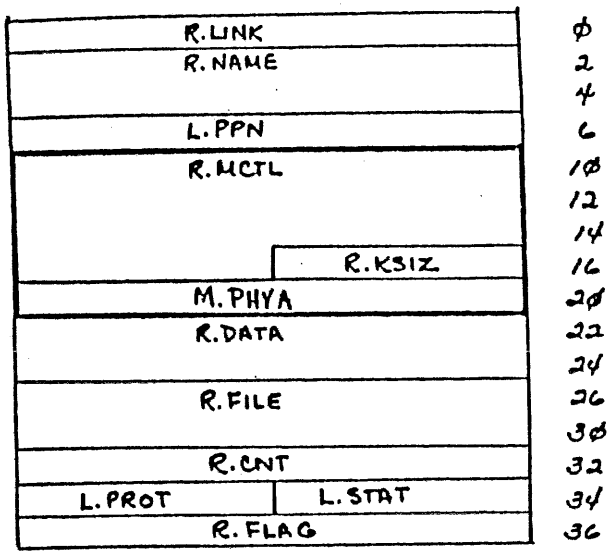


Figure MON-16. Resident Library Description Block (LIB)

R.LINK	Pointer to next resident library
R.NAME	Resident library name in RAD50
L.PPN	Resident library's owning PPN
R.MCTL	Memory control area
R.KSIZ	Resident library size in K words
M.PHYA	Physical address of start of library divided by 100 (octal)
R.DATA	FUN and FBN of file data (i.e. the library)
R.FILE	Name entry offset (/2) and FBN of UFD for file
R.CNT	Attach and map counts
L.STAT	Library's status
L.PROT	Library's protection code
R.FLAG	Resident library flags

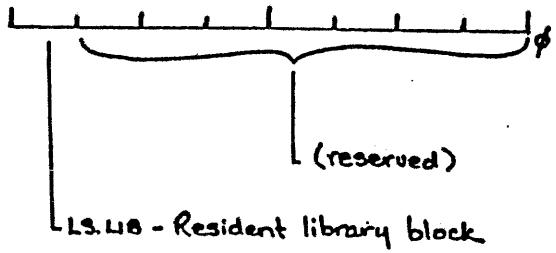


Figure MON-17. Flag Bits (L.STAT)

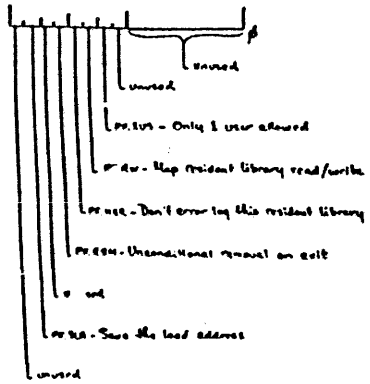


Figure MON-18. Resident Library Flags (R.FLAG)

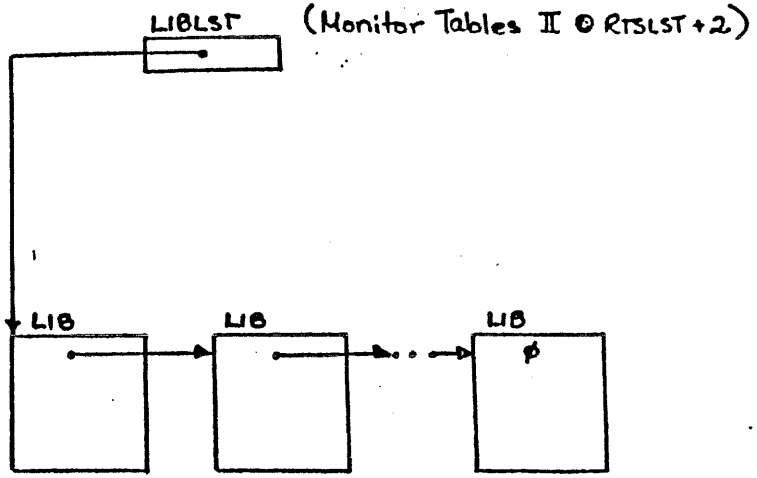


Figure MON-19. Monitor Tracking of 'Added' Resident Libraries

System Wide Logical Name Entries

- Each entry requires 5 words

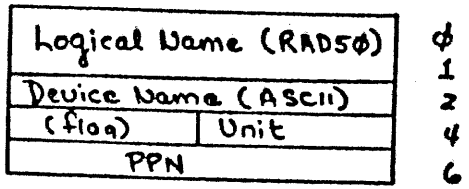


Figure MON-20

- System wide logical name table
- Each entry requires 5 words

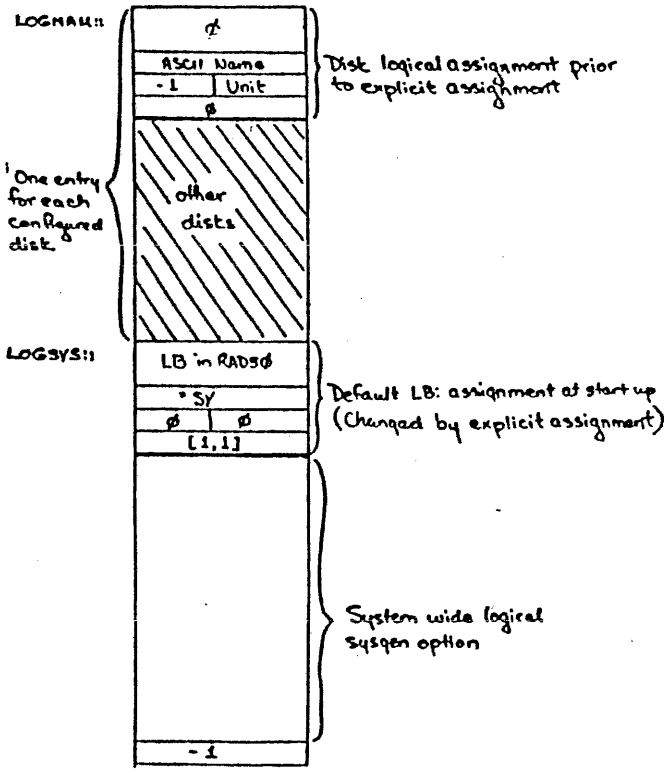


Figure MON-21. Logical Name Table (LOGNAM)

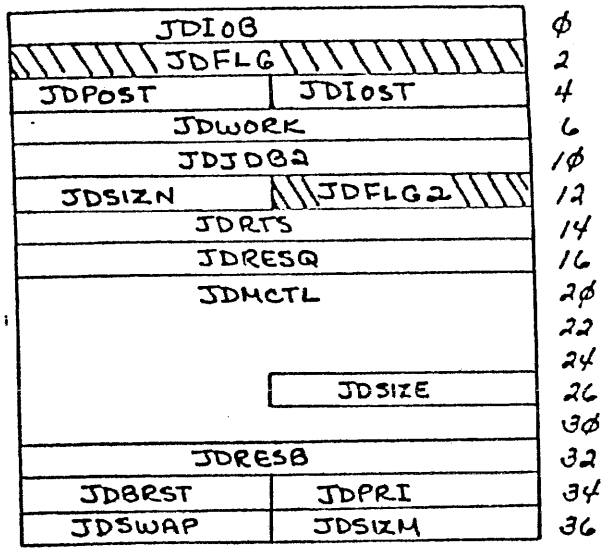


Figure MON-22. Job Data Block #1 (JDB)

JDIOB	Pointer to the I/O block (IOB)
JDFLG	Job status flags
JDIOST	I/O status for job, error codes
JDPOST	Index for bit pattern to post
JDWORK	Pointer to job work block (WRK)
JDJDB2	Pointer to job data block extension (JDB2)
JDFLG2	More job status flags
JDSIZN	Size for job to be on next residency
JDRTS	Pointer to run-time system block (RTS)
JDRESQ	Job's residency quantum
JDMCTL	Job's memory control
JDRESB	L3QUE bits to set on residency
JDPRI	Job's priority (-128. < = 127.)
JDBRST	Job's runnable burst
JDSIZM	Job's private memory maximum
JDSWAP	Job's swap parameter

JOB RELATED STRUCTURES

JOBCLK Table -- V6C Change

- Word table of sleep time for each job in seconds
- Length is job maximum + 1
- Indexed by job number times 2

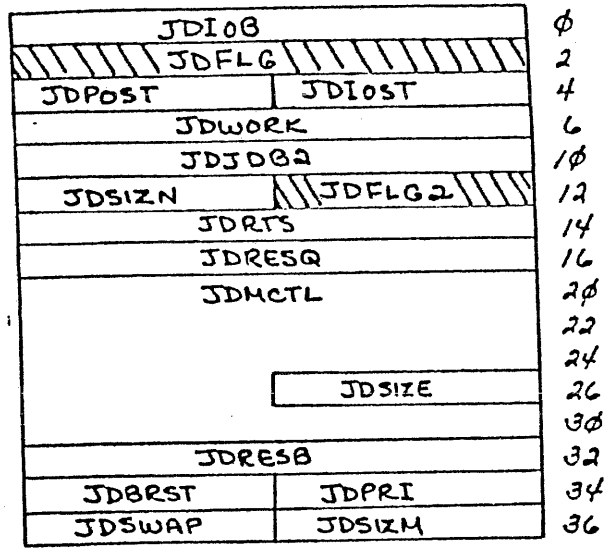


Figure MON-22. Job Data Block #1 (JDB)

JDIOB	Pointer to the I/O block (IOB)
JDFLG	Job status flags
JDIOST	I/O status for job, error codes
JDPOST	Index for bit pattern to post
JDWORK	Pointer to job work block (WRK)
JDJDB2	Pointer to job data block extension (JDB2)
JDFLG2	More job status flags
JDSIZN	Size for job to be on next residency
JDRTS	Pointer to run-time system block (RTS)
JDRESQ	Job's residency quantum
JDMCTL	Job's memory control
JDRESB	L3QUE bits to set on residency
JDPRI	Job's priority (-128. < = 127.)
JDBRST	Job's runnable burst
JDSIZM	Job's private memory maximum
JDSWAP	Job's swap parameter

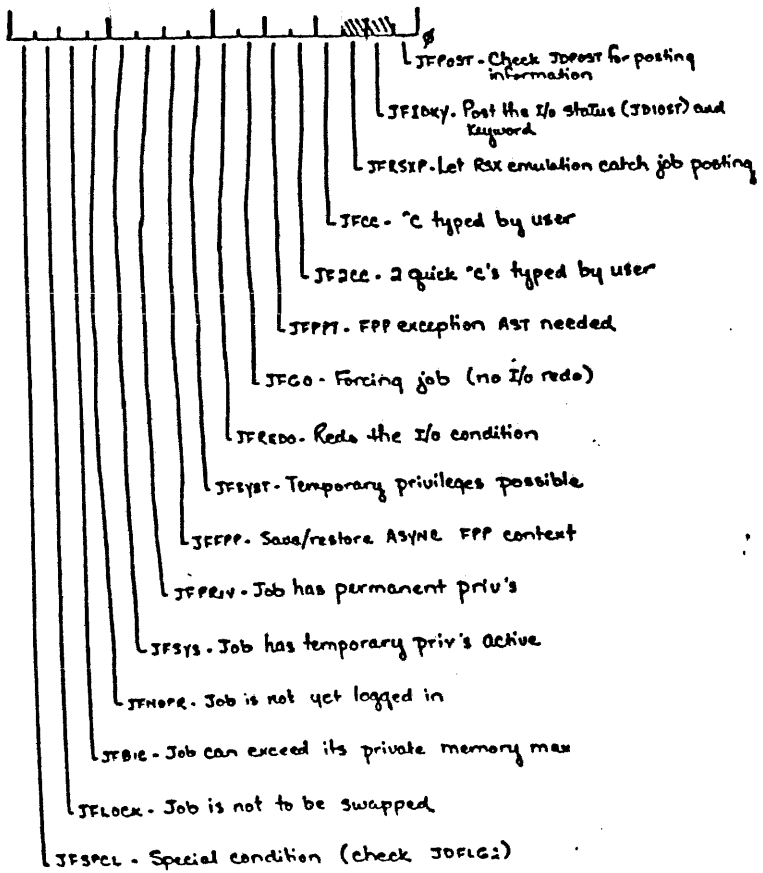


Figure MON-23. Job Flag Assignments (JDFLG)

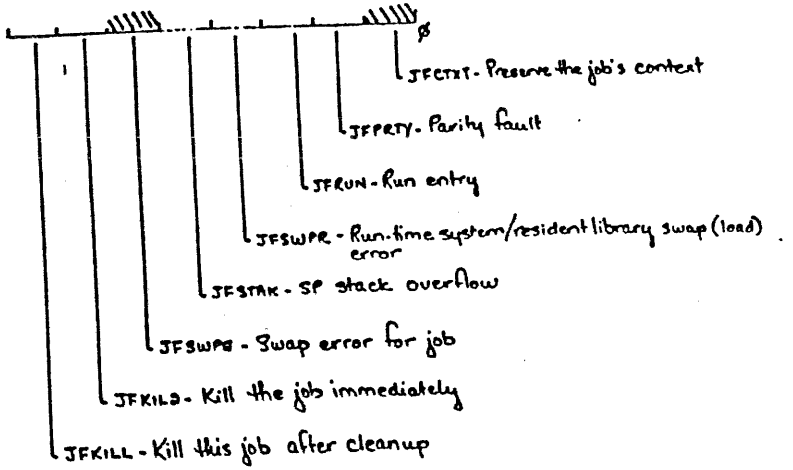


Figure MON-24. Job Flag Assignments (JDFLG2) -- V6C change

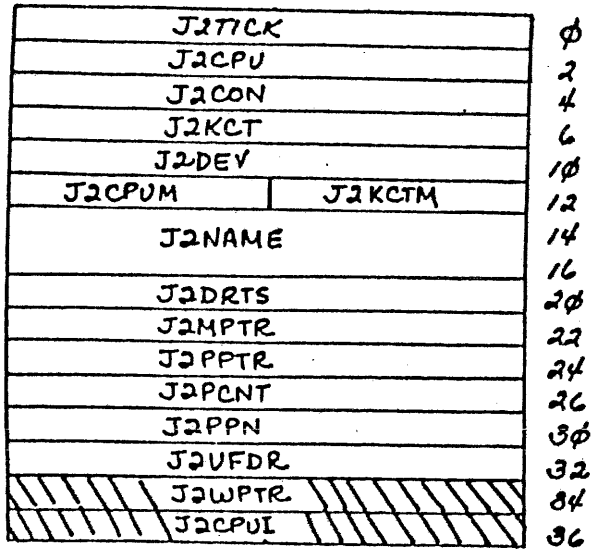


Figure MON-25. Job Data Block #2 (JDB2)

J2TICK	Number of clock ticks unconverted
J2CPU	CPU time collected for the job
J2CON	Connect time collected for the job
J2KCT	Kilo-core-ticks collected for the job
J2DEV	Device time collected for the job
J2KCTM	MSB of Kilo-core-ticks collected for the job
J2CPUM	MSB of CPU time collected for the job
J2NAME (2 words)	Job's program name
J2DRTS	Pointer to default run-time system for the job
J2MPTR	Pointer to job's receiver ID block (SND)
J2PPTR	Large data posting pointer
J2PCNT	Large data posting count
J2PPN	Job's project programmer number
J2UFDR	Retrieval pointer to job's system disk UFD
J2WPTR	Pointer to job's window descriptor block (WDB @ W.WIN1)
J2CPUI	Incremental CPU time calculation bucket

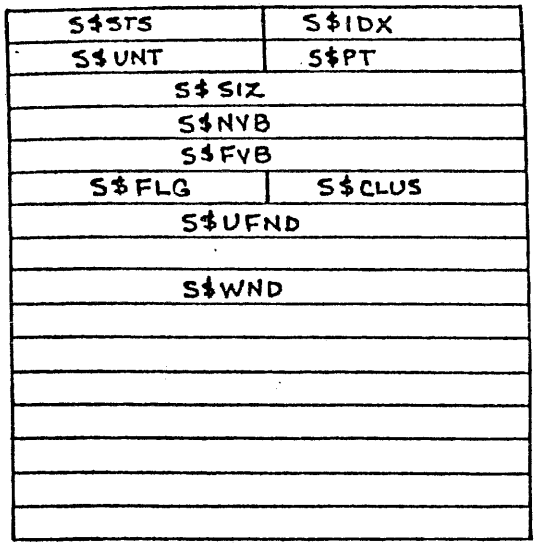


Figure MON-26. Small Control Block (SCB)

S$IDX	Driver Index
S$STS	Status bits for file
S$PT	Pending transfer count
S$UNT	FIP unit number
S$SIZ	Size of file
S$NVB	Next virtual block to read/write
S$FVB	First virtual block in window
S$CLUS	File Cluster size -1
S$FLG	SCB flag bits
S$UFND	Name entry offset (divided by 2) and FBN of UFD block
S$WND	Current retrieval window and flags

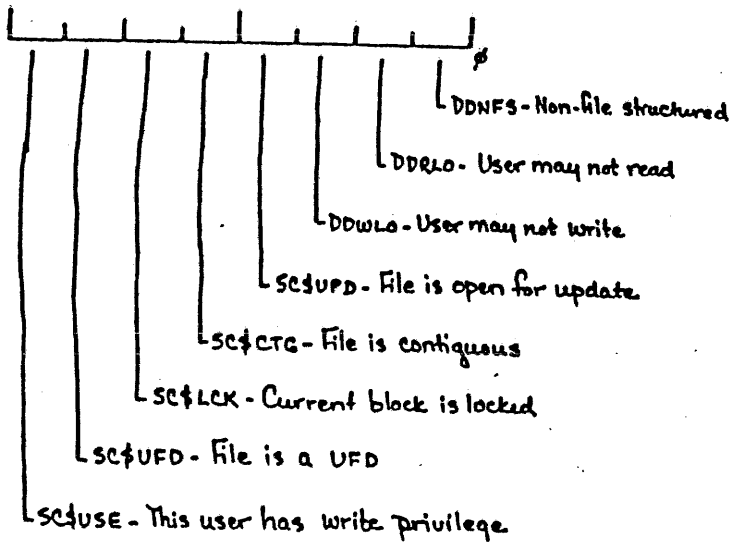


Figure MON-27. S$STS

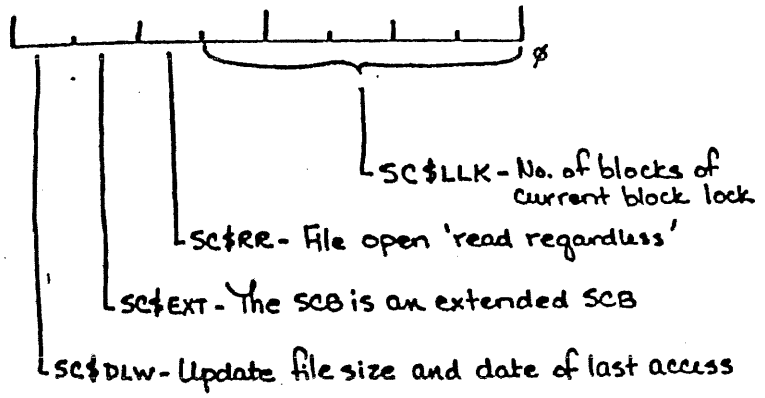


Figure MON-28. S$FLG

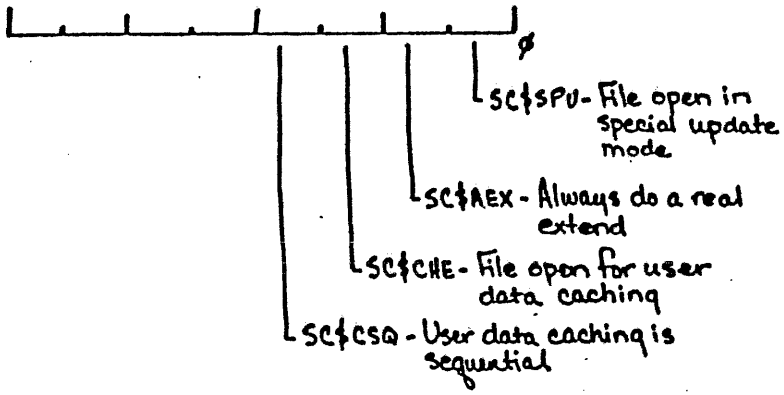


Figure MON-29. S$WND - flags

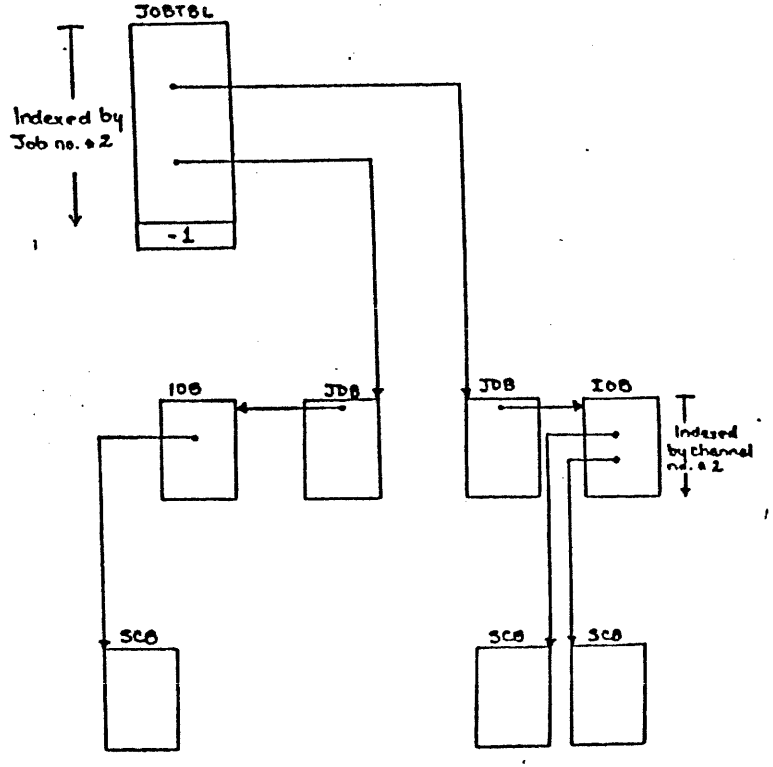


Figure MON-30. Monitor Tracking of Files Opened (Small File System)

- Write access and update information is kept on disk in the directory.

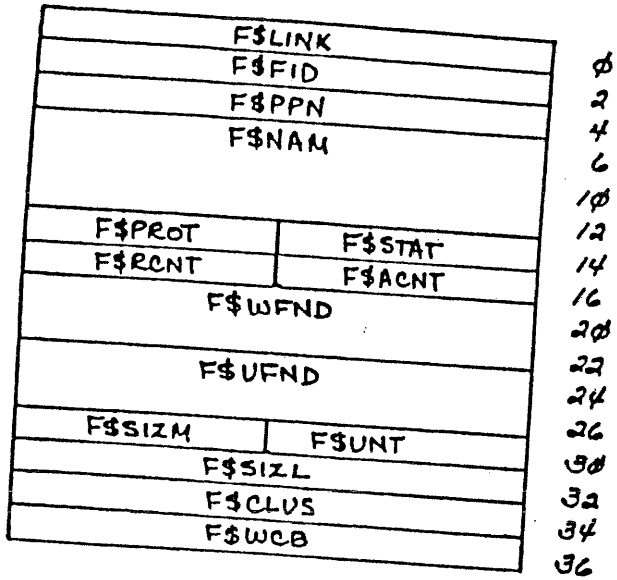


Figure MON-31. File Control Block (FCB)

FSLINK	Pointer to next FCB on this FIP unit
FSFID	File ID of file (link within UFD/MFD)
FSPPN	PPN of file
FSNAM	File name and extension in RAD50
FSSTAT	Status byte
FSPROT	Protection code
FSACNT	Access count for normal/update opens
FSRCNT	Access count for read regardless opens
FSWFND	FBB of first retrieval entry
FSUFND	FBB of name entry
FSUNT	FIP unit number
FSSIZM	File size MSB (number of FBN's if NFS)
FSSIZL	File size LSB
FSCLUS	File clustersize (WSFCB points here)
FSWCB	Pointer to first WCB open on this file

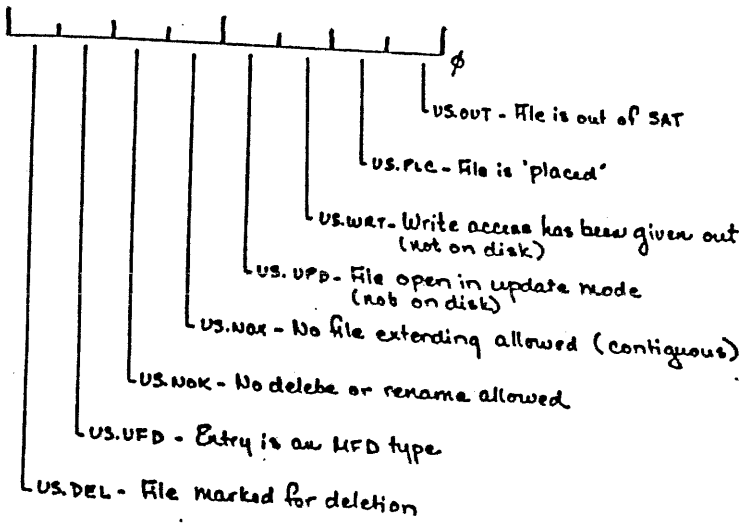


Figure MON-32. Status Byte (F$STAT)

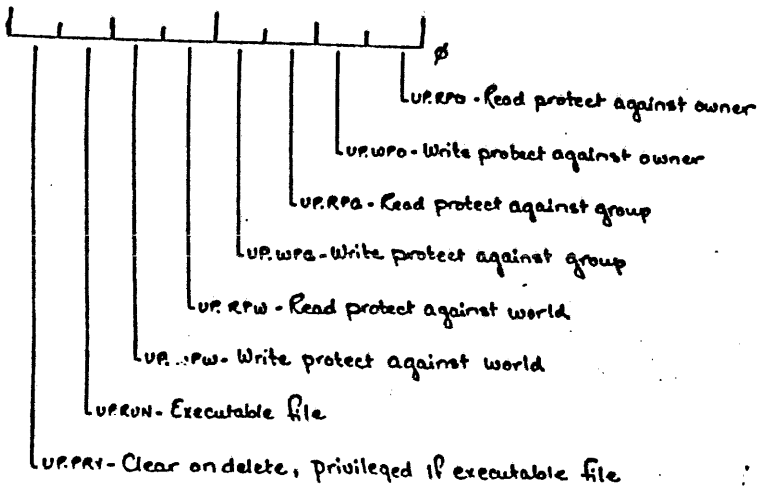


Figure MON-33. Protection Code (F$PROT)

W$STS	W$IDX	0
W$FLAG	W$JBNO	2
W$NVBM	W$PT	4
W$NVBL		6
W$FCB		10
W$REN		12
W$WCB		14
W$NXT		16
		20
		22
W$WND		24
		26
		30
		32
		34
		36

Figure MON-34. Window Control Block (WCB)

W$IDX	Driver index
W$STS	Status bits for file
W$JBNO	Job number *2 of owner
W$FLAG	WCB flag bits
W$PT	Pending transfer count
W$NVBM	Next virtual block to read/write MSB (FBN if NFS)
W$NVBL	Next virtual block to read/write LSB
W$FCB	Pointer to FCB for file @ F$CLUS
W$REN	Retrieval entry number of current window
W$WCB	Pointer to next WCB open on same FCB + flag bits
W$NXT	FBB of next window
W$WND	Current retrieval window

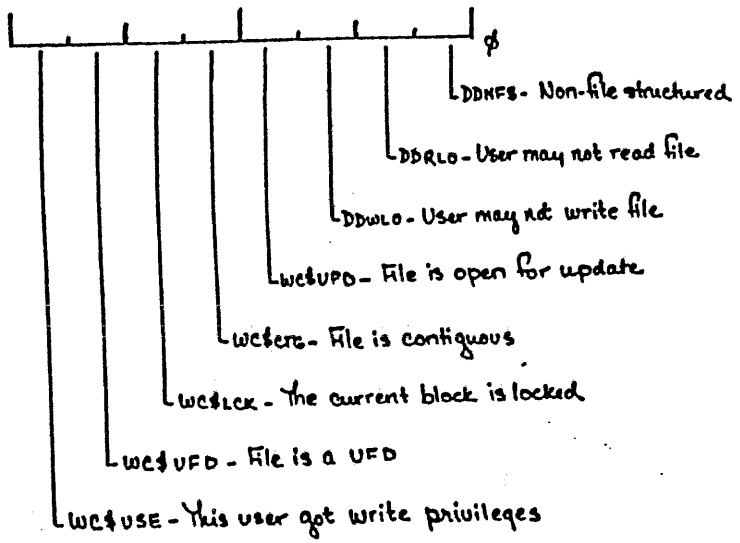


Figure MON-35. Status Bits (W$STS)

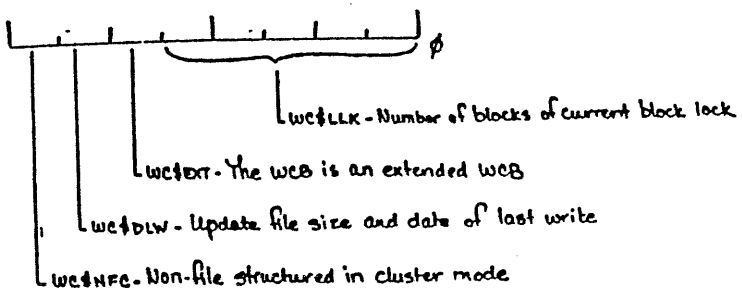


Figure MON-36. WCB Flag Bits (W$FLAG)

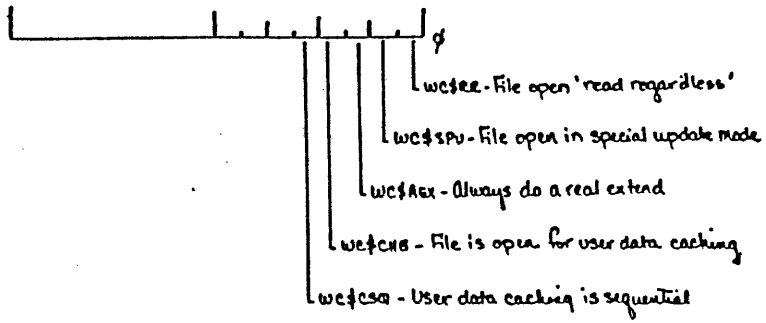


Figure MON-37. Flag Bits @ W$WCB

FCB List Table (FCBLST)

- Word table of pointers to FCBs
- One entry for every disk unit on the system
- Indexed by FIP unit number (FUN) *2

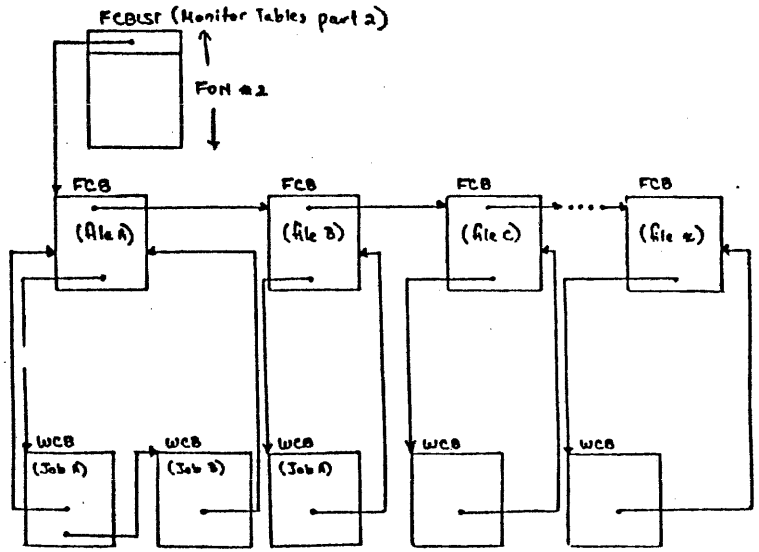


Figure MON-38. Monitor Tracking of Files Opened (Large File System)

- Write access and update information kept in FCB @ FSSTAT, not in directory on disk

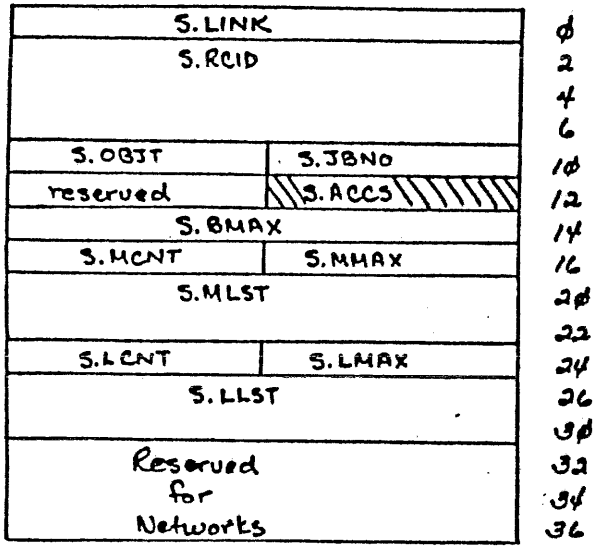


Figure MON-39. Receiver ID Block (RIB)

S.LINK Link to next block
S.RCID Receiver ID in ASCII
S.JBNO Job number times two
S.OBJT Object type (networks)
S.ACCS Access control
S.BMAX Buffer maximum in bytes
S.MMAX Message maximum
S.MCNT Message count
S.MLST Message list root/tail pointers
S.LMAX Link maximum
S.LCNT Link count
S.LLST Link list root/tail pointers

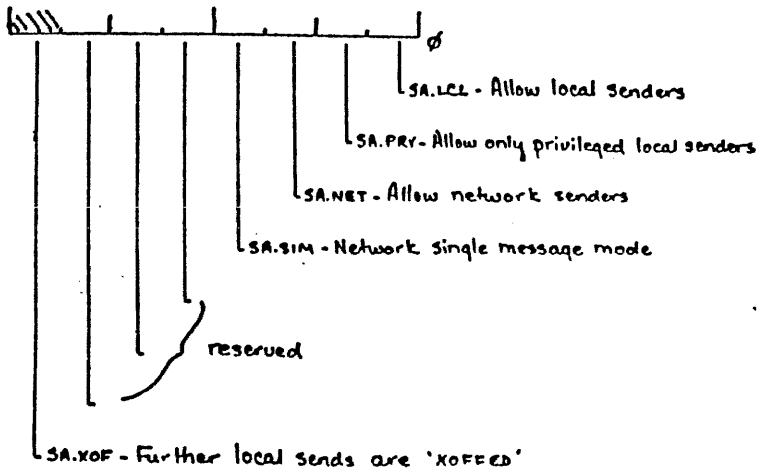


Figure MON-40. Access Control (S.ACCS)

Discontinued Utilities

- UETP - superseded by DEVTST
- SYSCAT.BAS - superseded by DIRECT
- PRIOR.BAS - superseded by UTILTY
- PIPSML.BAS - superseded by PIP.SAV
- PLX.BAS - superseded by PIP.SAV
- PIPEXT.BAS - superseded by PIP.SAV
- ROLLIN - superseded by SAVRES

BUNDLED SOFTWARE

- RMS-11K V1.0
- SORT-11 V2.0
- KMC11 driver
- II driver (IBM Interconnect)

UNBUNDLED SOFTWARE

- o BASIC-PLUS-2 V1.6
- o COBOL V4
- o DATATRIEVE V1.1
- o DIBOL/DEFORM V4C and V4D
- o RPG II V8.02
- o FORTRAN IV V2.1
- o FORTRAN IV PLUS V2.5
  - Category C
  - No RMS Support
- o KMC/3271 V1.0
- o DECNET/E V1.1
  - Virtual terminal support
- o WISE
- o DECAL

CATEGORY C SOFTWARE

- o APL-11
- o FORTRAN IV PLUS

DOCUMENTATION

1. Updated/reorganized manuals
   - o BASIC-PLUS Language Manual
   - o RSTS/E System User's Guide
   - o RSTS/E Programming Manual
   - o RSTS/E V7.0 System Manager's Guide
   - o RSTS/E V7.0 System Generation Manual
   - o RSTS/E Task Builder Manual
   - o RSTS/E V7.0 Documentation Directory
2. RSTS/E Programmer's Utilities Manual
   - o RSX based utilities manual
     - LBR
     - PAT
     - MAKSIL
     - MAC
3. RSTS/E System Directives Manual
   - o A MACRO programmer's manual
     - General monitor directives
     - RSX emulator directives
     - Run-Time Systems
4. RMS-11 User's Guide
   - o All RMS-11 documentation consolidated into one manual.
5. RMS-11 MACRO-11 Reference Manual
   - o RMS-11 implementation in MACRO-11

This page is for notes.

RSTS/E UPDATE

SYS

System Generation Procedures

System Generation Procedures

CONTENTS

Introduction..... SYS-1

Distribution Media..... SYS-2

General System Generation Procedures Flowchart... SYS-5

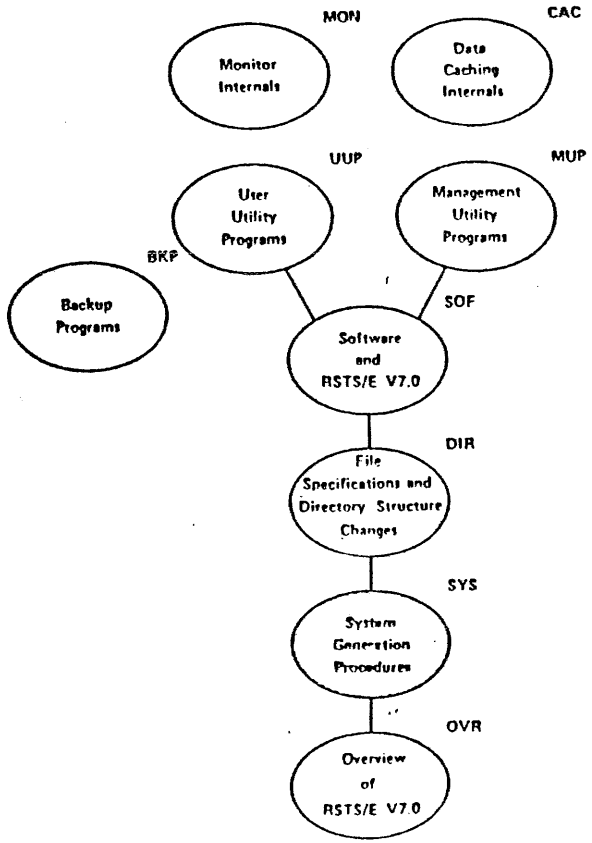
Enhanced INIT Options..... SYS-9
- DSKINT Option..... SYS-9
- HARDWR Option..... SYS-10
- REFRESH Option..... SYS-12
- DEFAULT Option..... SYS-15

New and Enhanced SYSGEN.SAV Questions..... SYS-16
- Distribution Questions..... SYS-16
- RSX Questions..... SYS-17
- Terminal Service Questions..... SYS-19
- Disk Questions..... SYS-20
- Peripheral Device Questions..... SYS-21
- Optional Monitor Feature Questions..... SYS-24

Programs Relating to Library Building..... SYS-26
- MINCOP Program..... SYS-26
- PATCPY Program..... SYS-28
- BUILD Program..... SYS-32
- ATPK Program..... SYS-37
- CSPCOM (CUSP.Compiler) Program..... SYS-44
- Autopatching Package..... SYS-45

Example System Generation..... SYS-46

COURSE MAP


MON
Monitor Internals
CAC
Data Caching Internals
UUP
User Utility Programs
MUP
Management Utility Programs
BKP
Backup Programs
SOF
Software and RSTS/E V7.0
DIR
File Specifications and Directory Structure Changes
SYS
System Generation Procedures
OVR
Overview of RSTS/E V7.0

System Generation Procedures

INTRODUCTION

Before RSTS/E can be made available for general time sharing:

- The monitor must be generated,
- The default run-time system, system library, and optional software must be built, and
- Command and control files must be created.

These procedures are normally done at system generation time.

With RSTS/E V7.0, the system generation procedures have been enhanced to allow support of:

- New hardware
- New INIT features
- New monitor features
- New library building features

This module covers the changes and enhancements to system generation for RSTS/E V7.0.

DISTRIBUTION MEDIA

Table SYS-1. Magtape Distribution Kits

DEC Order No.	BPI	Contents
7-Track	800	
AP-2772G-BC		System Generation
AP-2725G-BC		System Library I
AP-C726G-BC		System Library II
AP-C883B-BC		RSX Run-Time System and Library
AP-D047C-BC		RMS Run-Time System and Library
AP-H465A-BC		RSTS/E V7.0 Patch Kit "A"
9-Track	800	
AP-2773G-BC		System Generation
AP-2753G-BC		System Library I
AP-C725G-BC		System Library II
AP-C881B-BC		RSX Run-Time System and Library
AP-5226D-BC		RMS Run-Time System and Library
AP-H466A-BC		RSTS/E V7.0 Patch Kit "A"
9-Track	1600	(2400 ft. reel)
AB-H751G-BC		System Generation Libraries, RSX Run-Time System and Library
AB-H752G-BC		RMS Run-Time System and Library
AB-H753A-BC		RSTS/E V7.0 Patch Kit "A"

Table SYS-2. Disk Distribution Kits

DEC Order No.	Pack ID	Contents
RK05		
AN-2771G-BC	SYSGNG	System Generation
AN-2751G-BC	SYSL1G	System Library I
AN-5444G-BC	SYSL2G	System Library II
AN-C885B-BC	RSXLBB	RSX Run-Time System and Library
AN-5227D-BC	RMSKIT	RMS Run-Time System and Library
AN-H467A-BC	PATCHA	RSTS/E V7.0 Patch Kit "A"
RK06		
AM-2774G-BC	SYSGNG	System Generation, RSX Run-Time System and Library
AM-5445G-BC	SYSL1B	System Libraries
AM-5228C-BC	RMSKIT	RMS Run-Time System and Library
AM-H468A-BC	PATCHA	RSTS/E V7.0 Patch Kit "A"
RK07		
AY-D526G-BC	SYSGNG	System Generation, System Libraries, RSX Run-Time System and Library
AY-D946C-BC	RMSKIT	RMS Run-Time System and Library
AY-H469A-BC	PATCHA	RSTS/E V7.0 Patch Kit "A"

Table SYS-2. Disk Distribution Kits (Cont.)

DEC Order No.	Pack ID	Contents
RL01		
AX-D527G-BC	SYSGNG	System Generation, RSX Run-Time System and Library
AX-D528G-BC	SYSL1B	System Libraries
AX-D945C-BC	RMSKIT	RMS-Run-Time System and Library
AX-H470A-BC	PATCHA	RSTS/E V7.0 Patch Kit "A"

GENERAL SYSTEM GENERATION PROCEDURES FLOW CHART

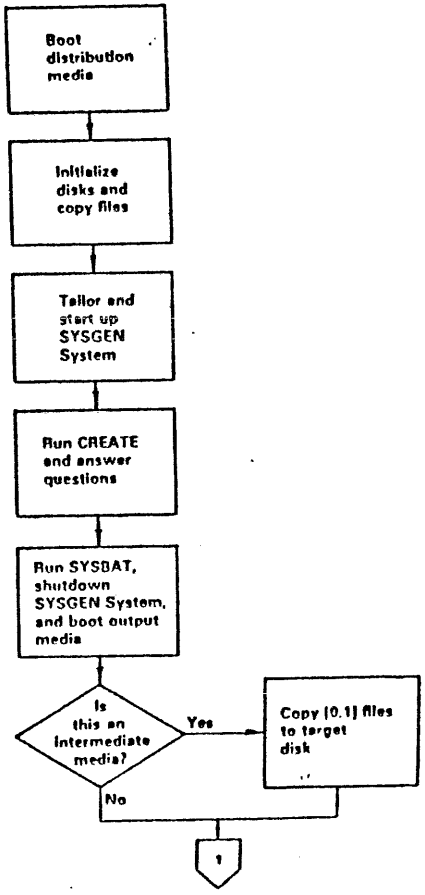


Figure SYS-1. Sysgen Flowchart (Page 1 of 4)

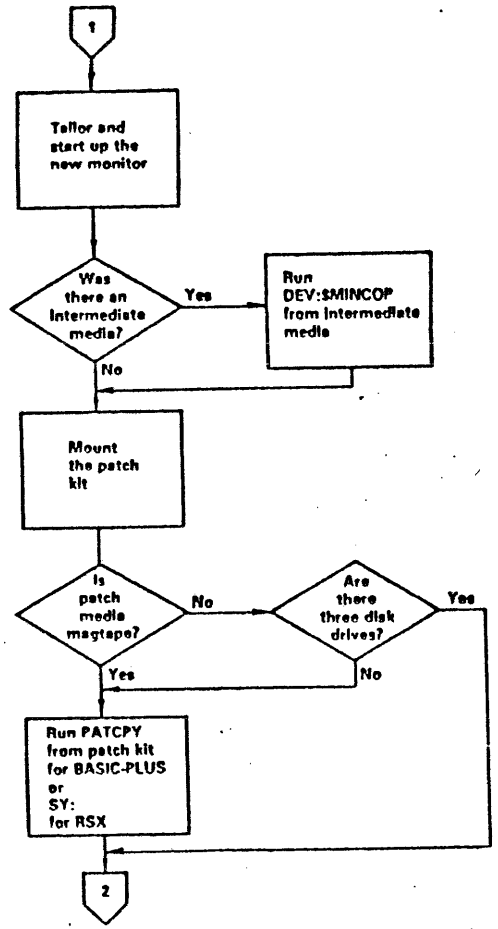


Figure SYS-1. Sysgen Flowchart (Page 2 of 4)

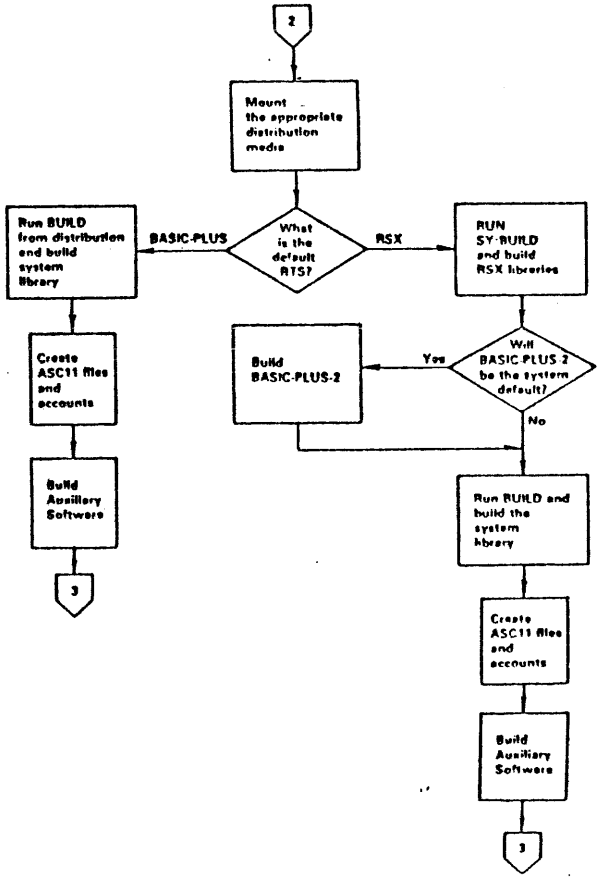


Figure SYS-1. Sysgen Flowchart (Page

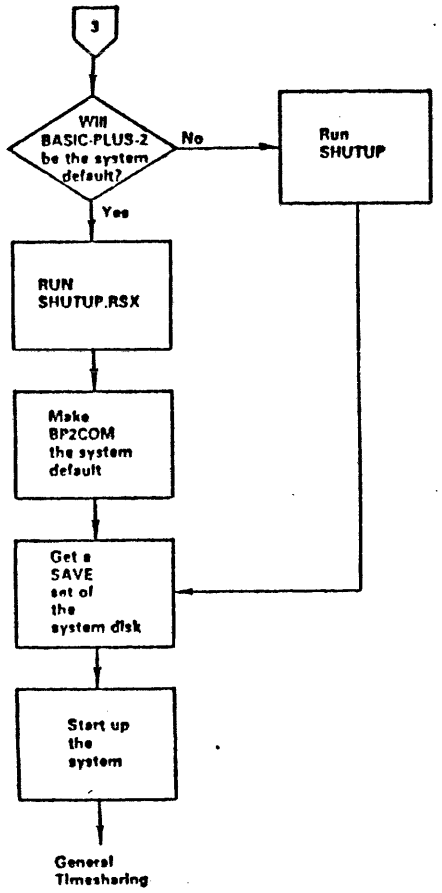


Figure SYS-1. Sysgen Flowchart (Page 4 of 4)

ENHANCED INIT OPTIONS

DSKINT Option

```
Option: DSKINT

07-Feb-79?
02:07 PM?

Disk? DM0:

This disk pack appears to be a RSTS/E formatted
disk with the following characteristics:

Pack ID :                SYSPAK
Pack Cluster Size :      8
Pack is currently :      Private,
                    Update access date on writes.


Pack ID? SYSPAK
Pack cluster size? 8
SATT.SYS base?
Type a block number to locate the file SATT.SYS at a
specific place on the disk, or type <LF> to let RSTS find a
location.
SATT.SYS base?
MFD password? V70
MFD cluster size? 16
Pre-extend directories (no) ? Y
PUB, PRI, or SYS? SYS
Library password? LIBR
Library UFD cluster size? 16
Library account base?
Type a block number to locate the library account at a
specific place on the disk, or type <LF> to let RSTS find a
location.
Library account base?
Date last modified (no)? Y
New files first (no)?
Use previous bad block info (no)? Y
Format (no)? Y
Patterns? 2
Proceed (Y or N)? Y
```

```
HARDWR Option

Option: HARDWR

HARDWR suboption? LIST

Name  Address  Vector   Comments
TT:     177560    060
RL:     174400    160    Units: 0(RL01) 2(RL02) 3(RL02)
RM:     177440    210    Units: 0(RK06) 1(RK06) 2(RK07)
RB:     176700    254    Units: 0(RP04 DP) 1(RP04 DP)
RR:     176300    150    Units:
TU:     172440    224    Units: 0(TU16 @TM03 #0) 1(TU16 @TM03 #0)

KW11L   177546    100
KW11P   172540    104
SR      177570
DR      177570

Hertz = 60.

Other: Cache

HARDWR suboption?  DISABLE

  Controller to disable? RR0

HARDWR suboption? UNITS

  Disk to set units for? DB
    Unit? 0
      Type of this unit?
    Unit? 1
      Type of this unit?
    Unit?

HARDWR suboption? TU58
  DL FOR DD0 (?)?
  DL FOR DD2 (?)? ^Z

HARDWR suboption? LIST

3 changes pending.
```



```
Name  Address  Vector   Comments
TT:     177560    060
RL:     174400    160    Units: 0(RL01) 2(RL02) 3(RL02)
RM:     177440    210    Units: 0(RK06) 1(RK06) 2(RK07)
RB:     176700    254    Units: 0(RP04 DP) 1(RP04 DP)
RR:     176300    150    Units:
(RR:    System           Disabled)
TU:     172440    224    Units: 0(TU16 @TM03 #0) 1(TU16 @TM03 #0)

KW11L   177546    100
KW11P   172540    104
SR      177570
DR      177570

Hertz = 60.

Other: Cache

HARDWR suboption? EXIT

3 changes being made.

Rebooting . . .
```

REFRESH Option

Option: REFRESH

07-Feb-79?
02:27 PM?

DISK? DM0

Clean? Y

Disk is being cleaned - wait ...

REFRESH suboption? LIST

File Name	Required?	File Flags	Status	Current Size	Minimum Size	Start LBN
System files:						
SWAP .SYS	YES		CRE		64	
SWAP0 .SYS	NO		OK			
SWAP1 .SYS	NO		OK			
SWAP3 .SYS	NO		OK			
OVR .SYS	NO		OK		54	
ERR .SYS	NO		OK		16	
BUFF .SYS	NO		OK			
CRASH .SYS	NO		OK		25	
Others:						
BADB .SYS		NOD		40		
SATT .SYS	NOD	CTG		1		13545
INIT .SYS	NOD	CTG		417		
ERR .ERR		CTG		16		649
SYSGEN.SIL	NOD	CTG		201		
RT11 .RTS		CTG		20		897

REFRESH suboption? CHANGE

SWAP.SYS changes? YES

Size? 20*31

Base? 13545

SWAP0.SYS changes?

SWAP1.SYS changes?

SWAP3.SYS changes?

OVR.SYS changes?

ERR.SYS changes?

BUFF.SYS changes?

CRASH.SYS changes? Y

Size? 80

Base?

REFRESH suboption? LIST

File Name	Required?	File Flags	Status	Current Size	Minimum Size	Start LBN
System files:						
SWAP .SYS	YES	NOD CTG	OK	2480	54	13665
SWAP0 .SYS	NO		OK			
SWAP1 .SYS	NO		OK			
SWAP3 .SYS	NO		OK			
OVR .SYS	NO		OK		54	
ERR .SYS	NO		OK		16	
BUFF .SYS	NO		OK			
CRASH .SYS	NO	NOD CTG	OK	80	25	921
Others:						
BADB .SYS		NOD		40		
SATT .SYS		NOD CTG		1		13545
INIT .SYS		NOD CTG		417		
ERR .ERR		CTG		16		649
SYSGEN.SIL		NOD CTG		201		
RT11 .RTS		CTG		20		897

REFRESH suboption?

DEFAULT Option

Option: DEFAULT

No defaults are currently set in SYSGEN.SIL

You currently have: JOB MAX = 2, SWAP MAX = 16K.

JOB MAX or SWAP MAX changes? Y

New JOB MAX? 2

New SWAP MAX? 28

You currently have: JOB MAX = 2, SWAP MAX = 28K.

JOB MAX or SWAP MAX changes?

Run Time System? RT11

Error message file? ERR

Installation name? SYSGEN

Memory allocation table:

```
  0K: 00000000 - 00127777 (  22K) : EXEC
 22K: 00130000 - 00147777 (   4K) : RTS (RT11)
 26K: 00150000 - 00757777 (  98K) : USER
124K: 00760000 - End               : NXM
```

Table suboption?

You currently have crash dump disabled.

Crash dump? Y

Magtape labelling default (none)? DOS

Preferred clock (P 100)?

Date format (ALPHABETIC)?

Time format (AM/PM)?

Power fail delay (1)? 50

NEW AND ENHANCED SYSGEN.SAV QUESTIONS

Distribution Questions

The RSTS/E system is distributed on magtape, RK05 (DK), RL01 (DL), and RK06 and RK07 (DM) cartridge disks, or may already have been transferred to the system disk (SY). For magtape, a distinction must be made between the TU10, TE10, and TS03 drives, which use the device name "MT"; the TU16, TE16, and TU45 drives, which use the name "MM"; and the TS04, which uses the name "MS". Enter the type of distribution medium for this system generation (MT, MM, MS, DK, DL, DM, or SY).

Distribution medium? !??!

RSX Questions

You have the option of installing RSX or BASIC-PLUS-2 as the system default Run-Time System. Will you want RSX as the system default Run-Time System (Yes or No)?

RSX as default RTS? !NO!

The system generation process can automatically patch the generated RSX RTS. You must have an RSX patch file on some RSTS/E file structured medium. Would you like to automatically patch the generated RSX RTS (Y or N)?

RSX patching? !??!

The RSX RTS patch file can exist on the current system disk (SY); another disk (DF, DK, DL, DM, DP, DR or DB); TU10, TE10, or TS03 magtape (MT); TU16, TE16, or TU45 magtape (MM), TS04 magtape (MS), or DECtape (DT). Enter the name of the medium which has your RSX RTS patch file.

Patch file medium?

The disk on which your patch file exists has a logical name called the pack identification. This was assigned to the disk when it was initialized. Please specify that pack ID.

Pack ID?

Enter the file name of the RSX RTS patch file. You may enter an account number (for example, [10,23]) or $ (for [1,2]) along with the file name.

Patch file name? ↓$RSXRTS.CMD ↓

Terminal Service Questions

The RSTS/E terminal service can be built to respond to CTRL/T by printing a one-line status report describing the job's status. This report includes the job name, RTS name, run state, job and RTS sizes, and total and incremental CPU times. The job is not disturbed when the report is printed. Would you like to have this feature in your monitor (YES or NO)?

One-line status report? ↓Y↓

Disk Questions

The RL11 controller is used to control any combination of up to 4 of the RL01 2.6 million word or RL02 5.2 million word top loading cartridge disk drives. Enter the number of RL01 and RL02 drives on this system (0 to 4).

RL01/RL02's?

Overlapped Seek? !Y!

The RK611 controller is used to control any combination of up to eight RK06 (6 million word) or RK07 (12 million word) top loading cartridge disk drives. Enter the total number of RK06 and RK07 drives on this system (0 to 8).

RK06/RK07's?

Overlapped seek?

The RM02/3 disk system consists of an RH11 controller and up to eight RM02 or RM03 moving-head disk pack drives. Type the total number of RM02 or RM03 drives in this configuration (0 to 8).

RM02/RM03's?

Overlapped seek?

Peripheral Device Questions

The TS04 magtape system consists of a TS11 controller and a TS04 drive. RSTS/E supports up to 8 TS04 systems. If this system has no TS04 magtape drives answer 0; otherwise, how many TS04's exist on this system (0 to 8)?

TS04's?

The TU58 DECtape - II system consists of two drives and a controller, connected through a DL11 interface. RSTS/E supports up to four TU58 systems for a maximum of eight drives. Please enter the number of TU58 drives attached to this system (0 to 8).

TU58's?

The RX11 or RX211 floppy disk system consists of a UNIBUS interface and an RX01 (single density) or RX02 (double density) floppy disk subsystem. The RX01 and RX02 each include a disk controller and two floppy disk drives. RSTS/E supports up to four RX11 or RX211 systems for a maximum of eight drives. Please enter the number of floppy disk drives attached to this system (0 to 8).

RX01/RX02's?

The KMC11 is an intelligent microprocessor controller which, when loaded with appropriate micro-codes, can be used to control various types of peripherals. How many KMC11 units do you have on this system (0 to 16)?

KMC11's?

The KMC11 can emulate the operation of various IBM protocols. This support permits the RSTS/E system to communicate with any IBM system which supports the protocol. Do you want the KMC IBM protocol support (YES or NO)?

KMC IBM protocol support? !NO!

Each KMC11 can control one DUP11 synchronous line interface to the 3271 host. How many simultaneous KMC/3271 links do you want?

Simultaneous links?

The LP driver can be made to take advantage of the extended buffer pool rather than always buffering character output from the small buffer pool. This will particularly benefit those systems which have an increased contention for system small buffers. Do you want the extended buffering for LP (YES or NO)?

Extended buffering for LP? !Y!

Optional Monitor Feature Questions

An optional feature of RSTS/E is the ability to read and write files which are longer than 65535 blocks. Including large file support also reduces file creation and access times and makes disk update mode significantly faster. Including large file support will make your monitor 2K larger, and will use more small buffers for each open file. Do you want your system to handle large files (YES or NO)?

Large files? !Y!

An extended version of FIP buffering is available under RSTS/E which allows jobs to specify, on a file by file basis, that user data files are to be cached in the extended buffer pool. This caching is in addition to the FIP directory buffering. Extended FIP buffering can also cache clusters of data, and do read ahead of user data. It requires you to allocate at least 2K words to the extended buffer pool. Do you want to include extended FIP buffering in your system (YES or NO)?

Extended data buffering? !NO!

At the cost of a physically larger monitor, the disk drivers and their subroutines can be built into a separate monitor "phase". If you want to maximize the size of your small buffer pool, you should answer "YES". Build monitor with a separate disk driver phase (YES or NO)?

Disk driver phase? !Y!

An optional feature of RSTS/E allows programs to reduce I/O for overlaying by attaching to sharable memory-resident code. If you have at least 124K words of memory, and intend to use the RMS-11 resident library, answer 'YES'.

Resident libraries? !Y!

The RSTS/E monitor can be made to take over some of the function of the RSX Run-Time System. Programs which normally run under RSX Run-Time System don't need any run-time system, so they can grow to 31K or use the address space for a resident library. This option makes the monitor 1K larger. Do you want to include RSX directives in your monitor (YES or NO)?

RSX directives? !Y!

PROGRAMS RELATING TO LIBRARY BUILDING

MINCOP Program

When an intermediate media is created during system generation, the MINCOP program must be run to copy the utility programs needed for building the system library.

RUN DEV:$MINCOP

DEV: is the device specification of the intermediate disk or magtape. Do not specify an extension for the program in the RUN command.

> **Note**
>
> If the intermediate media is a disk, the MINCOP program will logically dismount the disk when the copying is completed.

Programs Copied into SY:[1,2] by MINCOP

Table SYS-3.

Program	Prot	RTS
PIP.SAV	<232>	RT11
PIPSAV.TXT	<60>	RT11
ONLPAT.SAV	<104>	RT11
MACRO.SAV	<104>	RT11
CREF.SAV	<104>	RT11
LINK.SAV	<104>	RT11
LIBR.SAV	<104>	RT11
PATCH.SAV	<104>	RT11
PAT.SAV	<104>	RT11
UTILTY.SAV	<104>	RT11
SYSLIB.OBJ	<40>	RT11
COMMON.MAC	<40>	RT11
PATCPY.TSK*	<104>	RSX
LOGIN.TSK*	<104>	RSX
BUILD.TSK*	<104>	RSX
ATPK.TSK*	<104>	RSX
CPATCH.TSK*	<104>	RSX
AUTOED.TSK*	<104>	RSX

*Only copied if the system default run-time system is RSX.

PATCPY Program

The PATCPY program, executed prior to building the system library, transfers command and patch files from a patch kit to a disk. These files are used during the build procedure to automatically patch utility programs.

```
RUN DEV:$PATCPY
```

The device from which PATCPY is run depends upon the system default run-time system. For BASIC-PLUS, PATCPY is run from the patch kit. For RSX, PATCPY is run from SY:

Example PATCPY Dialogue

```
RUN MM0:$PATCPY

PATCPY X7.0-05 RSTS V7.0-05 TIMESHARING
Enter distribution device/PPN<MM0:[1,2]>:
Enter output device/PPN: SY:[200,200]
%Can't find file or account - SY:[200,200]
Attempt to create account SY:[200,200] <No>? Y
Account SY:[200,200] created with your password
Packages to patch? BUILD
Copying MM0: [1,2]BUILD.CMD to SY: [200,200]BUILD.CMD
Copying MM0: [1,2]PA1002.CMD to SY: [200,200]PA1002.CMD
Copying MM0: [1,2]PA1002.001 to SY: [200,200]PA1002.001
        .
        .
        .
5 files copied
Copy operation complete
```


Table SYS-4. PATCPY

Response to Packages to patch?	Files copied
ALL	*.*
EXEC	EXEC.CMD MONITR.CMD PA03??.* INIT.CMD PA01??.* RJ2780.CMD PA05??.* DECNET.* DECNT?.* PA45??.* BASIC.CMD PA04??.*
INIT	INIT.CMD PA01??.*
SYSGEN	SYSGEN.CMD PA02??.*
MONITR	MONITR.CMD PA03??.*
BASIC	BASIC.CMD PA04??.*
RJ2780	??2780.CMD PA05??.* PA25??.*
DECNET	DECNET.* DECNT?.* NETBLD.* PA45??.* PA46??.*

Table SYS-4. PATCPY (Cont.)

Response to Packges to patch?	Files Copied
BUILD	BUILD.CMD PA10??.* PA11??.* PA12??.* PA17??.* PA2218.*
SPLER	SPLER.CMD PA147?.*
BIGPRG	BIGPRG.CMD PA10??.* PA11??.* PA2219.*
BACKUP	BACKUP.CMD PA137?.*
DEVTST	DEVTST.CMD PA187?.*
RT11	RT11.CMD PA227?.*
RSX	RSX.CMD PA217?.*
RMS	RMS.* PA297?.*
SORT	SORT.CMD PA307?.*

Table SYS-4. PATCPY (Cont.)

Response to Packages to patch	Files Copied
STANDARD	INIT.CMD SYSGEN.CMD MONITR.CMD BASIC.CMD RJ2780.* DECNET.* NETBLD.* BUILD.CMD SPLER.CMD BIGPRG.CMD BACKUP.CMD DEVTST.CMD UNSUPP.CMD RSX.CMD RT11.CMD RMS.* SORT.CMD PA0???.* PA1???.* PA217?.* PA227?.* PA317?.*
Other valid responses	
BASIC2 FORTRAN DTR DMS500 DIBOLV4C DIBOLV4D RPG APL WISE DECAL DECNETV1.0 BASIC2V1.5	

BUILD Program

The BUILD program has been re-written for RSTS/E V7.0. Enhancements include:

- Autopatching as part of the BUILD procedure,
- Packages (BACKUP, SPLER, DEVTST, and Error Logging) in accounts other than [1,2],
- CUSPs build under BASIC-PLUS, RSX or BASIC-PLUS-2 with the same set of control files,
- Specifying 'LB:' on other than SY:[1,1],
- Dynamically creating START.CTL and CRASH.CTL files, and
- Automatically 'ADDs' RT11.

RUN DEV:$BUILD

The device from which BUILD is run depends upon the system default run-time system. For BASIC-PLUS execute BUILD from the distribution kit. For RSX, execute BUILD from SY:.

Example BUILD Dialogue

```
RUN-MM0: BUILD

BUILD X7.0-05 RSTS V7.0-05 TIMESHARING
System Build <No>? YES
Source Input Device <MM0:/DEN:800/PAR:ODD>?
Library Output Device <SY:>?
Target System Device <SY0:>?
Library Account <[1,2]>?

*** Copying file MM0:[1,2]BUILD.CTL to BLD1.TMP ***

Locate logical 'LB:' on <SY:[1,1]>?
Function (Build/Patch, Patch, Build) <Build/Patch>?
Save patched sources <No>?
Run-Time System <BASIC>?
Locate ERROR Package on <SY:[1,2]>? SY:[1,10]

%Can't find file or account - SY:[1,10]
Attempt to create account SY:[1,10] <No>? Y
Account SY:[1,10] created with your password
Additional Control File is <None>?
        .
        .
        .
```

Command Format for 'Source Input Device?' Prompt

[logfile=]dev:[/switches]

where

logfile

is the file specification of the file in which to log the build procedure.

dev:

is the actual source input device.

/switches

can be any one or more of the following:

```
/DEN[SITY]:800 OR /DEN[SITY]:1600 only valid for magtape
/PAR[ITY]:ODD  OR /PAR[ITY]:EVEN  only valid for magtape
/DOS                              only valid for magtape
/ANSI                             only valid for magtape
/DET[ACH]
```

The DETACH switch causes BUILD (actually ATPK) to detach. If a logfile is also specified, this can vastly speed up some BUILD/PATCH procedures, because it eliminates time spent waiting for the KB: output.

> **Note**
>
> DIGITAL recommends that the /DETACH switch is not used until after the standard CUSPs are built (BUILD.CTL).

Response to Run-Time System? question must be:

- A keyboard monitor which is present in [0,1] or is the current system default.
- A BASIC-PLUS, RSX or BASIC-PLUS-2 run-time system with an executable extension of .BAC or .TSK.

BUILD Control Files Supplied with Distribution

BUILD.CTL	Used in a system build. Utilitites needed on all systems.
BIGPRG.CTL	Large Utilities
SPLER.CTL	Operator Service and Spooling Utilities
BACKUP.CTL	BACKUP package
DEVTST.CTL	Device exercising package
RMSBLD.CTL	RMS-11 package
RSX.CTL	RSX package
TECO.CTL	TECO package

RSX.CTL Contents

```
RSX.RTS
SYSLIB.OLB
RSXMAC.SML
CSPCOM.OLB
TKB.TSK
MAC.TSK
LBR.TSK
PAT.TSK
EDT.TSK
CSPCOM.TSK
SHUTUP.RSX
EDTCOM.TXT
SLOTKB.TSK
```


ATPK Program

ATPK (pronounced "at-pee-kay") is included in RSTS/E V7.0 for use as a part of the BUILD procedure. It can also be used as a general purpose indirect command file processor. ATPK can control a job by use of a pseudo keyboard (PK) and run programs from a script previously defined in a command file. As an indirect command file processor, ATPK runs at the user's keyboard and reports the job dialogue there as it happens. ATPK can also be used as a simple batch processor, running detached and reporting the job dialogue in a log file.

Starting Up ATPK

ATPK may be invoked in one of the following ways:

```
RUN $ATPK
*<startup command>
```

or, if the CCL is installed,

```
ATPK <startup command>
```

or

```
ATPK
*<startup command>
```

ATPK will accept either of the CCL commands "ATPK" or "@" if they have been installed by the system manager.

<startup command> is of the form:

```
[<log file> =] <command file> [/<switches>]
```

Valid switches are:

/DET	Run ATPK detached. Note that if a log file is not specified, output will be to the keyboard whether or not ATPK detaches. At the end of a successful detached run, ATPK will kill itself.

/DEL	Delete command file on successful completion (used by BUILD).
/RTS:<rts>	Start the controlled job under the keyboard monitor <rts>. If <rts> is not a keyboard monitor RTS, no error will be reported and the new job will start up under the system default RTS. If this switch is not specified, the job will start under the user's current job default RTS.
/LOG	If the user is privileged, logins will be raised, if necessary, to start the controlled job (used by BUILD).
/CHA[IN]: <program> [;<line>] [=\<core command>\]	On successful completion, chain to <program> at line number <line>, if specified, with <core common> loaded in core common, if specified. Any character not contained in <core common> may be used in place of the backslashes to delimit the <core common> argument. This switch allows BUILD to patch itself or ATPK and then chain back to itself for the rest of the build.

The dialogue of the controlled job will normally be printed on the terminal. If a log file is specified, the dialogue will also be printed there. If both a log file and the /DET switch are specified, the dialogue will only be printed in the log file.

The default extensions for the command and log files are, respectively .CMD and .LOG.

Processing the Command File

ATPK usually sends each line of the control file "as is" to the PK when the job at the PK enters a program or monitor keyboard wait stall. Four characters in the command file are handled specially, however:

$	as first character on line, indicates an ATPK command.
!	as first character on line, indicates a comment.
^	indicates a control character.
←	indicates that the next character is not a special character. (i.e., an underscore character quotes the following character.)

ATPK Commands

Any command file line beginning with a dollar sign ($) will be considered an ATPK command. Valid ATPK commands are:

$LOG OFF	Turn dialogue reporting to the log file (and/or keyboard) off.
$LOG ON	Resume printing the controlled job's dialogue in the log file (and/or at the keyboard).
$ALLOW NO ERRORS	Abort if any (fatal or warning) errors are encountered by the controlled job.
$ALLOW WARNING ERRORS	Abort if a fatal error is encountered by the controlled job, but allow warning errors.
$ALLOW FATAL ERRORS	Do not abort if any errors are encountered by the controlled job (default setting).

$WAIT — Accept a line of input from the keyboard and send it as input to the controlled job. If a program is running at the controlled job after this line has been processed, continue accepting command lines from the keyboard until the program has stopped running and the controlled job is in a keyboard monitor wait (that is, SYSTAT would show the job in a "^C" state). After this command has been processed, input will resume from the command file.

$@<command file> — Switch input command stream to <command file>. ATPK will abort if the command file is not found. No nesting of these indirect command files is allowed; ATPK will abort if it finds the "$@" command in the <command file>.

$DETACH — Causes ATPK to detach. This command may be used after the $WAIT command to cause ATPK to detach after user input at the start of the controlled job.

$LOGIN [KB[n]:] [(p,pn)]

Log in a keyboard number n under the account specified by (p,pn). This command, which is only available to privileged users, causes ATPK to create a job running LOGIN with instructions to attach to the specified keyboard and log in under the specified account. If (p,pn) is not specified, the account that ATPK is running under will be used. If n is not specified (for example, $LOGIN KB:), LOGIN will be instructed to attach to the keyboard from which ATPK was started. This may be useful to log the user back in following a command that started up ATPK with the

/DET switch. If the keyboard specification is missing altogether, a new job will be started at the pseudo-keyboard under ATPK's control. The current job at the pseudo keyboard, if any, will first be killed in this case. If LOGIN fails to attach on log in at the pseudo-keyboard under ATPK, an error message will be printed and ATPK will abort. If LOGIN fails at any other keyboard, no error will be returned to ATPK.

Comments

Any line starting with an exclamation point will be recorded in the log and not sent to the controlled job.

Control Characters

Any character preceded by an uparrow will be converted to a control character before being sent to the controlled job. If a command line consists only of an uparrow followed by a single character, the single control character will be sent to the controlled job with no extra line terminator.

Quoted Characters

Any "$" or "!" that is meant to be the first character of a line to be sent to the controlled job, or any "^" or "_" in the command file that is not to be interpreted as a special character by ATPK must be preceded with the underscore character. Every underscore character (except those preceded by another underscore character) will be removed from the command line before it is sent to the controlled job.

Using ATPK for System Startup

If it is desired, most of system startup may be controlled under ATPK. For example, the file START.CTL to be processed by the INIT utility could contain only the following:

```
DETACH
LOGIN KB0: [1,2]
FORCE KB0: RUN $ATPK
FORCE KB0: $START.LOG=$START.CMD/DET/LOG
END
```

When designing the ATPK startup command file, most of the functions of INIT will have to be performed with UTILTY. The recommended procedure for bringing a non-system disk on-line after a crash is to clean the disk with ONLCLN.SAV and then MOUNT and UNLOCK the disk with UTILTY.

When starting up the spooling package under ATPK, two points in particular should be noted. First, the $LOGIN command should be used to re-log in the pseudo-keyboard after a spooling program detaches. Secondly, when the spooling programs are invoked, they generally prompt for input and then sleep while awaiting a response. Since ATPK will not respond unless the controlled program enters an input stall (as opposed to a sleep), ATPK will not detect when the spooler needs the command line. This problem may be circumvented by including the startup command(s) on the same command file line as the command that invokes the spooling program. If the commands are separated with an uparrow-M, ATPK will insert a carriage return (the equivalent of a CTRL/M) between the two commands, causing them to be sent to the pseudo-keyboard at the same time, but as two lines. As an example, the following sequence in the ATPK command file (START.CMD in the above example) will start up OPSER, QUEMAN, one line printer spooler and one batch processor.

```
RUN $OPSER ^MLOG OPSER.LOG;ALL ^MCHA KB0: ^MDETACH
$LOGIN
RUN $QUEMAN ^MDETACH
$LOGIN
RUN $SPOOL ^MLP0:
$LOGIN
RUN $BATCH ^MBA0:/NOQUEUE/NODELETE/ERROR:FATAL
$LOGIN
```


When using ATPK to start up programs such as VT50PY, care must be taken to avoid sequencing problems. If VT50PY is invoked and directed to detach and display information on another terminal, immediate attempts to force commands to that terminal that would select the desired VT50PY features to be displayed may be forced to the terminal before VT50PY has a chance to open the terminal. In this case, the forced characters may invoke LOGIN at that terminal resulting in a failure by VT50PY to gain access to the terminal. This problem may be avoided by having ATPK execute some other command(s) (such as an immediate mode SLEEP statement, if running under the BASIC-PLUS Run-Time System) after starting the display program and before forcing to the display program's terminal.

Other Features

By specifying a log file and KB: as your input file, you can make a copy of interactive dialogue to queue to the line printer, etc. Note that ATPK will not be completely transparent in such a session. The echo of everything you type is typed back at you again after you end each line, and nothing you type will be sent to the PK until the next time your controlled job enters a KB input wait.

When inputting from the terminal, CTRL/Z's get passed on through to the controlled job. A CTRL/C at any time will cause the controlled job to be killed, the log file closed, and ATPK to exit. A CTRL/C may be sent to the job by typing uparrow-C.

Installing ATPK As a CCL Command

The following sequence will install the CCL commands "ATPK" and "@" to run the ATPK program. Either or both of the commands may be installed.

```
RUN $UTILTY
#CCL AT-PK=[1,2]ATPK.*;PRIV 30000
#CCL @-=[1,2]ATPK.*;PRIV 30000
#^Z
```

(This assumes that ATPK and UTILTY are both located on SY:[1,2].)

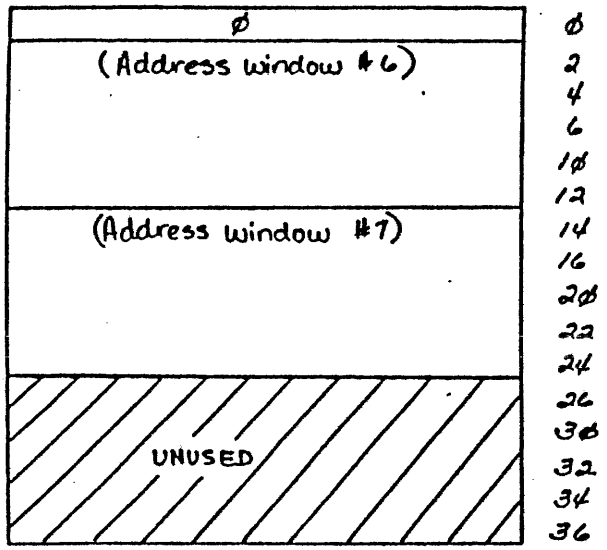


Figure MON-46. Window Descriptor Block #3 (WDB)

o Exists only if more than five APRs are mapping a resident library (or libraries)

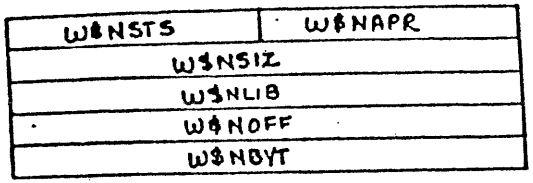


Figure MON-47. Address Windows

W$NAPR	Window base APR (1-7)
W$NSTS	Window status
W$NSIZ	Window size/100
W$NLIB	Pointer to library pointer in W.ALIB
W$NOFF	Map offset/100 into library
W$NBYT	Map length in bytes

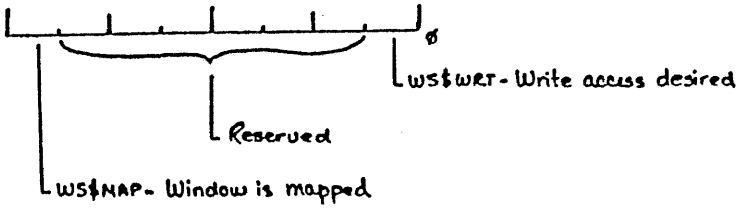


Figure MON-48. Flag Bits (W$NSTS)

Unchanged JOB Data Structures

Tables	Blocks
JOBTBL	RTS
JBSTAT	PMD
JBWAIT	IOB
JSBTBL	WRK
	MEMLST Sub block

DIGITAL RSTS/E SWS UPDATE
Monitor Internals

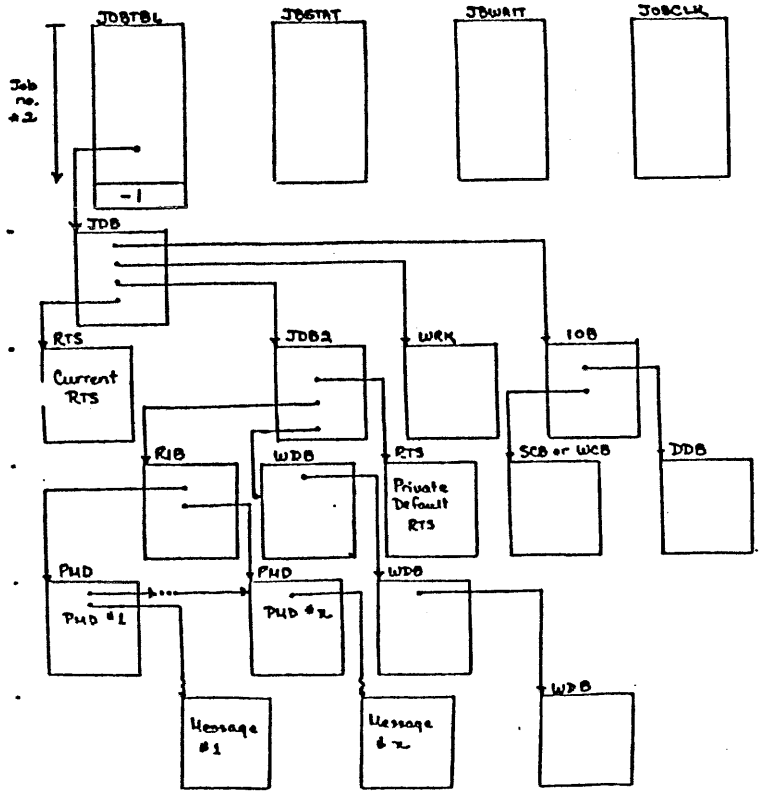


Figure MON-49. Job Structure

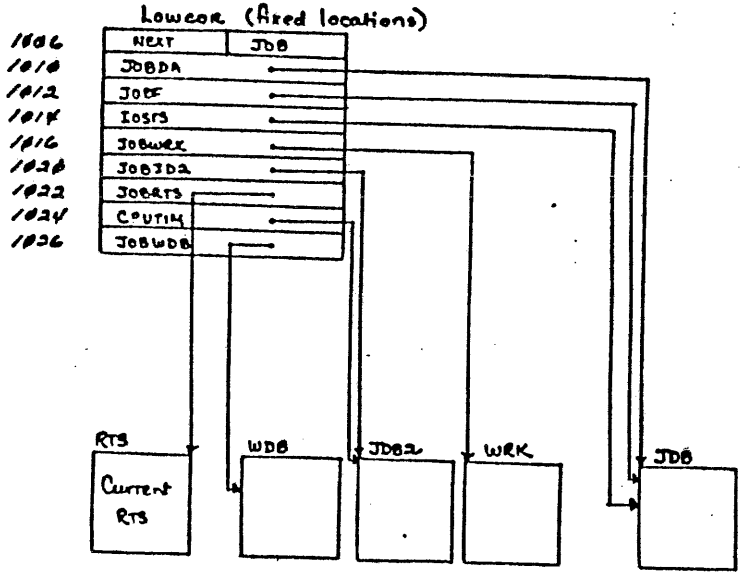


Figure MON-58. Monitor Tracking of "JOB"

DEVICE RELATED STRUCTURES

New Handler Indexes

36	DT2HND	DECtape II
40	KMCHND	KMC11 Microprocessor
42	IBMHND	IBM Interconnect

New Driver Indexes

- Values dynamically assigned at SYSGEN

IDX.MS	TS04 Magtape
IDX.DD	TU58 DECtape II
IDX.XK	KMC11 Microprocessor

Device Name	Alternate Name	DDB Size (bytes)	Control C Interruptable	Buffer Quota	Check before SLEEP	Handles Unibus map entries
KB	TT	(varies)	Yes		Yes	-
NL	-	8.	No	-	No	-
PK	-	12.	Yes		Yes	-
RJ	-	12.	No	-	No	-
XM	-	48.	Yes	-	Yes	-
LP	-	32.	Yes	20.	No	-
PR	-	16.	Yes	10.	No	-
PP	-	16.	Yes	10.	No	-
CR	-	178.	Yes	-	No	-
CD	CR	178.	Yes	-	No	-
MT	-	38.	No	-	No	Yes
MM	MT	38.	No	-	No	Yes
MS	MT	90.	No	-	No	Yes
DT	-	32.	No	-	No	-
DX	DY	28.	No	-	No	Yes
DD	-	24.	No	-	No	-

Figure MON-51. Device Information

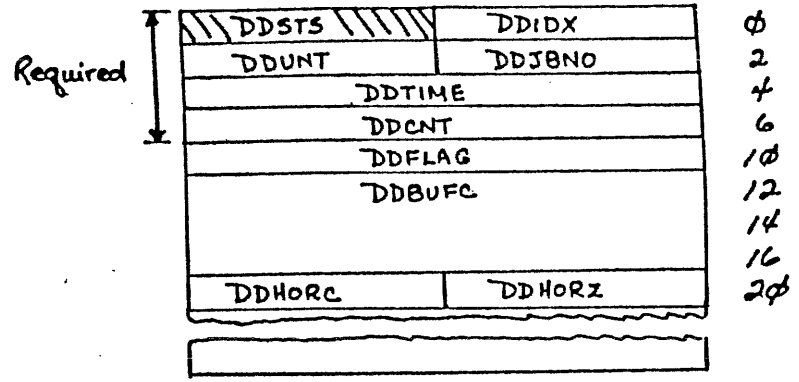


Figure MON-52. Device Data Block (DDB)

DDIDX	Driver index
DDSTS	Status and access control
DDJBNO	Owner job number times 2 (0 if free)
DDUNT	Device unit number
DDTIME	Time assigned or inited
DDCNT	Init count and assignment control
DDFLAG	Device dependent flags
DDBUFC	Buffer chain control area
DDHORZ	Horizontal position
DDHORC	Characters per line + 1

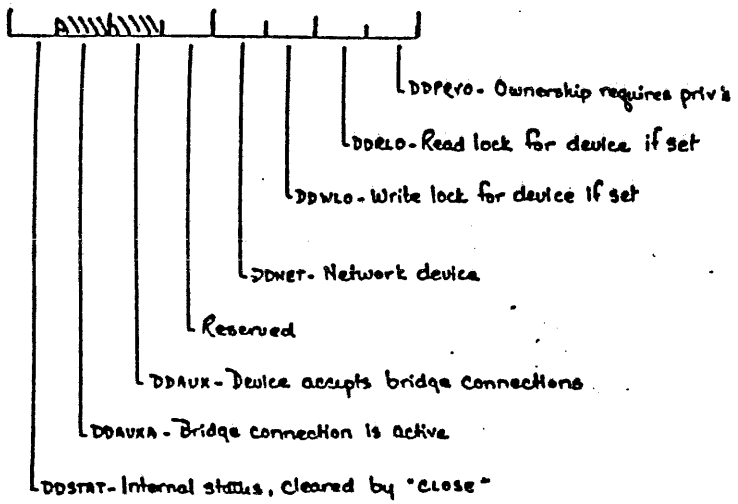


Figure MON-53. Status and Access Control (DDSTS)

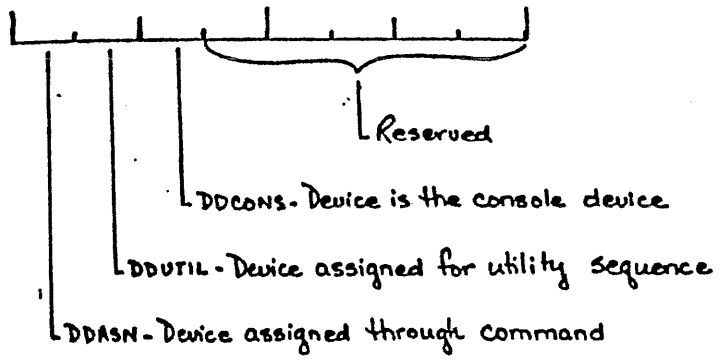


Figure MON-54. Init Count and Assignment Control

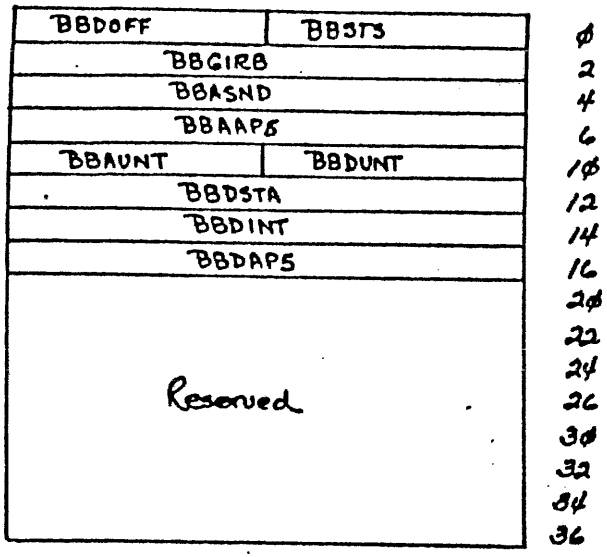


Figure MON-55. Device Driver to Auxiliary Processor Bridge Block (BB)

BBSTS	Status
BBDOFF	Data offset within data buffer
BBGIRB	Pointer to "GET IRB/data buffer" subroutine
BBASND	Pointer to "SEND IRB/data buffer" subroutine
BBAAP5	Pointer to PAR5 value for above two subroutines
BBDUNT	Device's control unit number
BBAUNT	Auxiliary's control unit number
BBDSTA	Pointer to "status change" entry in device driver
BBDINT	Pointer to "interrupt" entry in device driver
BBDAP5	Pointer to PAR5 value for above two entry points

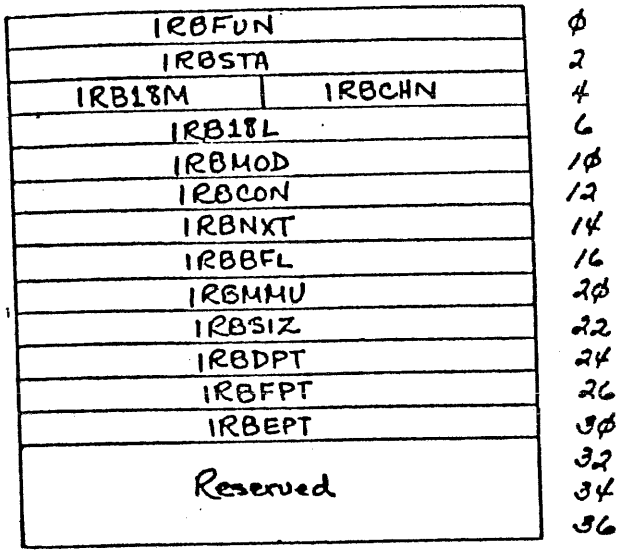


Figure MON-56. Interprocess Request Block (IRB)

IRBFUN	Function code and its modifier
IRBSTA	Status
IRBCHN	Channel number
IRB18M	MSB of 18-bit unibus address of data within buffer
IRB18L	LSB of 18-bit unibus address of data within buffer
IRBMOD	Data modifier
IRBCON	Contorted address of data buffer
IRBNXT	Link to next IRB (if chained)
IRBBFL	Byte size of data within data buffer
IRBMMU	MMU address of whole data buffer
IRBSIZ	Byte size of whole data buffer
IRBDPT	Virtual (through PAR6) done pointer
IRBFPT	Virtual (through PAR6) fill pointer
IRBEPT	Virtual (through PAR6) empty pointer

Name	Description
UNTCNT	Device flags and open counts
DEVCLU	Device cluster size - byte table
CLUFAC	Cluster ratio (PCS/DCS) - byte table at DEVCLU + 1
UNTCLU	Pack cluster size - byte table
UNTERR	Error count - byte table at UNTCLU + 1
UNTLIB	Starting DCN for [1,2] UFDs
UNTSIZ	Unit size in DCNs
UNTOWN	Owning job number - byte table
UNTOPT	Disk status bits - byte table at UNTOWN + 1
MID$XX	Massbus ID

Figure MON-57. Tables Indexed by FUN #2

Unit Count Table (UNTCNT)

- Word table of device flags and open counts
- One entry for each disk unit configured on the system
- Address returned in Monitor Tables I

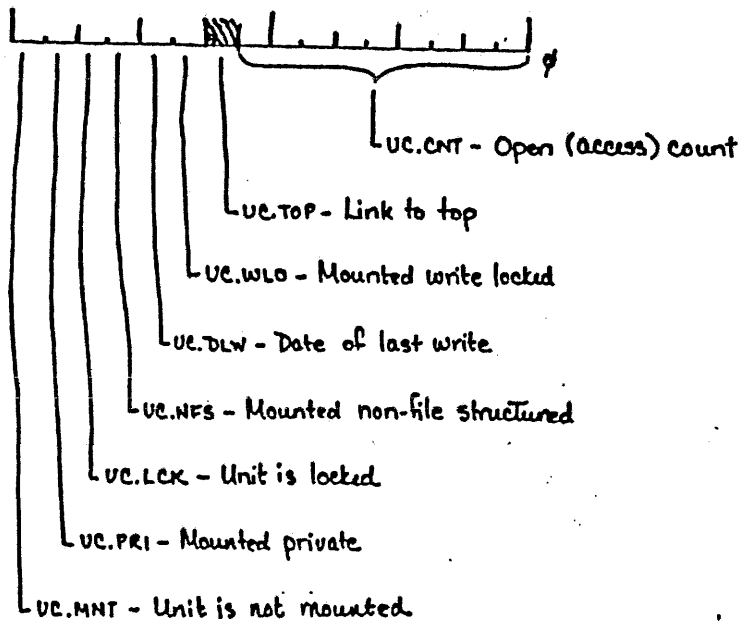


Figure MON-58. Unit Count Table (UNTCNT) Status Bits - V6C Change

Unit Options Table (UNTOPT)

- Byte table
- One entry for each disk unit configured on system
- Located at job owning table (UNTOWN) + 1

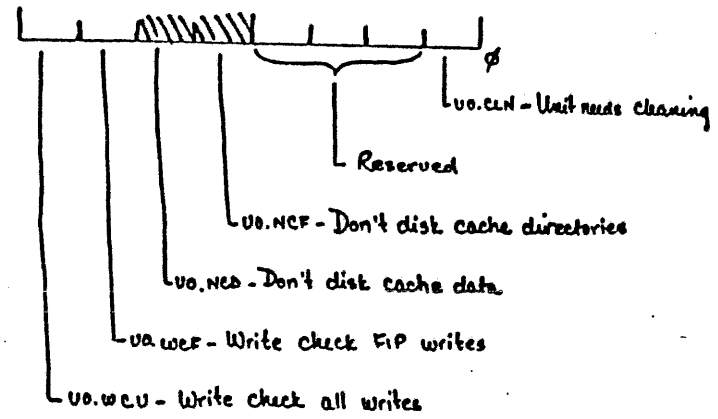


Figure MON-59. Unit Options Table (UNTOPT) Status Bits

Disk Indexes

- Used internally by disk subsystem
- Dynamically assigned at sysgen time

DSKMAP Table

- Word table of disk indexes
- Indexed by FUN *2

Name	Description
DSKDSP	Word dispatch table to dist initiate routines
DSKINT	Word dispatch table to interrupt service routines
DSKTMO	Word dispatch table to timeout service routines
DSKCSR	Pointers to CSRs
DSKPTR	Pointers to D?DINT @ DQS$00 - base queue root pointer

D?DINT Layout

Figure MON-60. Tables Indexed by Disk Index

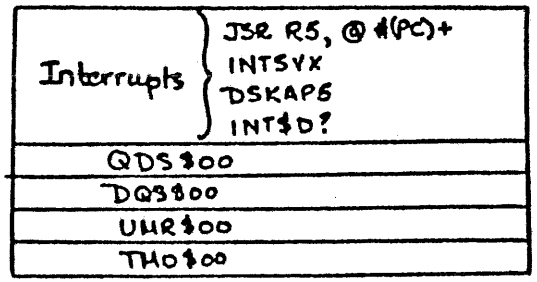


Figure MON-61. D?DINT Layout

QDS$00 - Entry for setting up DSQ
DQS$00 - Base queue root pointer
UMR$00 - Entry for UMR freed up
TMO$00 - Entry for time outs

Device Name Table (DEVNAM)

- Word table of ASCII device names
- One entry for each disk type supported by RSTS/E
- One entry for keyboards
- One entry for the null device
- One entry for pseudo keyboards
- One entry for every other device type sysgenned
- Entries for alternate names of sysgenned devices

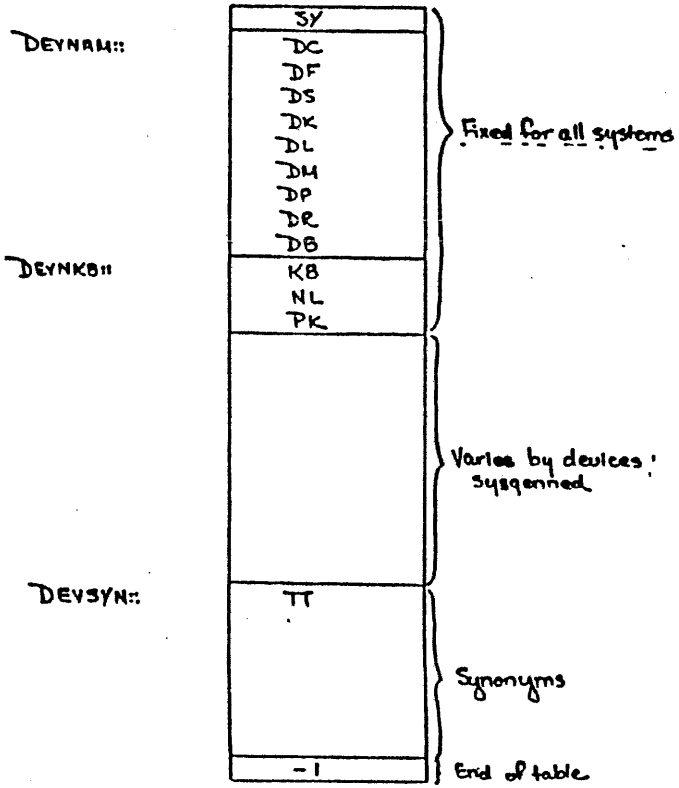


Figure MON-62. DEVNAM Table

Device Maximum Unit Number Table (DEVCNT)

- Word table of maximum unit numbers
- Length equals DEVNAM - 1
- Entries correspond to entries in DEVNAM table

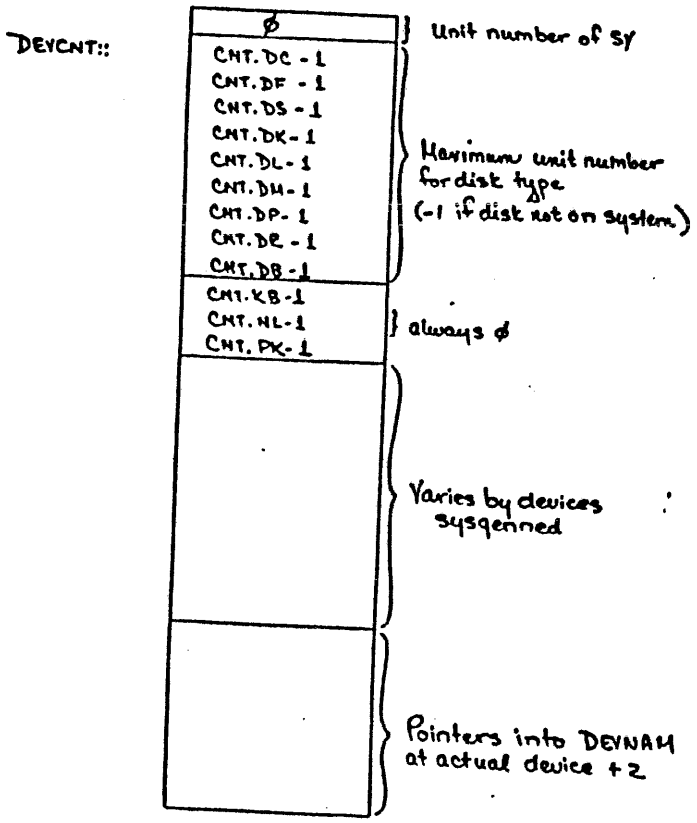


Figure MON-63. DEVCNT Table

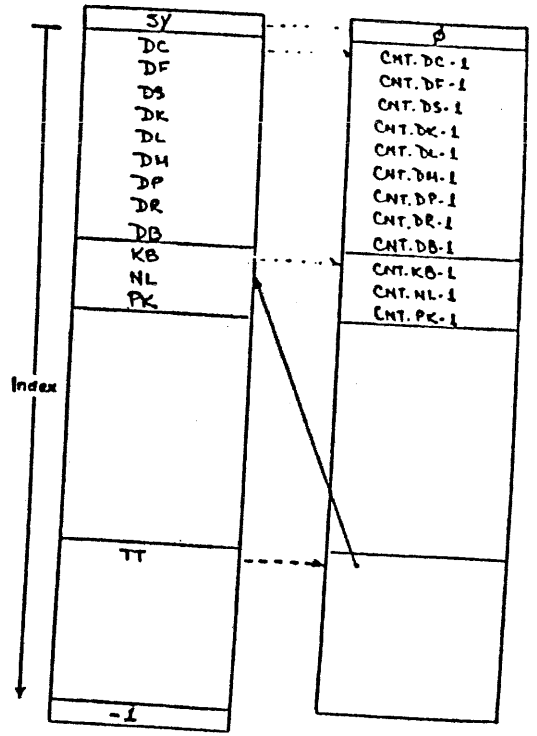


Figure MON-64. Indexing of DEVNAM and DEVCNT

Device Pointer Table (DEVPTR)

- Word table of pointers
- Length equals DEVCNT minus alternates
- Entries correspond to entries in DEVNAM table
- Disk entries point to unit 0 for each sygenned disk type in unit count table (UNTCNT)
- Non-disk entries point to unit 0 entries in the device table (DEVTBL)

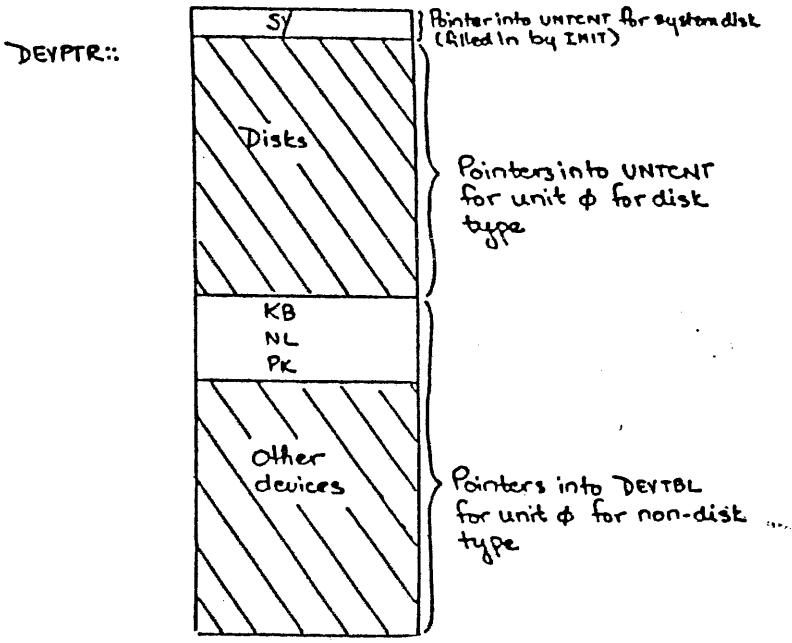


Figure MON-65. DEVPTR Table

DIGITAL RSTS/E SWS UPDATE
Monitor Internals

Device Table (DEVTBL)

- Word table of pointers to device data blocks (DDBs) for each non-disk device sygenned on the system.

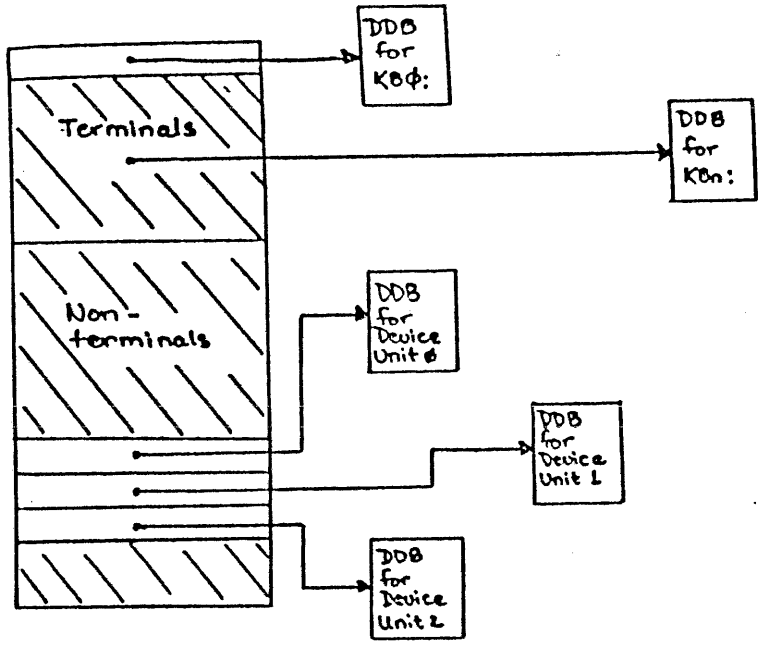


Figure MON-66. DEVTBL Table

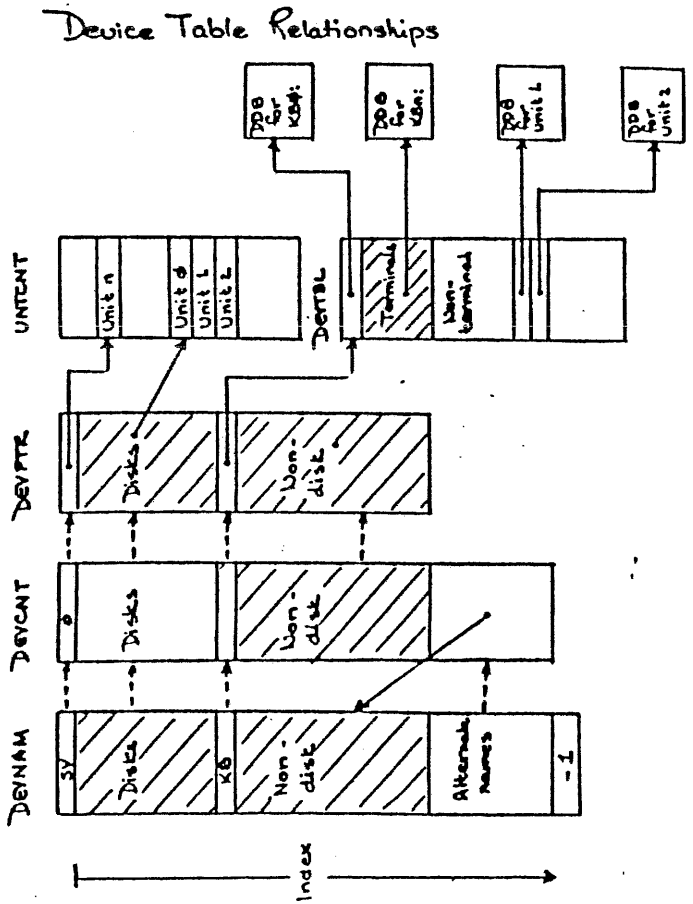


Figure MON-67. Device Table Relationships

Determining PIP Unit Numbers (PUN) for a System

1. From information obtained from DEVNAM and DEVCNT tables determine the devices and number of units actually sysgenned on the system.

   Note that the order of the devices in the DEVNAM table is important.

2. Based on this "order" PUN's are assigned starting with 0.

> **Note**
>
> Many tables referenced in this module and in the exercise need this information.

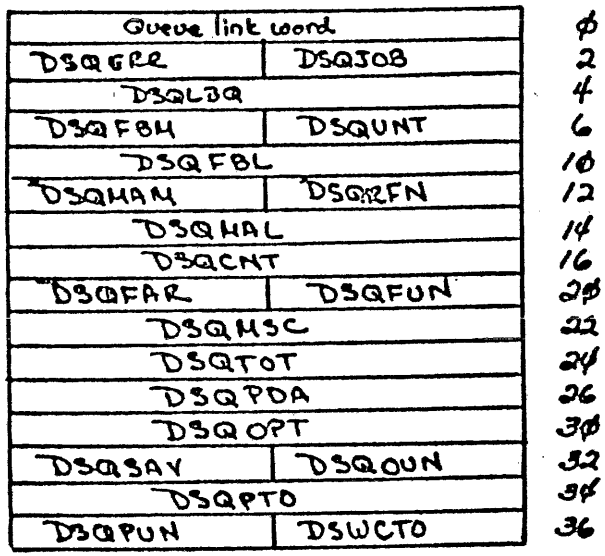


Figure MON-68. Disk Request Queue Block (DSQ)

DSQJOB Requesting job number *2
DSQERR Retry count and error flag
DSQL3Q Completion queue and L3Q bits to set
DSQUNT FIP unit number
DSQFBM FIP block number (MSB)
DSQFBL FIP block number (LSB)
DSQRFN RH11 function
DSQMAM Memory address of transfer (MSB)
DSQMAL Memory address of transfer (LSB)
DSQCNT Word count of transfer
DSQFUN Function code (WFUN.C=0-Write with write check disk, WFUN=2-Write disk, RFUN=4-Read disk, RFUN.C=6-Write check disk)

DSQFAR Queue fairness count
DSQMSC Miscellaneous pointer
DSQTOT Total transfer count
DSQPDA Physical disk address
DSQOPT Disk optimization word
DSQOUN Physical unit number *2
DSQSAV Saved function
DSQPTO Offset pointer
DSQCTO Offset retry count
DSQPUN Physical unit number

INTERFACING WITH THE MONITOR FROM USER MODE

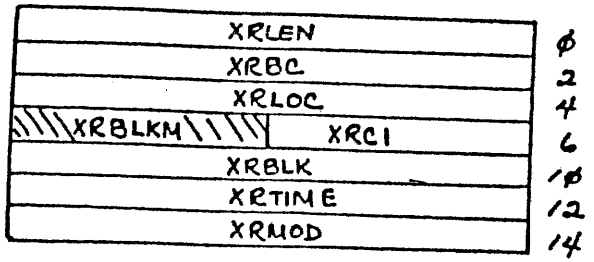


Figure MON-69. Transfer Request Queue Block (XRB)

XRLEN	Length of I/O buffer in bytes
XRBC	Byte count for transfer
XRLOC	Pointer to I/O buffer
XRCI	Channel number times two for transfer
XRBLKM	Random access block number (MSB)
XRBLK	Random access block number (LSB)
XRTIME	Wait time for terminal input
XRMOD	Modifiers

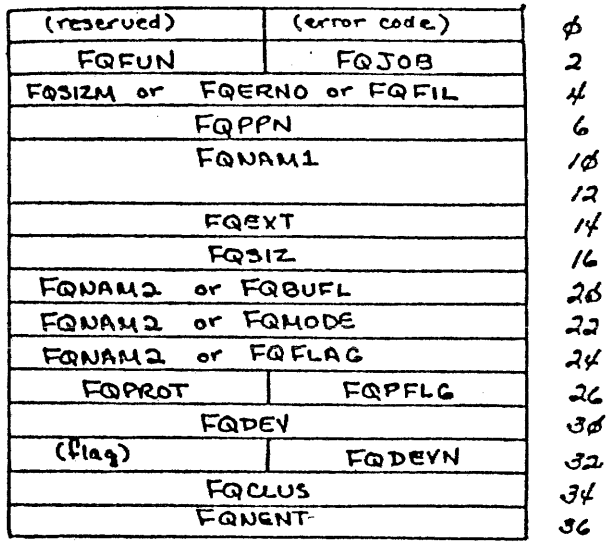


Figure MON-70. File Request Queue Block (FIRQB)

FQJOB	Job number times two
FQFUN	Function requested
FQERNO	Error message code and text begin
FQFIL (byte)	Channel number times two
FQSIZM (byte)	File size in blocks (MSB)
FQPPN	Project Programmer number
FQNAM1 (2 words)	File name in RAD50
FQEXT	Extension in RAD50
FQSIZ	File size in blocks (LSB)
FQNAM2 (2 words)	New filename and extension in RAD50

FQBUFL	Default buffer length
FQMODE	Mode indicator
FQFLAG	Opened file's flag word as returned
FQPFLG	'Protection code real' indicator
FQPROT	New protection code
FQDEV	ASCII device name
FQDEVN	Unit number
(flag)	'Unit number real' indicator
FQCLUS	File cluster size for file creates
FQNENT	Number of entries on directory lookup

New Monitor Calls (EMT's)

..PLAS	72	Resident library control
.RSX	74	Enter RSX emulation
.ULOG	76	ASSIGN/REASSIGN/DEASSIGN device/user logical

.PLAS (FIRQB @ FQERNO) Function Codes

ATRFQ	0	Attach region (resident library)
DTRFQ	2	Detach region (resident library)
CRAFQ	4	Create address window
ELAFQ	6	Eliminate address window
MAPFQ	10	Map address window
UMPFQ	12	Unmap address window

New .UUO/UUOFQ Subfunction Codes

UU.FIL	-26.	File utility
UU.CNV	20.	Convert date/time to ASCII (V6C)
UU.JOB	24.	Job creation (V6C)
UU.PPN	25.	Wild card PPN Lookup
UU.SYS	26.	Return job status
UU.KMC	27.	Connect KMC-11 to another device

Data Passed:
FIRQB

\ \ \ \ \ \ \ \ \ \ \ \		0
UU.JOB (=24.)	\ \ \ \ \ \	2
Must = 0	0 ⇒ Spawn 200 ⇒ Spawn regardless	4
Project number	Programmer No.	6
file name (2 words in RAD50)		10 12
file extension (RAD50)		14
10 bytes of any information. Placed in created job's core common		16 20 22 24 26
device name (2 ASCII chars.)		30
<>0 Unit no. real	device unit no.	32
Parameter word (same as .CCL)		34
\ \ \ \ \ \ \ \ \ \ \ \		36

Figure MON-71. UU.JOB - Job Creation

The Null (Disappearing) Run-time System

1. User runs an executable task image from a run-time system other than RSX.

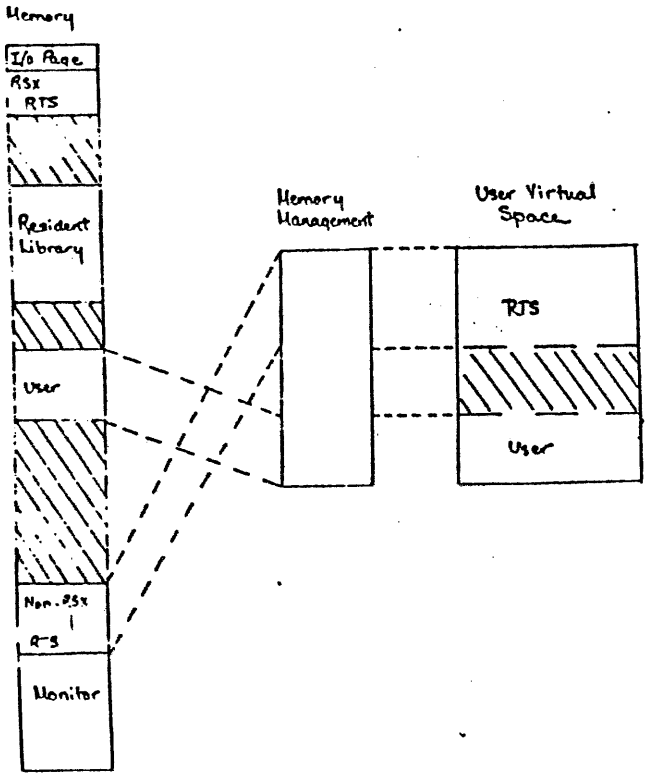


Figure MON-72.

2. Monitor remaps user space if necessary (loads run-time system if necessary).

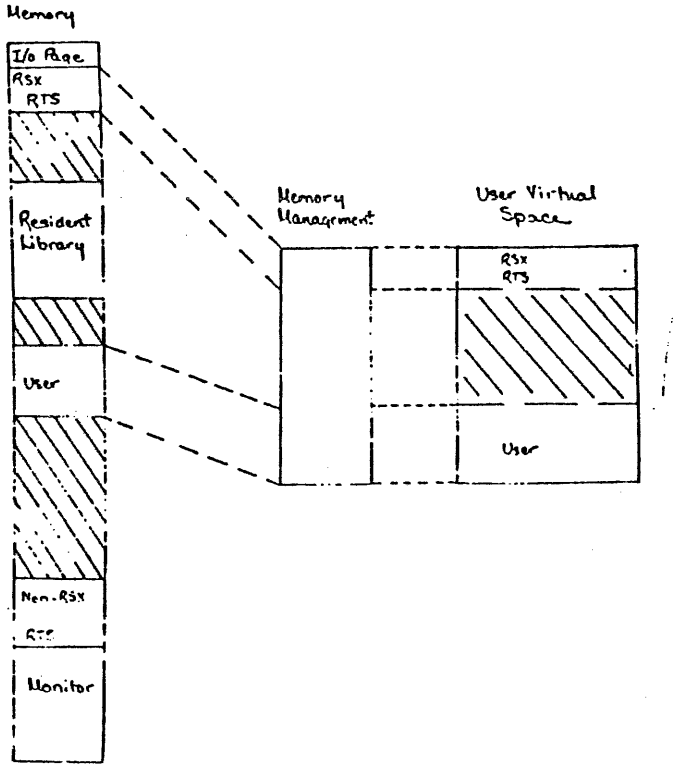
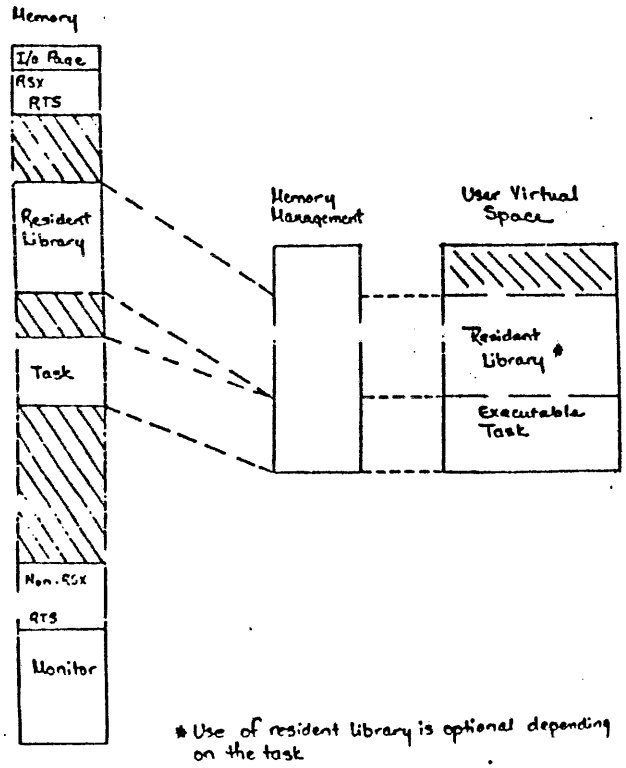


Figure MON-73.

3. RSX loads the executable task and issues the .RSX monitor call.



* Use of resident library is optional depending on the task

Location and size of resident library mapping varies based on APRs designated at task build time

Figure MON-74.

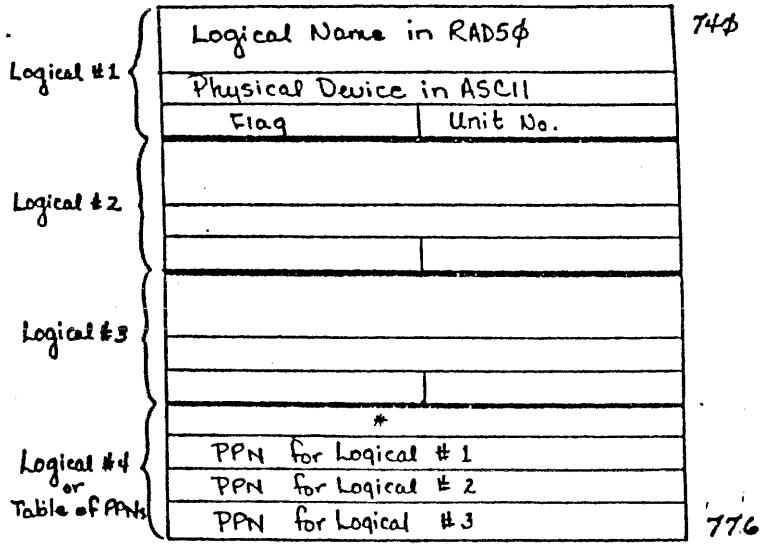


Figure MON-75. User Logical Table

* 0 Available for logical assignment
-1 Next three words for PPNs.
Anything else RAD50 for logical name

EXERCISE FOR MONITOR

Using the map and octal dump provided, answer the following questions.

1. Is 'JOB' attached to any resident libraries? _____

   If the answer is yes,

   a. How many libraries are attached? _____

   b. How many APRs are being used to map the library or libraries? _____

2. Have any resident libraries been added? _____

   If the answer is yes,

   a. How many resident libraries have been added? _____

   b. Answer the following questions about the library which is <u>first</u> in the list:

Resident library name _____

Its project, programmer number _____

Its size in Kwords (decimal) _____

Its protection code _____

Which resident library flags are set _____

_____

_____

_____

_____

_____

Load address of library (octal) _____

3. Are large files sysgenned on the system? _____

   If the answer is yes, go to question 5.

4. Considering the file opened on channel 3 for job number 21:

   a. Is the file opened for update? _____

   b. Is the file contiguous? _____

   c. Does the user have write privilege? _____

   d. What is the size (decimal) of the file? _____

e. What is the clustersize (decimal) of the file?

_____

f. Is data caching enabled for the file?

g. Is the data caching sequential or random?

_____

Go to question 6.

5. Considering the first file open on the first disk unit of the system:

a. What is the file's PPN?

_____

b. What is the file name and extension?

_____

c. Is the file placed?

_____

d. Has write access been given out?

_____

e. Is the file open for update?

_____

f. Is the file contiguous?

_____

g. Is file marked for deletion?

_____

h. What is the protection code?

_____

i. What is the access count for opens in normal or update mode?

_____

j. What is the clustersize?

_____

k. What is the octal address of the first WCB for the file?

_____

6. What disk types and how many units are sysgenned on this system?

_____

7. Considering DB2:

a. Is it mounted?

_____

b. Is it a date of last write disk?

_____

c. Are files added at the top of the directory?

_____

d. What is the current open count (octal)?

_____

e. What is the pack clustersize and error count?

_____

_____

8. What is the device designator of the system disk?

_____

Table MON-2. RAD 50

Single Char. or First Char.		Second Character		Third Character	
A	003100	A	000050	A	000001
B	003200	B	000120	B	000002
C	003300	C	000170	C	000003
D	003400	D	000240	D	000004
E	003500	E	000310	E	000005
F	003600	F	000360	F	000006
G	003700	G	000430	G	000007
H	004000	H	000500	H	000010
I	004100	I	000550	I	000011
J	004200	J	000620	J	000012
K	004300	K	000670	K	000013
L	004400	L	000740	L	000014
M	004500	M	001010	M	000015
N	004600	N	001060	N	000016
O	004700	O	001130	O	000017
P	005000	P	001200	P	000020
Q	005100	Q	001250	Q	000021
R	005200	R	001320	R	000022
S	005300	S	001370	S	000023
T	005400	T	001440	T	000024
U	005500	U	001510	U	000025
V	005600	V	001560	V	000026
W	005700	W	001630	W	000027
X	006000	X	001700	X	000030
Y	006100	Y	001750	Y	000031
Z	006200	Z	002020	Z	000032
$	006300	$	002070	$	000033
.	006400	.	002140	.	000034
unused	006500	unused	002210	unused	000035
0	006600	0	002260	0	000036
1	006700	1	002330	1	000037
2	007000	2	002400	2	000040
3	007100	3	002450	3	000041
4	007200	4	002520	4	000042
5	007300	5	002570	5	000043
6	007400	6	002640	6	000044
7	007500	7	002710	7	000045
8	007600	8	002760	8	000046
9	007700	9	003030	9	000047

DIGITAL

RSTS/E SWS UPDATE
Monitor Internals

This page is for notes.

MON-129 COMPANY CONFIDENTIAL

```
RT-11 Link  V05.05      Load Map                Wed 07-Feb-79 05:37:29
RSTS .SAV      Title    RSTS      Ident  V7.0-05  /PI:PR:PRO

Section  Addr   Size    Global  Value   Global  Value   Global

. ABS.  000000  000000  (RW,I,GBL,ABS,OVR)
                        $$CRT   000010  BADCLU  000027  BADCNT
                        BADMAP  000002  BADPAK  000032  BADSAT
                        BFD.NM  000012  BFD.LP  000024  BSE0C
                        BUF.LP  000200  BUF.PP  001000  BUF.SL
                        B.PRTY  000006  B.STAT  000008  B.SUBP
                        CLSIZT  177777  CLSTPP  177775  CLSISY
                        CNT.CR  000002  CNT.ES  000002  CNT.ET
                        CNT.PP  000003  CNT.PR  000017  CNT.SP
                        CPUPCF  000136  CPUPED  000137  CPUPFP
                        CPC.XX  176362  CDS.CT  000000  CDS.CX
                        CDS.NL  000010  CDS.PR  000710  CDS.SP
                        CEVCSR  000050  CFCSR   000000  CIPBC2
                        CLCSP   000000  CPOSP   000000  CSCSXX
                        CCP*CE  000157  CPCCAE  000162  CQE*CA
                        CCSERR  000150  CPLSLP  000000  CRRERR
                        FIELDS  000077  FIESST  000020  FIPERR
                        FLG,LP  035006  FLG,PP  000710  FLG,SL
                        FLTERR  000060  FPTERR  000062  FUCCLE
                        ICX,CT  000016  ICX,CX  000020  ICX,NB
                        ICX,PR  000004  ICX,SP  000000  ICX,SP
                        INUSE   000003  ICLERR  000070  JOB*SZ
                        JS,CX   000001  JS,PP   000002  JS,LP
                        JS,SP   000001  JS,SP   000710  LINECR
                        LCGSAP  100200  LCGSLP  100050  LCGSPP
                        MAGSEL  000007  MAXUNI  000012  MAXUN2
                        NCCERR  000073  NCCUFS  000200  NCCEVC
                        NCMACS  000100  NCRCCP  000000  NCPTS
                        NCTPAT  000025  NCTPTA  000101  NCVCPA
                        CNBAC   000072  CPASCT  177777  CPASPP
                        PANLCN  000026  PNIANT  000013  PN*CCE
                        PRNER2  000170  PNVICL  000012  QDTONE
                        QP*CCN  100000  QP*QAE  100010  QPQONE
                        SASPAE  000102  SATTPC  000035  SCAERR
                        SIZ,NB  000000  SIZ,LP  000200  SIZ,PP
                        SIZ,SP  000107  SLREAD  000066  SIPERR
                        STS,NB  000000  STS,LP  000002  STS,PP
                        ST2701  000000  SUBERR  000067  SYSLOG
                        TLIFFE  000117  TLINCC  000132  TLJAPP
                        TLNCTF  000121  TLNZSP  000115  TLCPAY
                        TLT*OF  000131  TLUPTP  000105  TLXCIP
                        TTCJ11  000012  TTCL11  000000  TTC*RR
                        TTPL11  000000  TT*SCA  000000  TTPARR
                        UNCFAI  000112  VCAERR  000053  VCCERR
                        VCCCIA  000000  XOFCIA  000000  XOLOIA
                        Y12700  000000
```

Figure MON-76. RSTS.MAP


```
Value   Global  Value   Global  Value   Global  Value

000037  PAFCIA  000001  PACERR  000171  PADPUD  000022
000103  PCNERR  000000  PFE.XX  000006  PF1.NB  000006
000056  PUF.CT  000776  PUF.CX  001000  PUF.NB  000200
000002  PUF.PP  000000  PUF.SY  001000  PUF.XX  001000
000005  P.IP    000002  P.SSR   000003  R.0     000001
000001  SAT.CP  000003  SAT.CR  000002  SAT.CX  000002
000002  SAT.CX  000002  SAT.NB  000130  SAT.LP  000021
000006  SCSERR  000113  SPASOF  000133  SPUFAX  000135
000130  STPLCE  000036  SATERR  000015  SCDS<   000000
000036  SDS,PE  000006  SDS.LP  000000  SDS.PP  000000
000060  SEINEX  000033  SEVXFS  000036  SEVCHS  000022
000151  SIP*ER  000170  SISERR  000172  SIVMTA  000075
176502  SSTPUL  000200  STCCCP  000021  SDA*SV  000103
000160  SYCCTP  000006  SYCPAT  000007  EOF     000013
000102  SYITAS  000111  SYITTP  000110  SXPERR  000001
000003  TLG,CT  000200  TLG,CX  002022  TLG,NB  000002
000020  TLG,PP  000020  TLG,SY  100000  TLG,XX  000030
000107  TLNERR  000130  TNCCEV  000010  TNGTTY  000017
000002  TCX,LP  000012  TCX,PP  000010  TDX,NL  000200
000010  IE      000102  INTLCP  000023  INTPAR  000031
000077  JCP*12  000176  JSALL   000036  JS,DT   000001
000022  JS,PP   000001  JS,AL   000001  JS,PR   000020
000057  LCGERR  000065  LCGSDT  100050  LUGSDX  100050
100052  LCGSPP  100006  LRGFIL  000001  MAGRLE  000050
000020  MINTER  000077  MCCCLN  000052  MDCERR  000125
000006  NCGCTC  000153  NCPATH  000175  NDNCIM  000107
000076  NCSUCP  000005  NCTAVL  000012  NDTCLS  000007
000011  NRNERR  000161  NNRTS   000051  ODC     000071
177775  OPASSY  000001  OPCSPP  000026  OUTCAS  000127
000012  PHERRS  000160  PRIVAT  000032  PRNER1  000100
100002  QCTPIP  100020  QCTPES  100100  QDPCCN  100000
100002  RESERV  000150  SAFNCS  000103  SASSCI  000100
000177  SIZERR  000050  SIZ,DT  000107  SIZ,DX  000107
000107  SIZ,NL  000107  SIZ,PR  000107  SIZ,SY  000107
000107  SIPERR  000173  STS,DT  000000  STS,CX  000000
000000  STS,NL  000000  STS,PP  000000  STS,SP  000000
000001  TLCNTC  000155  TLCCNI  000120  TLICIP  000152
000100  TLMFAC  000123  TLNCPA  000101  TLNCIT  000116
000110  TLPPAP  000156  TLCDUP  000122  TLPPAP  000120
000106  TTCC11  000222  TTCL11  000006  TTDP11  000010
000200  TTDZ11  000010  TTFCNT  000063  TTINEC  000000
000052  TTPK11  000010  TTPING  000050  UDMERR  000105
000055  VCSERR  000052  WRGMAN  000024  XCOCCP  000176
000000  XOFCIA  000000  XOCCSZ  000020  SSTPLN  000010
```

Figure MON-76. RSTS.MAP (Cont.)

```
[illegible]  [illegible]  [illegible]  (RW,I,GBL,REL,CON)
                              [illegible] [illegible] FIJBC[illegible] [illegible] FIJBC2
                              FISTA[illegible] [illegible] FILSE[illegible] [illegible] HALT
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
BUFPCL [illegible] [illegible] (RW,I,GBL,REL,CON)
                              BUFPCL [illegible] [illegible] [illegible]
ICBUF[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible]

[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] LPECTL [illegible] [illegible]
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible] DEVCLU [illegible] [illegible]
                              DEV,[illegible] [illegible] DEV,[illegible] [illegible] DEV,[illegible]
                              DEV,[illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              JPSTAT [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              SATEN[illegible] [illegible] [illegible] [illegible] SATSTL
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              [illegible] [illegible] [illegible] [illegible] [illegible]
                              UNTC[illegible] [illegible] UNTE[illegible] [illegible] UNTL[illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,OVR)
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible]
[illegible] [illegible] [illegible] (RW,I,GBL,REL,CON)
                              [illegible] [illegible]
```

Figure MON-76. RSTS.MAP (Cont.)


```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]  DEV,[illegible]  [illegible]  DEV,[illegible]  [illegible]  DEV,CT  [illegible]
[illegible]  DEV,[illegible]  [illegible]  DEV,[illegible]  [illegible]  DEV,[illegible]  [illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]  [illegible]  [illegible]  LI[illegible]LST  [illegible]  [illegible]  [illegible]
[illegible]
[illegible]
[illegible]  [illegible]  [illegible]  SNCLST  [illegible]  SPLLST  [illegible]
[illegible]
[illegible]  [illegible]  [illegible]  UNTCLU  [illegible]  [illegible]  [illegible]
[illegible]  UNTCPT  [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
```

Figure MON-76. RSTS.MAP (Cont.)

```
CXCCCP  065330  000070   (R=,I,GBL,REL,OVR)
                         CXCCCP  065330
CSCCTL  065422  000006   (R=,I,GBL,REL,OVR)
                         CSCCTL  065420  CSCPT1  065422  CSCPT2
CKCCTL  065426  000006   (R=,I,GBL,REL,OVR)
                         CKCCTL  065426  CKCPT1  065432  CKCPT2
CMCCTL  065430  000006   (R=,I,GBL,REL,OVR)
                         CMCCTL  065430  CMCPT1  065436  CMCPT2
CRCCTL  065442  000006   (R=,I,GBL,REL,OVR)
                         CRCCTL  065442  CRCPT1  065444  CRCPT2
CBCCTL  065450  000006   (R=,I,GBL,REL,OVR)
                         CBCCTL  065450  CBCPT1  065452  CBCPT2
SYCCTL  065454  000002   (R=,I,GBL,REL,OVR)
                         SYCCTL  065456
KBCCTL  065460  000056   (R=,I,GBL,REL,OVR)
                         JCBCNT  065460  KBCCTL  065460  KBEOCB
                         KBETBL  065470  KBETMP  065530  KBEUNT
                         TTYHCT  065466  $$JCBM  065462
NLOCTL  065536  000000   (R=,I,GBL,REL,OVR)
PKCCTL  065536  000006   (R=,I,GBL,REL,OVR)
                         PKCCTL  065536
XMCCTL  065544  000006   (R=,I,GBL,REL,OVR)
                         NSPQUE  065540  XMBSTC  065544  XMCCTL
LPCCTL  065552  000000   (R=,I,GBL,REL,OVR)
MMCCTL  065552  000010   (R=,I,GBL,REL,OVR)
                         MMCCTL  065552
CTCCTL  065566  000006   (R=,I,GBL,REL,OVR)
                         CTCCTL  065566
DXCCTL  065570  000030   (R=,I,GBL,REL,OVR)
                         DXCCTL  065570
RTICTL  065620  000076   (R=,I,GBL,REL,OVR)
                         CTRLEV  065632  FTLVEC  065620  MGHAD
                         MEMERR  065640  PARZER  065672  RTICTL
                         WAIT2T  065710  $$CCTL  065632  $$PZER
BUFCTL  065722  000030   (R=,I,GBL,REL,OVR)
                         BAKLRG  065740  BAKSML  065736  BUFCTL
                         RESTRX  065752  $$FRPP  065720
RESCTL  065756  000030   (R=,I,GBL,REL,OVR)
                         CCLLST  065770  FRPFEC  065756  JOBCNT
                         MEMADR  066002  P$NE=   066000  QUANT
                         S-PASE  066000  S-DCNQ  066500  S-CCNT
                         B-PUSE  066006  S-TPNE  066000  S-TBLK
ERLCTL  066612  000010   (R=,I,GBL,REL,OVR)
                         ERLCTL  066612
CHECTL  066626  000230   (R=,I,GBL,REL,OVR)
                         CHECTL  066626  CHETIC  066630  $$CACH
FIPCTL  067062  000120   (R=,I,GBL,REL,OVR)
                         ERBASU  067060  ERSIZE  067062  FIBERN
                         FIPCLU  067110  FIPCTL  067062  FIPERN
                         FIPR6   067112  FIPSP   067110  FIPUNT
                         H       067165  CVBASU  067076  SATERN
                         $$CVRC  067070
```

Figure MON-76. RSTS.MAP (Cont.)


```
065420
065432
065442
065446
065450
065500  KBEREG  065510  KBESCN  065516  KBESC4  065520
065530  LMTCNT  065600  MAXCNT  065602  MAXJCB  065601
065500
065636  LCHAD   065630  L3CUE   065716  L3CUE2  065720
065620  UPELGE  065666  UPELCG  065602  UPELSZ  065666
065672
065722  EXTSTL  065700  FREES   065722  MONSTL  065702
065776  JLNK    066000  MAXMGH  065760  MAXLC=  065762
065770  RESCTL  065756  RESLST  065766  SCHTBL  066002
065772  S=PCSC  066506  S=PMAP  066506  S=PMPE  066606
066002  $$PNE=  066000  $$S=MP  066506  $$S=PB  066000
066636  $$CPSZ  066656  $$CHEN  066626
067176  FIPMCC  067203  FIPUNT  067202  FIPAR   067122
067162  FIPELN  067106  FIPGEN  067205  FIPMUC  067116
067220  FIPL2C  067120  FIPVBN  067174  J       067164
067166  SATMCC  067173  SATUNT  067172  $$EPSZ  067062
```

Figure MON-76. RSTS.MAP (Cont.)

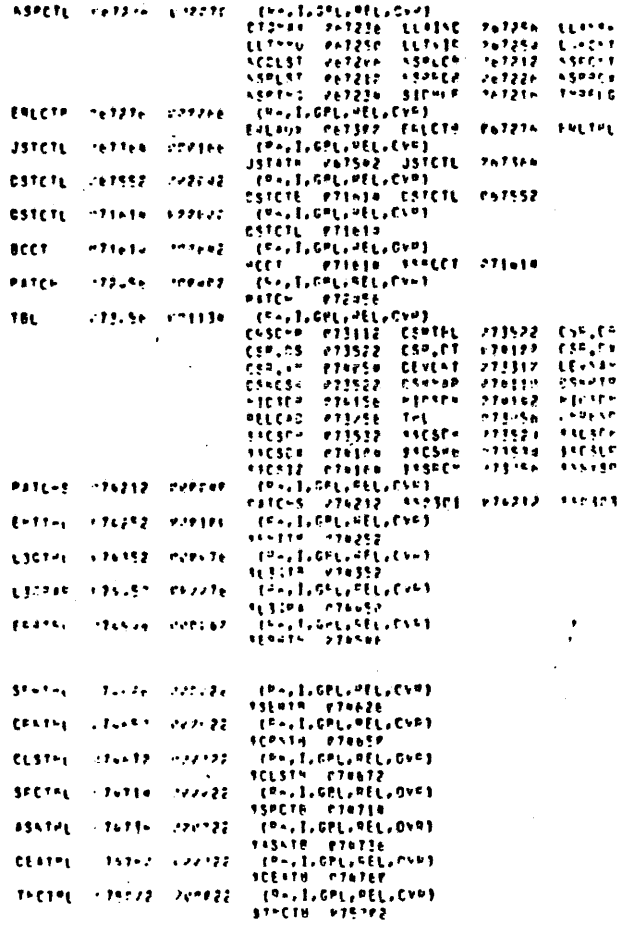


Figure MON-76. RSTS.MAP (Cont.)

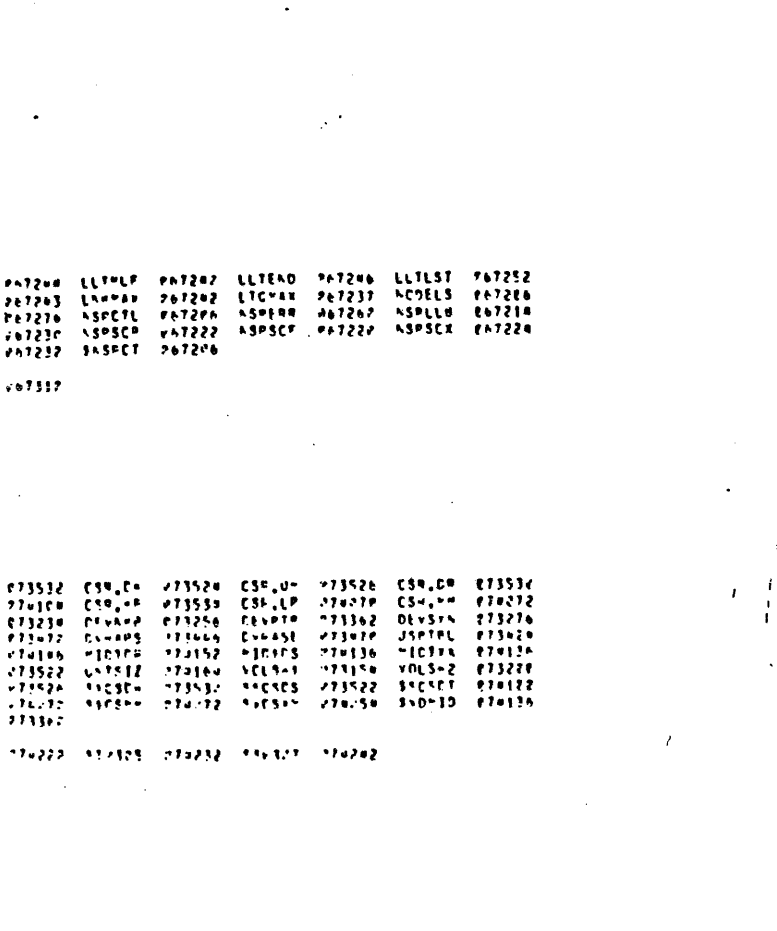


Figure MON-76. RSTS.MAP (Cont.)

```
$LPTPL  ·7···2·  ····22   (R=,I,GPL,REL,OVR)
                          $SLPTP  #75P2#
FLGTPL  ·7···0·  ····22   (R=,I,GPL,REL,OVR)
                          $FLGTP  #75···
SIZTPL  ·7···7·  ····22   (R=,I,GBL,REL,OVR)
                          $SIZTP  #75P7P
PLPTPL  ·7·11·  ····22    (R=,I,GPL,REL,OVR)
                          $PLPTP  #75112
PSATPL  ·7513·  ····0·    (R=,I,GPL,REL,OVR)
                          $PSATP  #75130
PIPTPL  ·7517·  ····50    (R=,I,GPL,REL,OVR)
                          $PIPTP  #75170
UUCTPL  ·752··  ····50    (R=,I,GBL,REL,OVR)
                          $UUCTP  #75250
UUCSAP  ·75·2·  ····6·    (R=,I,GBL,REL,OVR)
                          $UUCSA  #75·2·
STSTPL  ·7551·  ····1·    (R=,I,GPL,REL,OVR)
                          $TSTAL  #75512
TERCSP  ·75···  ····1·    (R=,I,GBL,REL,OVR)
                          CLRTER  #7552·  CLRTIA  #7553P  CHFJCP
                          $TERCS  #75522
CSHCSP  #7553·  ····2     (R=,I,GBL,REL,OVR)
                          GCCISK  #7553·  $CSHCS  #75536
EPTCSP  #755··  ····2     (R=,I,GBL,REL,OVR)
                          CCPCST  #7550P  $EPTCS  #7550P
PIPCSP  #75502  ····7·    (R=,I,GBL,REL,OVR)
                          PIPERR  #7550·  PIPCC   #75502  CVRPLP
RSXCSP  #7555P  ·2···     (R=,I,GBL,REL,OVR)
                          RSXAGN  #75552  RSXEAT  #7555·  RSXTRP
PTACSP  ·7555·  ····1·    (R=,I,GBL,REL,OVR)
                          $PTADS  #7550·
CTACSP  ·7557·  ····1·    (R=,I,GPL,REL,OVR)
                          $CTACS  #7557·  $SCRSZ  #75A1·
CSCINT  ·7561·  ·C··2·    (R=,I,GBL,REL,CYR)
                          CSCINT  #7561·  $CSDIA  #7561·
CMCINT  #7563·  ····2·    (R=,I,GBL,REL,OVR)
                          CMCINT  #7563·  $DMCIA  #7563·
CPCINT  #7565·  ····2·    (R=,I,GBL,REL,CYR)
                          CMCINT  #7565·  $DMCIA  #7565·
DRCINT  ·7567·  ····2·    (R=,I,GPL,REL,OVR)
                          DRDINT  #7567·  $FMCIA  #7567·
DECINT  ·7571·  ····2·    (R=,I,GBL,REL,OVR)
                          DMCINT  #7571·  $DBCIA  #7571·
SYDINT  #7573·  ··2·      (R=,I,GBL,REL,OVR)
                          FILGO   #76322  FILI·   #75742  FILSET
                          ,,BOAT  #7615·
MSCINT  #76636  ····52    (R=,I,GBL,REL,OVR)
                          CMMPA   #7673·  CMMINT  #76656  DZMMA
                          MBOAPS  #76602  MMFOCM  #7675·  XLMINT
                          $ICCTP  #7676·
ALCINT  #77·1·  ····      (R=,I,GBL,REL,OVR)
PMCINT  #77·1·  ····      (R=,I,GPL,REL,OVR)
                          PMCINT  #7731·
```

Figure MON-76. RSTS.MAP (Cont.)


```
#75532  ENDMEY  #75522  SETCTP  #7553·  SETPM·  #7552·

#7550·  $PIPCS  #75502

#7555P  $MSXCS  #7555P

#76676  FLOAE   #7573·  $YDAPS  #7570P  SYDINT  #7573·

#767··  DZMINT  #76676  ENTCCT  #76760  EPLSHB  #7673P
#76636  PTABLE  #76766  XCFFCM  #77P66  SHBOIN  #76636
```

Figure MON-76. RSTS.MAP (Cont.)

```
**CINT  177710  000156  (R=,I,GBL,REL,OVR)
                        ASSIGN  177700  CLOSE=  177706  CREATE
                        DCP3LE  177110  PPFLCE  177132  SE=INI
                        X=ESJP  177104  X=DAPS  177214  X=CINT
LPCINT  177166  000110  (R=,I,GBL,REL,OVR)
                        LPDAPS  177172  LPCINT  177166
**CINT  177176  000010  (R=,I,GBL,REL,OVR)
                        **DAPS  177202  **CINT  177176
DTCINT  177212  000120  (R=,I,GBL,REL,OVR)
                        DTDAPS  177216  DTCINT  177212
DXCINT  177232  000110  (R=,I,GBL,REL,OVR)
                        DXDAPS  177236  DXDINT  177232
ASPSLR  177262  000000  (R=,I,GBL,REL,OVR)
                        ASPAPS  177266  ASPSER  177262  ?ASPSL
ALLJCB  177246  000466  (R=,I,GBL,REL,OVR)
                        ALLJCB  177246
ASPCH=  177732  000002  (R=,I,GBL,REL,OVR)
                        ASPCH=  177732
ASPTPC  177770  000022  (R=,I,GBL,REL,OVR)
                        ASPT=C  177770
RJETPC  1=0016  000002  (R=,I,GBL,REL,OVR)
                        RJETPD  100016
PRCV=   1=0020  000000  (R=,I,GBL,=EL,OVR)
                        P4C=R   100020
*RCV=   1=0020  000002  (R=,I,GBL,REL,CVR)
                        *RCV=   100020
*EC     1=0020  000000  (R=,I,GBL,REL,OVR)
                        *EC     100020
LPE     1=0020  000000  (R=,I,GBL,REL,OVR)
                        LPE     100020
FPP     1=0020  000000  (R=,I,GBL,REL,CVR)
                        FPP     100020
RTI     1=0020  000000  (R=,I,GBL,REL,CVR)
                        PLFFER  122012  CLO=GC  120250  CLC=,L
                        ERRCA   121110  EXTPCF  122700  EXTPCL
                        FTLXX2  120110  FTLXX1  120122  FTLXX2
                        FTLXX6  120106  FTLXX7  120152  FTL=20
                        ICREIT  121106  ICREFC  121102  LNGPCL
                        RESQT=  120540  RETPIG  121002  RET=CA
                        RSXE=1  101250  RTI     100020  RTI3
                        SYIT=C  120302  TRYEXT  121004  TNT=CA
                        SICP0=  120244  SICCNT  120322  SICSET
                        SSFP02  122700  SSFP25  122052  SYLCAT
                        SICCTE  100230  SICCT0  100224  SIPSCA
                        SSVSAR  101022  ..P=RF  102310
C=ES=L  1=5104  000000  (R=,I,GBL,REL,OVR)
                        C=ES=L  105104
C=E=CA  1=5110  000000  (R=,I,GBL,REL,OVR)
                        C=E=CA  105110
C=EEXT  1=5114  000210  (R=,I,GBL,REL,OVR)
                        C=EEXT  105114
C=A=CA  1=5332  000032  (R=,I,GBL,REL,OVR)
                        C=A=CA  105332
```

Figure MON-76. RSTS.MAP (Cont.)


```
177262
```

```
120312  CLC=,P  120266  C=PJCB  101210  E=TAPS  101030
122730  FREF2=  122=06  FREE2S  122550  FTLSCA  103660
120120  FTLXX3  120132  FTLXX4  100136  FTLXXS  120142
120112  GET=IC  101722  INTSAS  1=0500  100074
122232  =CA=CL  122=0=  P=RSCA  103602  PEE=ER  105052
101150  RETUSR  101054  RETLSR  101400  RSXAPS  101030
120570  RTIQT   1=0050  SETRES  102470  SETUP   103060
122732  LRC=ER  105=00  SICCNT  100310  SICCSR  12=202
120306  SYRETC  122200  SYERTP  122036  SYFIS   100274
100200  SYLSGP  122032  SYNCAT  100052  SYCOTE  120710
103AP0  SYPTFC  121072  SYP=AP  103372  SYTICK  100410
```

Figure MON-76. RSTS.MAP (Cont.)

```
C++++++  1:0764  1:7567  (R+,I,GBL,REL,PAR)
C++EAT  1:93E4
REB     1:4000  1:7632  (R+,I,GBL,REL,OVR)
                        +RELST  1:6472  RF9A9E  1:8522  EALL++
                        CALL**  1:9574  CATCAY  1:5572  CIA7E+
                        FIPAPS  1:5A1F  FI*SYS  1:8E2E  P*CAE
                        CP*CLS  1:5A52  CYRA9F  1:9510  ***
                        SE-SPC  1:E0F3  STELPT  1:5A5A  S-**
                        SYSVA2  1:6005  TI*CAY  1:5596  **A0EL
                        **FJC0  1:5516  ***ILP  1:5A7E  91*22-
                        **SL    1:5A5E  915SEG  1:5AE2  A15-A-
                        ..RSAT  1:5552  ..*ICP  1:5A7A
**A**L  1:A57E  1:E0F1F  (R+,I,GBL,REL,OVR)
                        **ATBL  1:E07E  ..***.  1:E07A
CC**S-  1:61E0  1:F0A0  (R+,I,GBL,REL,OVR)
                        CC*CSA  1:61F0  CISA    1:61F6  DSKAPS
CHECK*  1:6152  1:7216  (R+,I,GBL,REL,OVR)
                        CHECKA  1:6152
CHEZAP  1:6372  1:F0F1F  (R+,I,GBL,REL,OVR)
                        CHEZAP  1:E37F
CHEULC  1:E0F0  1:F021F  (R+,I,GBL,REL,OVR)
                        CHEUUO  1:E0FF
ERL     1:A01F  1:51612  (R+,I,GBL,REL,OVR)
                        CLRCHE  1:7730  ERL     1:A01A  ERL*SC
SL*     1:1222  1:A0/FF  (R+,I,GBL,REL,OVR)
                        ACCMEM  1:1250  BADBTS  1:1246  BITBYT
                        CCLCHK  1:1372  CHKSTR  1:1F06  CHKXFR
                        CNVCC3  1:1200  CCNAEC  1:1754  CONNEC
                        FAC*SO  1:1356  FADBTS  1:1526  FREELF
                        ICFINI  1:1302  ICFINA  1:1306  LCCPCK
                        *INUS1  1:13FF  *CVBYT  1:1F26  *OV*RD
                        REGRES  1:15A0  REGSAV  1:1552  *EGSCR
                        RESEAT  1:1336  RESJCB  1:1312  RETDEV
                        SAVJCB  1:15AF  SAVTI*  1:166F  SCRUPR
                        SL*     1:1222  S*RTB   1:1267F  SARTSL
                        UNCCHK  1:1276  ZERC    1:1F76  ***ZCV
STCHE   1:1222  1:F0F72  (R+,I,GBL,REL,OVR)
                        STCHE   1:1222
FETCH   1:131A  1:1F23A  (R+,I,GBL,REL,OVR)
                        FETCH   1:131A  GETJBR  1:151A
CLRBUF  1:1552  1:F0182  (R+,I,GBL,REL,OVR)
                        CLRBUF  1:1552
CHACVR  1:1716  1:F017F  (R+,I,GBL,REL,OVR)
                        CHACVR  1:1716  ERLCHA  1:171A  GETUPR
                        SIRUPR  1:151E2
BSTALL  1:151A  1:F0E2A  (R+,I,GBL,REL,OVR)
                        BSTALL  1:151FA
ERLCVR  1:1517E  1:F01FA  (R+,I,GBL,REL,OVR)
                        ERLCVR  1:1513E
JCSTAT  1:1523A  1:F1E2A  (R+,I,GBL,REL,OVR)
                        JCSTAT  1:1523A
CSTAT   1:1628E  1:E22E  (R+,I,GBL,REL,OVR)
                        CSTAT   1:1628E
SYCVR   1:1666  1:F1272  (R+,I,GBL,REL,CVR)
                        SERISY  1:1666  SYDVR   1:1666  SIEP**
```

Figure MON-76. RSTS.MAP (Cont.)

RSTS/E SWS UPDATE
Monitor Internals

```
1:61A4
                                                        11F112  *1ECPL  1:7732  *1E*EL  1:7736  *1ESID  1:773F
                                                        111286  PIT*AC  111222  EL**CV  113610  BYEPTS  11276F
                                                        111748  CL**SG  113366  CLUCVT  112206  CAVDCA  11243A
                                                        111754  PI54EA  11*6A3  CI5A*E  110656  FADJCB  113762
                                                        11F362  GETCEV  112154  GUNSUP  112F32  IOFINC  111316
                                                        11F6A8  L3USET  111172  L3GSEP  111166  MAPBUF  11017A
                                                        111F7A  CLEL3G  11F7FA  CUEUEA  11F622  QUEUEB  110652
                                                        112562  REPJCB  113FFF  REP*TS  112748  RESCH*  113276
                                                        112226  A15PC   111122  SAVJAL  1116*E  SAVJCB  111*EA
                                                        111236  SCRUP*  111232  SIG**E  113372  ST*CVF  111216
                                                        112674  TAPSET  11F3F2  UNLCC*  113F72  VNLCC3  113076
                                                        111746  ***CV2  114C26  ***CV*  110F7A  3SRTSP  111122
                                                        1151FE  *ELUP*  1151F2  PHP*DR  115016  *1GUP*  1151EF
                                                        11F4E6  ..CP**  117572  ..3YDF  116654
```

Figure MON-76. RSTS.MAP (Cont.)

```
SYDLRG  117760  000000   (R=,I,GBL,REL,OVR)
                         SYDLRG  117760
UPDCHK  117760  000330   (R=,I,GBL,REL,OVR)
                         UPDCHK  117760
UPDUNL  120310  000500   (R=,I,GBL,REL,OVR)
                         SPC$ST  120310  UPDUNL  120310  ..RMCB
NLCVR   121020  000032   (R=,I,GBL,REL,OVR)
                         ASN$NL  121022  CLS$NL  121024  DEA$NL
                         SER$NL  121026  SPC$NL  121220  TMC$NL
LPCVRX  121052  000000   (R=,I,GBL,REL,OVR)
                         LPDVRX  121052
LPCVR   121052  002030   (R=,I,GBL,REL,OVR)
                         ASN$LP  121030  CLS$LP  122260  DEA$LP
                         SER$LP  121050  SPC$LP  121252  TMC$LP
DTCVR   123506  001530   (R=,I,GBL,REL,OVR)
                         ASN$DT  123510  DEA$DT  123502  CNE$DT
                         INT$DT  123500  RES$DT  124736  SER$DT
MMCVR   125236  003210   (R=,I,GBL,REL,OVR)
                         ASN$MM  125236  CCN$MM  126150  DEA$MM
                         MMCVR   125236  SER$MM  126030  SPC$MM
XMCVR   130050  000000   (R=,I,GBL,REL,OVR)
                         XMDVR   130050
DXCVR   130050  003210   (R=,I,GBL,REL,OVR)
                         ASN$DX  131030  CLS$DX  131252  CCN$DX
                         INT$DX  131032  CPN$DX  131152  SER$DX
                         ..DXIT  133460
BUFEXT  133670  000000   (R=,I,GBL,REL,OVR)
                         BUFEXT  133670
BUF     133670  003500   (R=,I,GBL,REL,OVR)
                         BUF     133670  CHECKX  134000  ..CAGE
LIGHTS  137376  000000   (R=,I,GBL,REL,OVR)
                         LIGHTS  137376
UPDATE  137376  000000   (R=,I,GBL,REL,OVR)
                         UPDATE  137376
UPDLRG  137376  000000   (R=,I,GBL,REL,OVR)
                         UPDLRG  137376
IRB     137376  000002   (R=,I,GBL,REL,OVR)
                         GETIRX  137570  IRB     137376  RETIRB
```

Figure MON-76. RSTS.MAP (Cont.)


```
                                                                  1
                                                                  120002
                                                                  121020  INT$NL  121020  NLCVR   121020  OPN$NL  121020
                                                                  121020
                                                                  121000  INT$LP  123232  LPCVR   121052  OPN$LP  121250
                                                                  122160  ..LP60  121350
                                                                  120202  DTCVR   123506  ERL$DT  124170  FIP$DT  125102
                                                                  120200  SPC$DT  123506  TMC$DT  124150
                                                                  125266  CNE$MM  126160  ERL$MM  125066  INT$MM  125360
                                                                  125760  TMC$MM  125270  UMR$MM  126140  ..MMGP  126562
                                                                  131000  DEA$DX  131032  DXCVR   130050  ERL$DX  131110
                                                                  131256  SPC$DX  130060  TMC$DX  131030  UMR$DX  131002
                                                                  134552  ..CHFT  136566
                                                                  137376  STCIRP  137020
```

Figure MON-76. RSTS.MAP (Cont.)

Core Dump of Monitor

```
000000/ 000186  004340  100310  004340  100372  004340  100246  004340
000020/ 100070  004340  103214  004340  101216  004340  101066  004140
000040/ 000137  073056  021533  000156  000773  000434  000000  000770
000060/ 076636  004200  076646  004200  100116  004356  100116  004357
000100/ 115234  004300  115242  004317  000137  100112  103620  004340
000120/ 100122  004344  103122  004345  100122  004346  100122  004347
000140/ 100122  004350  000167  072742  075674  004240  100122  004353
000160/ 100122  004354  100122  004355  100122  004356  100122  004357
000200/ 077166  004200  075614  004240  075654  004240  077212  004300
000220/ 100126  004344  077176  004240  100126  004346  075512  004347
000240/ 100126  004350  045742  004340  100434  004340  075714  004240
000260/ 100126  004354  100126  004355  100126  004356  100126  004357
000300/ 076656  004240  076666  004240  076656  004241  076666  004241
000320/ 076656  004242  076666  004242  077010  004240  077020  004240
000340/ 077010  004241  077020  004241  077010  004242  077020  004242
000360/ 077010  004243  077020  004243  077010  004244  077020  004244
000400/ 077010  004245  077020  004245  100136  004342  100136  004343
000420/ 076676  004240  076706  004240  076676  004241  076706  004241
000440/ 076676  004242  076706  004242  100136  004352  100136  004353
000460/ 100136  004354  100136  004355  100136  004356  100136  004357
000500/ 100142  004340  100142  004341  100142  004342  100142  004343
000520/ 100142  004344  100142  004345  100142  004346  100142  004347
000540/ 100142  004350  100142  004351  100142  004352  100142  004353
000560/ 100142  004354  100142  004355  100142  004356  100142  004357
000600/ 100146  004340  100146  004341  100146  004342  100146  004343
000620/ 100146  004344  100146  004345  100146  004346  100146  004347
000640/ 100146  004350  100146  004351  100146  004352  100146  004353
000660/ 100146  004354  100146  004355  100146  004356  100146  004357
000700/ 100152  004340  100152  004341  100152  004342  100152  004343
000720/ 100152  004344  100152  004345  100152  004346  100152  004347
000740/ 100152  004350  100152  004351  100152  004352  100152  004353
000760/ 100152  004354  100152  004355  100152  004356  100152  004357
001000/ 021533  000114  011017  000000  000000  110276  066000  000000
001020/ 066000  000000  013700  000000  000000  000000  000000  000000
001040/ 000000  001110  000057  100057  000000  000000  000000  000000
001060/ 000000  000000  000000  000000  000000  000000  000000  000000
001120/ 205440  036273  152560  006253  001042  044442  000020  100020
001122/ 002740  000012  010362  000230  000023  106014  001020  060400
001140/ 011752  000000  000001  100001  040000  000000  000000  000000
001160/ 000000  000000  000000  000000  000000  000000  000000  000000
001200/ 025600  051105  046122  043517  000002  000003  002200  001050
001220/ 012300  013440  000000  000000  000000  000000  000000  000000
001240/ 000000  111574  071620  000000  000000  000000  000000  100000
001260/ 000000  000000  000000  000000  000000  000401  000437  000000
001300/ 014500  000026  000747  015100  013700  116767  104130  177754
001320/ 005726  016767  177452  176514  016767  177446  176510  005000
```

Figure MON-77. RSTS Dump


```
[illegible]
001500/ 000000  000000  000000  000000  000000  162746  000010  005356
001620/ 176700  001040  104246  062156  000015  110604  052172  104560
001640/ 000204  076746  104560  025300  000002  110504  052162  176312
001660/ 061060  000015  110604  052162  176312  000014  176300  002640
001700/ 000001  000100  010004  010004  100030  001736  003000  000000
001720/ 110604  052162  176312  000014  176300  057122  000031  123050
001740/ 104554  001670  000002  052162  000372  000003  176300  052210
001760/ 167077  122704  104554  001670  037400  000372  000003  002374
002000/ 120120  000003  000374  054040  000000  000000  000000  000000
002020/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  4. Duplicate Lines (002040 - 002120)  ******
002140/ 000000  000000  000000  000000  000000  000000  000000  162746
002160/ 000006  110604  052172  104560  104560  000220  062324  000017
002200/ 160266  021440  102770  065741  102222  010413  011504  021420
002220/ 067304  021554  001200  013440  021510  021500  000000  103026
002240/ 001420  000010  000000  021500  006740  000040  010567  172354
002260/ 000000  006740  100716  000046  103024  034144  006540  136646
002300/ 067122  134332  110604  000000  000000  027265  000020  067122
002320/ 000014  105752  002070  106216  106132  110604  000000  000000
002340/ 027265  000001  067122  125130  122466  014600  014606  122214
002360/ 067202  003120  123722  031640  041120  124124  131120  122100
002400/ 000001  000356  000000  007046  000000  000000  060123  071726
002420/ 001100  005600  000000  000000  000412  007033  000000  000000
002440/ 001000  001000  015246  000010  000001  010632  000000  000002
002460/ 135400  070014  005325  002401  000000  000000  000000  000000
002500/ 052720  000000  006000  006140  006300  000000  000000  000000
002520/ 000000  000000  000000  000000  000000  000000  000000  000000
002540/ 000004  000220  000000  000464  000000  000000  016366  132022
002560/ 001100  000000  000000  000000  000412  007033  000000  000000
002600/ 000000  026032  000416  000770  000003  000000  000000  000000
002620/ 000000  000000  000000  000000  000000  000000  000000  145324
002640/ 000000  173434  075734  000406  026701  006571  025146  177400
002660/ 000400  017740  000000  002004  000742  002000  127600  000000
002700/ 000000  000000  000000  000000  000000  064426  000000  000000
002720/ 110710  041035  000000  000000  051610  140000  010247  001035
002740/ 002500  003002  162400  002440  002400  010000  001100  000000
002760/ 013662  006460  000020  000020  014440  000000  002270  000037
003000/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  21. Duplicate Lines (003020 - 003520)  ******
003540/ 000001  021510  000105  021510  000510  006273  152560  000010
003560/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  2. Duplicate Lines (003600 - 003620)  ******
003640/ 041120  023146  141057  111233  024000  000000  031540  031660
003660/ 051240  007061  007123  007301  007611  010175  010357  010373
```

Figure MON-77. RSTS Dump (Cont.)

```
003700/ 000000 000000 000000 000000 000000 000000 000000 000000
        ******  2. Duplicate Lines (003720 - 003740)    ******
003760/ 000000 045651 045655 062445 063447 000000 000000 000000
004000/ 177777 177777 177777 177777 177777 177777 177777 177777
004020/ 177777 177777 177777 177777 177777 177777 177777 177777
004040/ 177777 177577 177777 177777 177777 177777 177777 177777
004060/ 177777 177777 177777 177777 177777 177777 177777 177777
        ******  17. Duplicate Lines (004100 - 004500)   ******
004520/ 177777 177777 177777 102163 075614 031440 077506 000000
004540/ 077506 031640 077406 000200 077506 032040 077406 000400
004560/ 077506 032240 037506 000600 077506 033440 077402 001000
004600/ 077402 033640 077402 001200 077402 034040 077402 010567
004620/ 000015 061146 000000 001000 000000 030020 177400 051104
004660/ 036010 100000 100000 001000 000004 077430 005472 015464
004720/ 024050 000014 001034 000000 025200 000000 000002 000424
004740/ 000001 000006 151634 145440 061146 001740 027265 000000
004760/ 000000 110504 134332 067122 136646 006540 034144 103024
005000/ 000046 000000 010567 172354 000000 140000 002270 000046
005020/ 005400 006273 034270 006253 013162 014660 006034 100020
005040/ 024240 000000 000332 000210 000011 000000 001020 020400
005060/ 000000 071233 070533 003001 000003 000000 000000 001027
005100/ 013200 000000 001112 000240 000012 000000 025200 040000
005120/ 015400 005116 005121 000402 046537 025160 177777 052101
005140/ 040524 044103 000377 000000 000000 000000 005121 172460
005160/ 001120 000000 000000 000300 000412 007033 000300 000000
005200/ 000002 000623 000000 014317 000000 000000 106303 137031
005220/ 001100 000000 000000 000000 000412 007033 000000 000000
005240/ 001750 000000 015264 000002 000000 000000 020000 000002
005260/ 135400 070014 005125 043001 013000 000000 000000 000000
005300/ 062136 062136 000000 000000 000000 000000 000000 000000
005320/ 000000 000000 000000 000000 000000 000000 000000 000000
005340/ 053006 000000 000000 000000 000000 000000 000000 000000
005360/ 000000 000000 000000 000000 000000 000000 000000 000000
005400/ 020740 000000 000000 000000 000000 064566 000000 000000
005420/ 174110 001035 000000 000000 020207 140000 007101 001035
005440/ 005000 031573 077537 000000 006412 012660 000034 100034
005460/ 011400 000012 000032 000078 000021 100000 001004 160777
005500/ 020001 000324 000001 006136 000000 000000 066625 071726
005520/ 001100 007300 000000 000000 000412 007033 000000 000000
005540/ 001000 001000 015746 000004 000002 010646 000000 000002
005560/ 052044 046505 030120 027064 000000 000105 000000 000000
005600/ 007300 050117 042523 020122 000006 000001 000000 000036
005620/ 000000 000620 000000 000000 005626 000000 000000 000000
005640/ 000540 110160 000402 066615 102010 075273 024004 000001
005660/ 000130 024012 000070 024012 000012 000320 000010 006740
005720/ 000200 100160 000412 060123 021020 046537 036004 000001
005740/ 052720 102423 000070 070335 000012 000012 000010 006000
005760/ 000000 006740 006700 007000 064052 007040 064052 000000
```

Figure MON-77. RSTS Dump (Cont.)


```
006200/ 000000 000000 000000 000012 005734 000000 000000 177777
006220/ 000000 000000 000000 000000 000000 000000 000000 000000
006240/ 020240 006256 006261 000402 010603 070516 100003 030502
006260/ 051103 043105 000377 000000 000000 000000 006261 072460
006300/ 100000 000406 000000 000001 006234 000000 000020 177777
006320/ 177777 004317 004320 004321 000000 000000 000000 000000
006340/ 012200 006356 006361 000402 046166 042300 073376 044514
006360/ 045516 000377 000000 000000 051504 177400 006361 020020
006400/ 006500 076713 056700 076713 044444 005450 000010 100010
006420/ 011000 000000 000042 000250 000011 100000 000200 000400
006440/ 005740 000002 162400 005540 005500 010000 001100 000000
006460/ 002762 010020 000020 000020 015440 000000 002270 000037
006500/ 011500 071677 140700 073376 000000 000000 000000 001004
006520/ 000000 000000 000442 000300 000011 000000 001034 120777
006540/ 005700 022120 000402 076725 064322 066615 026204 000001
006560/ 000060 050423 000050 050423 000012 000054 000310 000700
006600/ 005640 110120 000402 066615 102010 111233 026004 000001
006620/ 000110 024012 000050 024012 000012 000021 000010 000700
006640/ 006600 020300 000412 006274 013624 111233 036004 000001
006660/ 000160 070323 000140 070323 000012 000011 000210 000400
006700/ 100000 040410 000000 000016 006574 000000 000300 177777
006720/ 177777 014103 015707 017670 020475 025651 025700 000000
006740/ 100000 040410 000000 000002 005674 000000 000000 000130
006760/ 024013 004324 004326 010016 031730 032370 042406 042671
007000/ 100000 040410 000000 000002 006634 000000 000000 177777
007020/ 177777 004325 004331 020126 000000 000000 000000 000000
007040/ 002000 000010 000000 000000 006234 000000 006300 177777
007060/ 177777 004317 004320 004321 000000 000000 000000 000000
007100/ 000006 000000 000000 000000 000000 000000 074514 034354
007120/ 001100 007340 000000 000000 000412 007033 000000 000000
007140/ 000000 023012 004377 052521 046505 047101 002002 000000
007160/ 000000 000000 000000 000000 000000 000000 000000 000000
007200/ 052720 007540 000000 000000 015200 000000 000000 000000
007220/ 000000 000000 000000 000000 000000 000000 000000 000000
007240/ 007200 003002 162400 007140 007100 010000 001100 000000
007260/ 011512 010460 000020 000020 017500 000000 002370 002237
007300/ 007340 052521 046505 047101 000010 000001 000000 000074
007320/ 000000 007320 000000 000003 007325 000000 000000 000200
007340/ 007640 050114 051460 046120 000012 000003 000000 000005
007360/ 000000 007360 000000 000000 007366 000000 000000 000000
007400/ 100000 040414 000000 000001 006674 000000 000000 177777
007420/ 177777 000744 017123 000000 000000 000000 000000 000000
007440/ 000003 000047 000000 000723 000000 000000 006274 034354
```

Figure MON-77. RSTS Dump (Cont.)

```
007460/ 001100  007640  000000  000000  000412  007033  000000  000300
007500/ 011240  140640  000412  074514  140123  111233  036004  000001
007520/ 000330  070335  000320  070335  000012  000007  000010  007540
007540/ 140000  040412  000000  000001  007534  000000  000000  177777
007560/ 177777  000743  000000  000000  000000  000000  000000  000000
007600/ 000000  024014  004377  000412  051460  000000  000002  000000
007620/ 000000  000300  000000  000300  000000  000000  000000  000P00
007640/ 010140  040502  051460  046120  000014  000003  000000  000005
007660/ 000P00  007660  000000  000000  007666  000000  000000  000000
007700/ 000004  000047  000000  000724  000000  000000  000000  000000
007720/ 001100  010140  000000  000000  000412  007033  000000  000000
007740/ 052720  007400  000000  000000  000000  000000  000000  000000
007760/ 000000  000000  000000  000000  000000  000000  000000  000000
010000/ 007740  003002  162400  007600  007440  006400  001100  000000
010020/ 006462  011510  000015  000015  016440  000000  001770  000037
010040/ 020140  000000  000000  000000  000000  064346  000000  000000
010060/ 171710  001035  000000  000000  013607  140000  007057  001035
010100/ 000000  024016  004377  000412  051461  000000  000002  000000
010120/ 000000  000000  000000  000000  000000  000000  000000  000000
010140/ 010540  040502  051461  046120  000016  000003  000000  000005
010160/ 000000  010160  000000  000000  010166  000000  000000  000000
010200/ 006640  170240  000412  006274  013625  111233  036004  000001
010220/ 000300  102421  000120  070340  000012  000011  000010  010500
010240/ 052720  010503  000000  000000  000000  000000  000000  000000
010260/ 000000  000000  000000  000600  000000  000000  000000  000000
010300/ 013440  013300  000377  000000  000000  000064  000007  000004
010320/ 000036  021533  000114  011017  000000  000000  000000  000000
010340/ 007500  000020  000412  012001  071570  066615  036000  000001
010360/ 002160  070335  000010  070321  000012  000002  000010  015200
010400/ 020400  053516  045520  030061  174424  000004  000000  000005
010420/ 000000  010420  000001  000000  010426  000000  000000  000000
010440/ 010240  003002  162400  010100  007700  006400  001100  000000
010460/ 007262  011060  000015  000015  020500  000000  001770  000037
010500/ 100000  040416  000000  000001  010234  000002  000000  177777
010520/ 177777  001515  020245  000000  000000  000000  000000  000000
010540/ 013000  040502  051462  046120  000020  000003  000000  000005
010560/ 000000  010560  000000  000000  010566  000000  000000  000000
010600/ 010200  031220  000412  006274  013626  111233  036004  000001
010620/ 000120  102424  000110  102424  000012  000011  000010  010700
010640/ 000007  000046  000000  000710  000000  000000  006274  034354
010660/ 001100  010540  000300  000000  000412  007033  000000  000000
010700/ 100000  040420  000000  000001  010634  000000  000000  177777
010720/ 177777  015136  015237  000000  000000  000000  017402  000000
010740/ 000000  024020  004377  000412  051462  000000  000000  000000
010760/ 000000  000000  000000  000000  000000  000000  000000  000000
011000/ 052720  012700  000000  000000  000000  000000  000000  000000
011020/ 000000  000000  000000  000000  000000  000000  000000  000000
011040/ 011000  003002  162400  010740  010640  006400  001100  000000
011060/ 013462  011360  000015  000015  021340  000000  001770  000037
011100/ 000004  000043  000000  000625  000000  000000  054015  000000
011120/ 001100  013000  000000  001000  000412  007033  000000  000000
```

Figure MON-77. RSTS Dump (Cont.)


```
011140/ 177777  032400  032547  000000  000000  000000  000000  000000
011160/ 177520  000000  000000  000000  000000  000000  000000  000000
011200/ 000000  000000  000000  000542  000030  000012  000300  000434
011220/ 000000  000000  000572  000060  000012  000000  000000  001004
011240/ 000000  000000  000712  000120  000012  000000  000434  000400
011260/ 000000  000000  000742  000150  000012  000401  200420  000000
011300/ 000000  000000  001362  000210  000012  000000  000434  000400
011320/ 000000  000000  001462  000210  000012  000000  000434  000400
011340/ 000000  000000  001562  000210  000012  000000  000434  000400
011360/ 000000  000000  001662  000210  000012  000000  000434  000400
011400/ 000000  000000  001762  000210  000012  000000  000434  000400
011420/ 000000  000000  002062  000210  000012  000000  000434  000400
011440/ 000000  000000  002162  000210  000012  000000  000434  000400
011460/ 000000  000000  002262  000210  000012  000000  000434  000400
011500/ 000000  000000  002362  000210  000012  000000  000434  000400
011520/ 000000  000000  002462  000210  000012  000000  000434  000400
011540/ 000000  000000  002562  000210  000012  000000  000434  000400
011560/ 000000  000000  002662  000210  000012  000000  000434  000400
011600/ 000000  000000  002762  000210  000012  000000  000434  000400
011620/ 000000  000000  003062  000210  000012  000000  000434  000400
011640/ 000000  000000  003162  000210  000012  000000  000434  000400
011660/ 000000  000000  003262  000210  000012  000000  000434  000400
011700/ 000000  000000  003362  000210  000012  000000  000434  000400
011720/ 000000  000000  003462  000210  000012  000000  000434  000400
011740/ 000000  000000  003562  000210  000012  000000  000434  000400
011760/ 000000  000000  003662  000210  000012  000000  000434  000400
012000/ 000000  000000  003762  000210  000012  000000  000434  000400
012020/ 000000  000000  004062  000210  000012  000000  000434  000400
012040/ 000000  000000  004162  000210  000012  000000  000434  000400
012060/ 000000  000000  004262  000210  000012  000000  000434  000400
012100/ 000000  000000  004362  000210  000012  000000  000434  000400
012120/ 000000  000000  004462  000210  000012  000000  000434  000400
012140/ 000000  000000  004562  000210  000012  000000  000434  000400
012160/ 000000  000000  004662  000210  000012  000000  000434  000400
012200/ 000000  000000  004762  000210  000012  000000  000434  000400
012220/ 000000  000000  005062  000210  000012  000000  000434  000400
012240/ 000000  000000  005162  000210  000012  000000  000434  000400
012260/ 000000  000000  005262  000210  000012  000000  000434  000400
012300/ 000000  000000  005362  000210  000012  000000  000434  000400
012320/ 000000  000000  005462  000210  000012  000000  000434  000400
012340/ 000000  000000  005562  000210  000012  000000  000434  000400
012360/ 000000  000000  005662  000210  000012  000000  000434  000400
012400/ 000000  000000  005762  000210  000012  000000  000434  000400
012420/ 000000  000000  006062  000210  000012  000000  000434  000400
012440/ 000000  000000  006162  000210  000012  000000  000434  000400
012460/ 000000  000000  006262  000210  000012  000000  000434  000400
012500/ 000000  000000  006362  000210  000012  000000  000434  000400
012520/ 000000  000000  006462  000210  000012  000000  000434  000400
012540/ 000000  000000  006562  000210  000012  000000  000434  000400
012560/ 000000  000000  006662  000210  000012  000000  000434  000400
012600/ 000000  000000  006762  000210  000012  000000  000434  000400
```

Figure MON-77. RSTS Dump (Cont.)

```
012620/ 000000  000000  002000  000000  000000  000000  000000  000000
012640/ 012600  000003  006005  012540  012500  002400  001100  000000
012660/ 025452  013660  000013  000005  013200  000000  003370  000037
012700/ 011202  012716  012721  000402  054204  143660  100003  043116
012720/ 177524  000200  000000  003000  000000  000000  012721  100000
012740/ 000006  000061  000001  001141  000001  000000  055013  016202
012760/ 001120  010400  000003  001000  000412  007033  000000  000000
013000/ 010400  041516  020125  020040  011422  000007  001000  000005
013020/ 000200  013020  000010  000000  013026  000000  000000  000000
013040/ 000000  026024  002002  000000  000000  000000  000000  000000
013060/ 000000  000764  000000  000000  000000  000000  000000  000000
013100/ 052720  000000  064052  000000  000000  000000  000000  000000
013120/ 000000  000000  000000  000000  000000  000000  000000  000000
013140/ 013100  000003  006005  013040  012740  004000  001100  000000
013160/ 011362  005010  000017  000010  023100  000000  001370  000037
013200/ 020000  177414  075415  000406  026701  006571  025146  177400
013220/ 000404  013740  000000  002004  000742  002000  127600  000000
013240/ 000000  066144  026145  020040  020040  027040  022463  067514
013260/ 072163  020015  020040  027063  020070  030040  115446  021531
013300/ 000100  000010  000000  000064  000400  001022  006740  000040
013320/ 010560  172354  010567  140000  103020  034140  001000  002274
013340/ 006540  136646  065752  010004  010413  010402  177777  000000
013360/ 001400  002004  000000  056702  000004  004004  127600  000000
013400/ 025027  025052  040440  045522  020040  020055  042504  067103
013420/ 072145  042457  025040  025052  000000  000000  000000  000000
013440/ 000000  021500  000377  000000  000000  000064  000007  000004
013460/ 000037  021533  000114  011017  000000  000000  000000  000000
013500/ 017700  002312  047516  044101  000000  000215  000006  064506
013520/ 000000  000000  000000  000000  001035  000040  000000  013534
013540/ 020340  000340  000000  000013  013374  000000  000000  177777
013560/ 177777  005353  045577  000000  000000  000000  000000  000000
013600/ 000300  005062  006412  006412  000012  064346  000000  000000
013620/ 000000  001035  000030  000000  010207  140000  007041  140014
013640/ 005340  002002  162400  002600  002540  001000  001240  000000
013660/ 012662  002760  000012  000002  013740  000000  005170  000037
013700/ 000001  005442  000002  125212  000000  000000  006253  012446
013720/ 001100  020400  000000  000000  000747  031133  000000  000000
013740/ 000000  030020  024050  015464  005472  077430  036024  000001
013760/ 000020  021134  000010  021134  000006  000004  000010  017740
014000/ 000000  000000  000006  000000  001200  000000  012540  000000
014020/ 000000  000000  000000  000000  000000  000000  000000  000000
014040/ 020700  001700  051101  000113  000000  000214  000000  000000
014060/ 000000  000000  000000  000000  001035  000040  000000  014074
014100/ 000000  051516  020120  020040  000003  000000  000000  000000
014120/ 000300  014120  000040  000000  014126  000000  000000  000000
014140/ 000240  000010  000000  000000  000000  000000  001023  000004
014160/ 000020  021533  000153  025037  000000  000000  000000  000000
014200/ 017700  000000  000000  000000  000411  000001  000412  103151
014220/ 047671  011500  000014  006273  152560  002740  001030  000000
014240/ 177500  121510  004010  000000  000000  041104  177400  000000
```

Figure MON-77. RSTS Dump (Cont.)


```
014400/ 000200  000010  012467  020170  047516  044101  000036  000002
014420/ 003000  053516  052124  031060  030000  000030  000021  000221
014440/ 000375  174400  000000  000000  000000  000000  000000  000000
014460/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  4. Duplicate Lines (014500 - 014560)    ******
014600/ 000000  001026  000002  000747  023150  135760  111233  000000
014620/ 000000  100400  000000  024377  051104  177400  000000  000000
014640/ 005300  023002  162400  005240  005200  010000  001100  000000
014660/ 005012  015060  000020  000020  025640  000000  002370  000037
014700/ 000200  000000  031414  000002  000000  000000  020000  000002
014720/ 135400  070014  005325  041001  013000  000000  000000  000000
014740/ 062050  000000  000000  000000  000000  000000  000000  000000
014760/ 000000  000000  000000  000000  000000  000000  000000  000000
015000/ 061606  061606  000000  000000  000000  000000  000000  000000
015020/ 000000  000000  000000  000000  000000  000000  000000  000000
015040/ 015000  003002  162400  014700  005140  006400  001100  000000
015060/ 014662  022420  000054  000015  026640  000000  001770  000037
015100/ 014740  023003  002000  014600  013700  004000  001100  000000
015120/ 022422  011750  000020  000010  032440  000000  002200  000037
015140/ 000000  027062  020071  020040  027062  115460  020110  020040
015160/ 030056  045633  005015  045504  035061  020040  020040  020040
015200/ 002000  000012  000000  000000  010374  000000  000000  177777
015220/ 177777  007412  000000  000300  000000  000000  000000  000000
015240/ 000000  002712  044120  047522  000107  000217  000036  064346
015260/ 000000  000000  000000  000000  001035  000040  000000  015274
015300/ 000000  026512  006523  000012  002674  000000  000000  177777
015320/ 177777  001167  001170  001171  001172  001173  001174  000000
015340/ 000000  015340  021100  020300  000300  015350  000000  015354
015360/ 000000  015360  000000  015364  000003  000000  002004  000000
015400/ 006240  015416  015421  000402  004560  042330  177777  052101
015420/ 045520  000377  000000  000000  000000  000000  015421  172460
015440/ 000000  006520  047455  045015  051455  005015  105000  152400
015460/ 000404  015300  000000  000412  000314  002004  127600  001115
015500/ 015640  015516  015521  000404  014501  000000  006253  042504
015520/ 052131  046511  177505  000000  000000  000000  015521  100000
015540/ 015700  015556  015561  000402  015172  017714  177777  044504
015560/ 042522  052103  051117  177531  000000  000000  015561  172460
015600/ 016000  015616  015621  000402  023752  077721  073376  047506
015620/ 052122  040522  177516  000000  000000  000000  015621  020020
015640/ 015540  015656  015661  000402  102527  102524  177777  044504
015660/ 046523  052517  052116  000377  000000  020000  015661  172460
015700/ 015740  015716  015720  000304  015532  004540  006253  047504
015720/ 040502  177524  000000  000000  000000  000000  015720  172460
015740/ 016200  015756  015761  000402  017764  000000  100003  042125
015760/ 177524  000000  000000  000000  000000  000000  015761  000000
016000/ 016440  016016  016021  000402  025340  000000  100003  022126
```

Figure MON-77. RSTS Dump (Cont.)

```
016220/ 177520  000030  000000  000000  020000  000000  016021  000022
016240/ 016320  016056  016061  000402  102527  102604  177777  047515
016260/ 247125  177524  000000  000030  000000  002000  016061  172460
016100/ 216240  016116  016121  000404  047014  046537  006253  051514
016120/ 046124  043517  000377  000000  000000  000000  016121  100000
016140/ 020500  016156  016161  000402  007403  070516  177777  050102
016160/ 051103  043105  000377  000000  000000  000000  016161  072460
016200/ 015600  016216  016220  000404  020604  021020  006253  047105
016220/ 042524  177522  000000  000000  000000  000000  016220  072460
016240/ 016040  016256  016261  000402  076713  056700  076713  040515
016260/ 042513  000377  000000  000000  000000  000000  016261  000000
016302/ 016600  016316  016320  000402  076713  056700  076713  052515
016320/ 042516  000377  000000  000000  000000  000000  016320  000000
016340/ 016540  016356  016361  000402  046152  070200  073376  044514
016360/ 051102  000377  000000  000000  000000  000000  016361  020020
016400/ 016700  016416  016420  000402  062745  004475  177777  046120
016420/ 040505  042523  000377  000000  000000  000000  016420  172460
016440/ 016500  016456  016461  000402  046537  035160  177777  042510
016460/ 046114  177517  000000  000000  000000  000000  016461  100000
016500/ 016340  016516  016520  000402  042614  157700  073376  042513
016520/ 032504  177462  000000  000000  000000  000000  016520  020020
016540/ 017000  016556  016560  000402  100071  073634  177777  042523
016560/ 177524  000000  000000  000000  000000  000000  016560  172460
016600/ 016400  016616  016621  000402  062074  000000  100003  040520
016620/ 177524  000000  000000  000000  000000  000000  016621  000000
016640/ 016100  016656  016661  000402  046537  035160  177777  047514
016660/ 044507  177516  000000  000000  000000  000000  016661  100000
016700/ 016540  016716  016720  000402  066615  000000  177777  052521
016720/ 052505  177505  000000  000000  000000  000000  016720  172460
016740/ 017140  016756  016761  000402  066615  000000  177777  052523
016760/ 046502  052111  000377  000000  000000  000000  016761  172460
017000/ 017100  017016  017021  000402  023374  000000  177777  044506
017020/ 177524  000000  000000  000000  000000  000000  017021  172460
017040/ 017200  017056  017060  000402  075273  076474  177777  054523
017060/ 052123  052101  000377  000000  000000  000000  017060  172460
017100/ 016740  017116  017121  000402  074064  103070  073376  044523
017120/ 052514  177523  000000  000000  000000  000000  017121  020000
017140/ 017040  017156  017160  000402  075141  076600  177777  053523
017160/ 052111  044103  000377  000000  000000  000000  017160  172460
017200/ 017300  017216  017220  000404  076521  076400  006253  041124
017220/ 052101  000377  000000  000000  000000  000000  017220  172460
017240/ 017500  017256  017261  024050  054137  015464  100003  047104
017260/ 177504  000000  000000  000000  051104  177400  017261  000000
017300/ 017340  017316  017320  000402  076713  056700  076713  042524
017320/ 047503  000377  000000  000000  000000  000000  017320  000000
017340/ 017440  017356  017360  000402  100370  017500  076713  054524
017360/ 042520  000377  000000  000000  000000  000000  017360  000010
017400/ 000000  053057  032524  006462  033012  006460  047012  006517
017420/ 054412  051505  005015  046502  041454  055123  045633  054633
017440/ 017240  017456  017460  000402  103151  047071  177777  052125
017460/ 046111  054524  000377  000000  000000  000000  017460  072460
```

Figure MON-77. RSTS Dump (Cont.)


```
020100/ 020200  020116  020121  000402  006404  006274  000000  006253  040502
020120/ 177524  000000  000000  000000  000000  000000  020121  172460
020140/ 020300  000000  000000  000000  000000  064346  000000  000000
020160/ 175210  001035  000000  000000  022407  140000  007112  001035
020200/ 005100  020216  020221  000402  046537  050434  177777  054502
020220/ 177505  000000  000000  000000  000000  000000  020221  100000
020240/ 016140  020256  020261  000402  010603  070516  100003  031102
020260/ 051103  043105  000377  000000  000000  000000  020261  072460
020300/ 000000  000000  000000  000000  000000  064346  000000  000000
020320/ 000510  041035  000000  000000  031207  140000  007145  001035
020340/ 020540  000040  075734  000406  026701  006561  025146  000400
020360/ 000402  017740  000000  002004  000742  001000  127600  000045
020400/ 000000  040502  045503  030461  000026  000303  000000  000005
020420/ 000000  020420  000000  000000  020426  000000  000000  000000
020440/ 000000  020440  021040  021200  000000  020450  000000  020454
020460/ 000000  020460  000000  020464  000000  000000  002004  000000
020500/ 015500  020516  020521  000402  012626  000003  100003  051103
020520/ 177506  000000  000000  000000  000000  000000  020521  000000
020540/ 020640  000040  075734  000012  000304  006745  062156  000000
020560/ 000004  010300  000000  000000  042407  140000  007212  140014
020600/ 017600  000000  000000  000000  000000  064506  000000  000000
020620/ 001610  041035  000000  000000  033407  140000  007156  001035
020640/ 021140  000040  031472  115463  006513  115412  020113  020040
020660/ 030440  027065  022461  071525  071145  020054  020040  033440
020700/ 013500  002312  040526  032130  000000  000220  000207  064566
020720/ 000000  000000  000000  000000  001100  000040  000000  020734
020740/ 000000  000000  000000  000000  000000  064566  000000  000000
020760/ 173010  001035  000000  000000  015007  140000  007276  001035
021000/ 000000  021000  022240  020740  000006  021010  000000  021014
021020/ 000000  021020  000000  021024  000000  000000  002004  000000
021040/ 020600  000000  000000  000000  000000  064506  000000  000000
021060/ 177410  001035  000000  000000  027007  140000  007134  001035
021100/ 010040  000000  000000  000000  000000  064346  000000  000000
021120/ 004110  041035  000000  000000  040207  140000  007201  001035
021140/ 021240  000040  075734  001012  151051  006571  062156  174000
021160/ 000404  010300  000000  002416  000672  002004  127600  001111
```

Figure MON-77. RSTS Dump (Cont.)

```
021200/ 000000 000000 000000 000000 000000 064506 000000 000000
021220/ 176312 001035 000000 000000 024607 140000 007123 001035
021240/ 021600 000040 075734 001012 101131 006571 062156 174000
021260/ 000404 010320 000000 000000 003424 000611 002004 127600 001000
021300/ 001340 000000 000000 000000 000000 064426 000000 000000
021320/ 131610 041035 000000 000000 113410 140000 010456 001035
021340/ 021400 000000 000000 000000 111210 140000 010445 001035
021360/ 130510 041035 000000 000000 111210 140000 010445 001035
021400/ 000700 000000 000000 000000 000000 064426 000000 000000
021420/ 127410 041035 000000 000000 005015 005015 026511 026455 044511
021440/ 000000 005012 005012 005015 005015 026511 026455 044511
021460/ 005015 007055 000000 010020 001404 002004 127600 001061
021500/ 000120 000310 000000 000064 000400 001022 006740 000040
021520/ 000000 177764 000000 140000 100024 014144 001000 002274
021540/ 006540 136646 000413 010402 011504 001420 177777 000000
021560/ 001400 002004 000002 056700 000004 104004 020072 020040
021600/ 021740 000100 020040 027040 115460 006513 042012 030502
021620/ 020072 020040 020040 020040 027040 115460 006513 042012
021640/ 022000 000000 020040 027071 020066 020040 027040 020064
021660/ 020040 030440 034465 033456 020040 034440 022460 030440
021700/ 020040 000000 000000 000000 000000 000000 000000 000000
021720/ 000000 000000 000020 000000 000000 000000 000000 015240
021740/ 022200 000140 030456 020040 034064 030056 020040 030440
021760/ 032456 045633 005015 005015 057524 060564 071554 020040
022000/ 022200 000100 020040 020040 020040 030056 045633 005015
022020/ 051104 035060 020040 020040 020040 020040 030056 045633
022040/ 022200 000040 020040 020040 020040 020040 031040 031062 031456
022060/ 005015 041104 035062 020040 020040 020040 054137 015046
022100/ 000004 021454 000025 024600 000000 000000 062144 112517 000000 000000
022120/ 001100 000000 000000 000000 062144 112517 000000 000000 000002
022140/ 001000 001000 061146 000014 000001 000000 000000 000000 000000
022160/ 042116 040400 027524 000101 000000 000000 000000 000000 000000
022200/ 022300 000040 020040 020040 020040 027040 115460 006513 042012
022220/ 030515 020072 020040 020040 020040 027040 115460 006513 042012
022244/ 023700 000000 000000 000000 000000 000000 000000 000000 000000
022260/ 016710 041035 000020 000000 065607 140000 007327 001035
022300/ 022440 000040 034440 022464 020040 034440 022471 031040
022320/ 032461 031056 020040 033040 034456 020040 020040 020040 031056
022340/ 062312 000000 000000 000000 000000 000000 000000 017740 000000
022360/ 000000 000000 000000 000000 000000 000000 000000 000000 000000
022400/ 022340 025403 162400 022140 022100 007000 011740 000000 000000
022420/ 015062 015120 000020 100016 031440 000000 001770 000037
022440/ 023740 001240 000000 000000 000411 177672 000747 006253
022460/ 043757 032440 100020 006273 152560 002740 001027 000000 000001
022500/ 046515 177401 000023 000314 000426 000125 100001 000021 000361
022520/ 000000 000000 000000 077206 016156 010004 000021 172440
022540/ 001002 010660 004060 000000 014754 014600 001401 150660 100000
022560/ 002017 144660 000000 072162 000000 000100 000015 000000
022600/ 000001 000040 001000 142011 062500 102301 000000 000000
022620/ 007376 001100 141100 140660 140660 000000 000000 000000
```

Figure MON-77. RSTS Dump (Cont.)


```
023240/ 023740 000040 000402 121210 000000 072410 000000 000000
023260/ 010352 001100 141100 140011 140011 000000 000000 000000
023300/ 023740 000400 000402 113510 000000 057210 000000 000020
023320/ 010275 001100 141100 140555 140555 000000 000000 000000
023340/ 023740 000340 041515 027040 051514 006524 024012 047117
023360/ 052040 040522 051516 042506 024522 006412 127600 001000
023400/ 023740 000300 036040 042522 037124 072040 020157 072560
023420/ 020164 072151 067440 026156 067440 020162 046074 037106
023440/ 023740 000000 020165 060550 062566 063040 072557 062156
023460/ 060440 020156 066105 062566 020156 066103 060557 020153
023500/ 023740 000200 075734 002012 001301 006571 072156 177776
023520/ 000404 020740 000000 010000 001164 002004 127600 001106
023540/ 023740 000140 075734 000412 074571 006571 062156 174000
023560/ 000404 020740 000000 001420 000347 002004 127600 001106
023600/ 023740 000100 020040 030056 045633 005015 041104 035060
023620/ 020040 020040 020040 020040 030056 045633 005015 041104
023640/ 024040 000000 020040 020040 020040 020040 030056 045633
023660/ 005015 041104 035062 020040 020040 020040 034466 031456
023700/ 005400 000000 000000 000000 000000 064566 000000 000000
023720/ 170110 001035 000000 000000 010207 140000 007041 001035
023740/ 000000 020500 034440 022460 020040 034440 022466 020040
023760/ 032465 034056 020040 031061 034056 020040 020040 020040 033456
024000/ 000000 000000 032071 020045 032440 027067 020064 030440
024020/ 027070 020066 020040 027061 020061 020040 030430 034461
024040/ 000000 020400 075734 000012 143701 006571 062156 174000
024060/ 000404 022500 000000 003020 000172 002004 127600 001072
024100/ 000000 020340 041115 042105 042504 177104 000512 055533
024120/ 036133 036074 036074 036074 036074 022074 022044 022044
024140/ 024440 000000 022445 037045 037076 037076 037076 037076
024160/ 056476 056535 055412 055533 036074 036074 036074 036074
024200/ 024440 000140 022044 022044 022044 022044 022044 020441
024220/ 020441 020441 020441 020441 021443 021443 021443 021443
024240/ 024440 000100 037477 037477 037477 037477 037477 036475
024260/ 036475 036475 036475 036475 030060 030060 030060 030060
024300/ 024440 000040 025052 025052 025052 025052 025052 022046
```

Figure MON-77. RSTS Dump (Cont.)

```
024320/ 023046  023046  023046  023046  025453  025453  025453  025453
024340/ 024440  000000  022445  022445  022445  022445  022445  037076
024360/ 037076  037076  037076  037076  056535  006535  177414  005015
024400/ 000000  000000  037402  075510  000000  023204  000000  000000
024420/ 004115  001100  141100  140714  140714  000000  000000  000000
024440/ 000000  020000  075734  000406  026703  006571  025146  177400
024460/ 000404  015140  000000  002006  000742  002000  127600  000000
024500/ 000000  017740  000023  000014  000426  000125  100001  000001
024520/ 000000  000000  000000  077206  016156  010004  000021  000361
024540/ 001002  013660  001363  000000  014754  014600  001401  172440
024560/ 002017  144660  000000  072162  000000  000100  150640  100000
024500/ 000001  000002  001100  142011  062500  102301  000015  000000
024620/ 037076  037076  056535  006535  177414  045415  045513  045767
024640/ 000000  017600  075734  000012  012121  002571  130110  177400
024660/ 000404  022500  000020  004020  000014  002004  127600  001115
024700/ 000003  017540  046515  046515  176515  041503  173503  041503
024720/ 005103  045513  173513  045513  176513  046515  046515  046515
024740/ 000000  000000  046515  046515  176515  041503  173503  041503
024760/ 005103  045513  175113  045513  175113  046515  176515  046515
025000/ 000000  017440  075734  000412  074751  006571  062156  174000
025020/ 000404  020540  000000  004022  000347  002004  127600  001113
025040/ 000000  017400  075734  000006  107521  006171  063312  174000
025060/ 000404  025000  000000  000424  000345  002000  127600  000000
025100/ 000003  000000  000032  000010  000000  064346  000000  000000
025120/ 000000  041035  000000  000000  073407  140000  007356  140014
025140/ 000000  017300  075734  001412  120201  006571  062156  176400
025160/ 002404  024600  000000  006020  001070  002004  127600  001060
025200/ 000000  017240  022044  022044  022044  022044  020441  020441
025220/ 020441  020441  020441  021443  021443  021443  021443  021443
025240/ 025500  000200  030372  030060  176412  030461  030461  030461
025260/ 030461  140451  030461  030461  030461  030461  173461  030060
025300/ 026400  000000  030060  030060  030060  176460  176412  030461
025320/ 030461  030461  030461  140461  030461  030461  030461  030461 '
025340/ 026400  000000  030060  030060  030060  030060  176460  005012 '
025360/ 055412  055533  036074  036074  036074  036074  036074  022044
025400/ 026400  025042  022053  022046  023046  023046  023046  025446
025420/ 025453  025453  025453  025453  022453  022445  022445  022445
025440/ 026400  000000  037045  037076  037076  037076  037076  056476
025460/ 056535  055412  055533  036074  036074  036074  036074  036074
025500/ 026400  000100  037477  037477  037477  037477  036475  036475
025520/ 036475  036475  036475  030060  030060  030060  030060  030060
025540/ 026400  000040  025052  025052  025052  025052  023046  023046
025560/ 023046  023046  023046  025453  025453  025453  025453  025453
025600/ 026400  000000  022445  022445  022445  022445  037076  037076
025620/ 037076  037076  037076  056535  005135  005012  051775  051523
025640/ 026400  024142  051122  051122  051122  051122  175122  052124
025660/ 052124  052124  052124  052124  052124  052124  176524  052525
025700/ 026400  000000  052525  176525  050120  050120  050120  050120
025720/ 050120  050120  005375  051523  173523  051523  173523  052124
025740/ 026400  000000  050120  173520  050120  005120  051523  173523
025760/ 051523  173523  052124  173524  051122  173522  051122  173522
```

Figure MON-77. RSTS Dump (Cont.)


```
026300/ 000000  000000  173524  052525  173525  052525  176525  052120
026320/ 050120  050120  050120  050120  050120  005375  051775  051523
026340/ 000000  016100  051523  051523  052364  052124  051367  051122
026360/ 051122  051122  051122  051122  172122  052124  173524  052525
026400/ 000000  016040  075734  000012  171571  006571  062156  174000
026420/ 000404  022000  000000  001012  000225  002004  127600  001061
026440/ 000000  000000  075734  000012  043172  003161  061246  000400
026460/ 001002  006300  000000  001411  000053  001004  127600  001064
026500/ 000000  015740  005012  055412  055533  036074  036074  036074
026520/ 036074  036074  022044  022044  022044  022044  022044  020441
026540/ 000000  015700  020441  020441  020441  021443  021443  021443
026560/ 021443  021443  021443  037477  037477  037477  037477  036475
026600/ 000000  015640  075734  002012  017171  006571  062156  173776
026620/ 000404  015200  000000  007414  001205  002004  127600  001052
026640/ 000000  015600  023046  023046  023046  023046  025453  025453
026660/ 025453  025453  025453  022445  022445  022445  022445  022445
026700/ 000000  015540  075734  000006  164561  006171  062156  173776
026720/ 000404  023400  000000  001024  000565  002000  127600  000000
026740/ 000000  015500  075734  000012  000021  036400  062156  000000
026760/ 000004  020540  000000  000000  000000  000000  000000  000000
027000/ 000000  015440  000000  000000  000000  000000  000000  000200
027020/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  427. Duplicate Lines (027040 - 044300)  ******
044320/ 000000  000000  000000  012767  000100  000002  076600  000100
044340/ 010021  005215  005267  177766  026727  177762  000111  101766
044360/ 000207  065672  065672  065672  065672  065672  065672  065672
044400/ 065672  065672  065672  065672  065672  065672  065672  065672
044420/ 065672  000000  000000  000000  000000  000000  000000  000000
044440/ 000000  001112  006410  000121  100120  000740  000000  000000
044460/ 000000  000000  000020  000003  000000  000000  000000  000000
044500/ 000000  000000  000000  000000  000000  000000  000000  000000
044520/ 000000  000000  000000  172070  172040  176750  176700  176350
044540/ 176300  172474  172440  000003  002412  007353  002412  007354
044560/ 002412  007355  002412  007356  002412  007357  002412  007360
044600/ 002412  007361  002412  007362  002412  007363  002412  007364
044620/ 002412  007365  002412  007366  002412  007367  002412  007370
044640/ 002412  007371  002412  007372  002412  007373  002412  007374
044660/ 002412  007375  002412  007376  002412  007377  002412  007400
044700/ 002412  007401  002412  007402  002412  007403  002412  007404
```

Figure MON-77. RSTS Dump (Cont.)

```
046720/ 002412  007405  002412  007406  002412  007407  002412  007410
046740/ 002412  007411  002412  007412  002412  007413  002412  007414
046760/ 002412  007415  002412  007416  002412  007417  002412  007420
047000/ 002412  007421  002412  007422  002412  007423  002412  007424
047020/ 002412  007425  002412  007426  002412  007427  002412  007430
047040/ 002412  007431  002412  007432  002412  007433  002412  007434
047060/ 002412  007435  002412  007436  002412  007437  002412  007440
047100/ 002412  007441  002412  007442  002412  007443  002412  007444
047120/ 002412  007445  002412  007446  002412  007447  002412  007450
047140/ 002412  007451  010346  010246  005400  062700  007777  072027
047160/ 177764  042700  177760  016702  000160  022702  045432  103435
047200/ 010003  005722  001372  077303  010003  010442  077302  062702
047220/ 026542  006302  012622  010246  000405  016212  177774  062722
047240/ 020008  005501  010122  077007  012602  162702  170202  072227
047260/ 000003  000302  110201  105002  000404  012767  000020  000050
047320/ 012602  012603  000207  010046  016700  000036  020420  001002
047328/ 005068  177776  020027  045434  103771  012600  056767  000010
047348/ 020356  005067  000002  000207  000000  045374  177777  177777
047368/ 177777  177777  177777  177777  177777  177777  000000  000000
047408/ 020000  000000  000000  000000  000000  000000  000000  000000
047428/ 000000  000000  000000  000000  000000  000000  177777  012705
047448/ 073522  012701  073470  015137  172352  020127  073446  101407
047468/ 014502  001771  010146  004767  000042  012601  000764  016737
047508/ 040442  172352  014501  012100  016503  000032  011102  004767
047528/ 000012  020527  073470  101366  000137  104570  010546  012737
047548/ 034240  177776  004712  012737  034140  177776  012605  000207
047568/ 032711  001008  001433  022027  000172  103002  004737  111216
047608/ 172220  170127  040200  016720  020144  016720  020142  174020
047628/ 174120  174220  174320  172404  172505  174020  174120  170127
047648/ 042000  005067  020110  005067  020106  000207  005720  012720
047668/ 040003  005020  005020  000763  170127  040200  032711  001000
047708/ 001417  062700  000066  172540  172440  174105  174004  172740
047728/ 172640  172540  172440  014067  020026  014067  020020  170140
047748/ 000207  170367  020010. 052777  000040  133036  000137  104656
047768/ 016203  000010  016202  000014  004715  016000  000034  001413
046320/ 014002  001406  032702  000037  001403  042702  000037  004715
046028/ 030027  000035  001365  005725  000205  016000  000034  001413
046042/ 005710  001401  004715  062700  000012  030027  000037  001370
046060/ 016000  177740  001365  005725  000205  004537  110562  016700
245120/ 132716  162703  000002  004767  000006  010366  000010  000207
046128/ 004567  177706  002401  000421  005710  100017  111002  072227
046148/ 000015  020302  103412  066002  000010  020302  103006  017003
046168/ 000004  042703  000037  005126  012605  000207  016102  000006
046220/ 016220  000034  001444  026700  132614  001002  005067  132606
046228/ 005062  000034  014002  001417  012746  000001  122761  000002
046248/ 020025  001405  032702  000037  001402  012716  000401  042702
046268/ 002037  162662  000032  030027  000035  001354  005740  010446
046308/ 012004  011003  000004  000040  012604  005700  001367  000207
046328/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  31. Duplicate Lines (046340 - 047300)   ******
```

Figure MON-77. RSTS Dump (Cont.)


```
047520/ 052632  052726  052026  052274  052162  052250  052336  052626
047540/ 052512  052662  052666  052754  052642  052130  052216  052326
047560/ 052372  052460  052545  052524  052722  052512  052576  052154
047600/ 052252  052340  052426  052514  052602  052756  053044  053132
047620/ 056132  056220  056306  056374  056462  056550  056636  056724
047640/ 057012  057100  057166  057254  057342  057430  057516  057624
047660/ 057672  057760  060046  060134  060222  060310  060376  060464
047700/ 060552  060640  060726  061014  061102  061170  061256  061344
047720/ 061432  061520  061606  061674  061762  062050  062136  062224
047740/ 062312  062400  062466  062554  062642  062730  063016  063104
047760/ 063172  063260  063346  063434  063522  063610  063676  063764
048000/ 064052  064062  064076  064112  064126  064142  064156  064172
048020/ 064346  064426  064506  064566  064646  064726  065006  065046
048040/ 014640  002740  006440  007240  013003  013440  011040  012640
048060/ 013140  015100  015040  013640  022400  000200  000000  000000
048100/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  4. Duplicate Lines (047760 - 050040)   ******
050060/ 000000  000000  000000  000000  000000  000000  177777  000000
050100/ 020000  004000  000000  000000  000000  000000  000000  000000
050120/ 000000  000000  000000  004000  000000  000000  000000  000000
050140/ 000000  020000  020000  020000  020000  020000  020000  020000
050160/ 020000  020000  020000  020000  020000  020000  020000  020000
        ******  1. Duplicate Lines (050200 - 050260)   ******
050260/ 020000  020000  020000  020000  020000  020000  020000  000040
050300/ 020002  020002  020002  020002  020002  020002  020002  020002
050320/ 020002  020002  010000  020002  020002  000001  000003  002000
050340/ 000000  000000  000000  000003  000000  000000  000000  000000
        ******  4. Duplicate Lines (050360 - 050440)   ******
050460/ 000000  000000  000000  000000  000000  000000  000000  001100
050500/ 005040  001200  067206  000000  000000  000000  000060  000270
050520/ 010650  077470  074344  074360  074371  174444  000216  000003
050540/ 000020  000003  000000  000000  000000  000000  000000  000000
050560/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  5. Duplicate Lines (050600 - 050700)   ******
050720/ 000000  000000  175401  000000  000000  000000  000000  000000
050740/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  11. Duplicate Lines (050760 - 051220)   ******
051240/ 000000  000000  000000  000000  000000  000000  000000  000020
051260/ 000034  000063  000024  000000  000000  000000  177777  000000
051300/ 000000  000000  000000  000000  000000  000000  177777  000000
051320/ 177400  000000  111052  044630  051504  177401  000000  000000
051340/ 000000  045504  177400  000000  000000  000000  045504  177401
051360/ 000000  000000  000000  046504  177400  000000  000000  000000
051400/ 046504  177401  000000  111052  042632  051104  177400  000000
051420/ 040335  050500  051104  177401  000000  000000  000000  041104
051440/ 177400  000000  000000  000000  041104  177401  000000  024433
051460/ 000000  041104  177402  000000  000000  000000  054523  000000
051500/ 000000  000000  000000  000000  000000  000000  000000  000000
        ******  12. Duplicate Lines (051520 - 052000)   ******
052020/ 000000  000000  111146  000000  054523  000000  000410  077430
```

Figure MON-77. RSTS Dump (Cont.)

```
052440/ 000000 045653 000001 000000 000000 025725 000376 000376
052460/ 000000 000000 000000 000000 040106 040106 000000 000000
052500/ 121506 000111 000111 000000 000000 000000 000000 165141
052520/ 004621 000000 000000 070371 000000 000000 000000 000000
052540/ 000000 000000 000000 000001 000000 000000 000002 000000
052560/ 033730 000000 000000 000000 000000 027570 021370 000000
052600/ 000000 003110 000000 000000 000000 000000 000000 000000
052620/ 000420 000001 000000 000000 000000 000002 000000 000126
052640/ 020001 002001 000000 000003 000012 102605 000020 000000
052660/ 000036 000020 000000 106220 010000 000000 000000 000006
052700/ 020000 000000 000000 000000 006200 044400 000000 000000
052720/ 000002 000401 000124 020001 000000 000000 000000 000012
052740/ 044511 000000 000000 000006 000000 000000 122230 010010
052760/ 000000 000200 000006 020000 000000 100000 000000 122230
053000/ 044400 000000 000000 000002 001001 000150 020001 000000
053020/ 000000 000000 000012 044511 000000 000000 000006 000000
053040/ 000200 122230 010010 000000 000000 000006 020000 000000
053060/ 100000 000000 122230 044400 000000 000000 000002 001401
053100/ 000020 000000 000000 000000 000000 000012 044511 000000
053120/ 000000 000006 000000 000000 122230 010010 000000 000000
053140/ 000006 020000 000000 100000 000000 122230 044400 000000
053160/ 000200 000002 002001 000000 000000 000000 000000 000000
053200/ 000012 044511 000000 000000 000006 000000 000000 122230
053220/ 010010 000000 000000 000006 020000 000000 100000 000000
053240/ 122230 044400 000000 000000 000002 002401 000000 000000
053260/ 000200 000000 000000 000012 044511 000090 000000 000006
053300/ 000000 000000 122230 010010 000030 000000 000006 020000
053320/ 000000 100000 000000 122230 044400 000000 000000 000002
053340/ 003001 000200 000000 000000 000000 000000 000012 044511
053360/ 002000 000000 000006 000000 000000 122230 010010 000000
053400/ 000000 000006 020000 000000 100000 000000 122230 044400
053420/ 000000 000000 000002 003401 000000 000000 000000 000000
053440/ 000000 000012 044511 000000 000000 000006 000000 000000
053460/ 122230 010010 000000 000000 000006 020000 000000 100000
```

Figure MON-77. RSTS Dump (Cont.)


```
053500/ 000000 122230 044400 000000 000000 000002 000001 000000
053520/ 000000 000000 000000 000000 000012 044511 000000 000000
053540/ 000006 000000 000000 122230 010010 000000 000000 000020
053560/ 020000 000000 100000 000000 122230 044400 000000 000000
053600/ 000002 004401 000000 000200 000000 000003 122230 010010
053620/ 044511 000000 000000 000006 000000 100000 000000 122230
053640/ 000000 000000 000006 020000 000000 100000 000000 000000
053660/ 044400 000000 000000 000002 005001 000000 000000 000030
053700/ 000000 000000 000012 044511 000000 000200 000006 002000
053720/ 000000 122230 010010 000000 000000 000006 020000 000000
053740/ 100000 000000 122230 044400 000000 000200 020002 025401
053760/ 000000 000000 000000 000000 000000 000012 044511 000000
054000/ 000000 000006 000000 000000 122230 010010 000000 000000
054020/ 000006 020000 000000 100000 000000 122230 044400 000000
054040/ 000000 000002 006001 000000 000000 000000 000300 000000
054060/ 000012 044511 000000 000000 000006 020000 000200 122230
054100/ 010010 000000 000000 000000 000006 020000 000200 100000
054120/ 122230 044400 000000 000000 000002 006401 000000 000000
054140/ 000000 000000 000000 000012 044511 000000 000000 000006
054160/ 000006 000000 122230 010010 000000 000000 000006 020000
054200/ 000000 100000 000000 122230 044400 000000 000000 000002
054220/ 007001 000000 000000 000000 000000 000000 000012 044511
054240/ 000000 000000 000006 000000 000000 122230 010010 000000
054260/ 000000 000006 020000 000000 100000 000000 122230 044400
054300/ 000000 000000 000002 007401 000000 000000 000000 000000
054320/ 000000 000012 044511 000000 000000 000006 000000 000000
054340/ 122230 010010 000000 000000 000006 020000 000000 100000
054360/ 000000 122230 044400 000000 000000 000002 010000 000000
054400/ 000000 000001 000000 000000 000012 050521 000000 000000
054420/ 000006 000000 000000 146260 123014 000000 000200 020006
054440/ 020000 033503 000000 006307 006200 044400 000000 000000
054460/ 000002 010400 000000 000000 000001 000000 000000 000012
054500/ 050521 000000 000000 000006 020000 000000 146260 120414
054520/ 000000 000000 000006 020000 033503 000000 006307 006200
054540/ 044400 000000 000000 000002 011000 000000 000000 000001
054560/ 000000 000000 000012 050521 000000 000000 000026 000200
054600/ 000000 146260 121014 000000 000000 000006 020000 033503
054620/ 000000 006307 006200 044400 000000 000000 000000 011400
054640/ 000000 000000 000001 000000 000000 000012 050521 000000
054660/ 000000 000006 000000 000000 146260 121414 000000 000000
054700/ 000006 020000 033503 000000 006307 006200 044400 000000
054720/ 000000 000002 012000 000000 000000 000001 000200 000200
054740/ 000012 050521 000000 000000 000006 000000 000000 146260
054760/ 122014 000000 000000 000000 020000 033503 000000 006307
055000/ 006200 044400 000000 000000 000002 012400 000000 000000
055020/ 000001 000000 000000 000012 050521 000000 000000 000206
055040/ 000000 000000 146260 122414 000020 000000 000206 020000
055060/ 033503 000000 006307 006200 044400 000200 000000 000002
055100/ 013000 000000 000000 000001 000000 000000 000012 050521
055120/ 000000 000000 000006 000000 000000 146260 123014 000000
```

Figure MON-77. RSTS Dump (Cont.)

```
055140/ 000720  000006  020000  027303  000200  006307  006200  044420
055160/ 000000  000000  000002  013400  000000  000200  000001  000000
055200/ 000000  020012  050521  000200  003036  000000  000000  000000
055220/ 146260  123414  000000  000000  000006  020000  033503  000200
055240/ 000000  000000  044511  000000  000000  000002  014000  000000
055260/ 000000  000001  000000  000000  000012  050521  000000  000000
055300/ 000006  000000  000000  146260  124014  000000  000000  000006
055320/ 020000  026703  000000  006307  006200  044400  000000  000000
055340/ 000002  014400  000000  000000  000001  000000  000000  000012
055360/ 044511  000000  000000  000006  000000  000000  046260  124414
055400/ 000000  000000  000006  020000  026503  000000  006307  006200
055420/ 044400  001400  000000  000002  015000  000000  000000  000001
055440/ 000200  000000  000012  050521  000000  000000  000000  000000
055460/ 000000  146260  125014  000000  000000  000000  000002  015400
055500/ 000200  006307  006200  044400  000000  000000  000000  000000
055520/ 000200  000000  000001  000000  000000  000012  050521  000000
055540/ 000000  000006  000000  000000  146260  125414  000000  000000
055560/ 000000  020000  033503  000000  006307  006200  044400  000000
055600/ 000000  000002  016000  000000  000000  000001  000000  000000
055620/ 000012  050521  000000  000000  000006  000000  000000  146260
055640/ 126014  000000  000000  000006  020000  033503  000000  006307
055660/ 006200  044400  000000  000000  000002  016400  000000  000000
055700/ 000201  000000  000000  000012  050521  000000  000000  000006
055720/ 000000  000000  146260  126414  000000  000000  000006  020000
055740/ 026503  000000  006307  006200  044400  000000  000000  000002
055760/ 017000  000000  000000  000001  000000  000200  000012  050521
056000/ 000200  000000  000006  000000  000000  146260  127014  000000
056020/ 000000  000006  020000  033503  000000  006307  006200  044400
056040/ 000200  000000  000002  017400  000000  000000  000001  000000
056060/ 000000  000012  050521  000000  000000  000006  000000  000000
056120/ 146260  127414  000000  000000  020006  020000  033503  000000
056120/ 006307  006200  044400  000000  000000  000002  020000  000000
056140/ 000200  000001  000000  000000  000012  050521  000000  000000
056160/ 000006  000000  000000  146260  120014  000000  000000  000000
056200/ 020000  033503  000000  006307  006200  044400  000000  000000
056220/ 000200  020000  000000  000000  000001  000000  000000  000012
056240/ 050521  000200  000000  000006  000000  000000  146260  120414
056260/ 000000  000000  000000  020000  033503  000000  006307  006200
056300/ 044400  000000  000000  000002  021000  000000  000000  000001
056320/ 000000  000000  000012  050521  000000  000000  000006  000000
056340/ 000000  146260  121014  000000  000000  000006  020000  033503
056360/ 000000  006307  006200  044400  000000  006000  000002  021400
056400/ 000000  000200  000001  000000  000000  000012  050521  000000
056420/ 000200  000006  000000  000000  146260  121414  000000  000000
056440/ 000000  020000  033503  000000  006307  006200  044400  000000
056460/ 000000  000002  022000  000000  000000  000001  000000  000000
056500/ 000012  050521  000000  000000  000006  000000  000000  146260
056520/ 122014  000000  000000  000006  020000  026503  000000  006307
056540/ 006200  044400  000000  000000  000002  022400  000000  000000
056560/ 000001  000000  000000  000012  050521  000000  000000  000006
```

Figure MON-77. RSTS Dump (Cont.)


```
057100/ 000002  024400  000000  000200  000001  000000  000000  000012
057120/ 050521  000000  000000  000006  000000  146260  124414  000000
057140/ 000000  000000  000006  020000  027303  000000  006307  006200
057160/ 044400  000000  000000  000002  025000  000000  000000  000201
057200/ 000000  000000  000012  050521  000000  000000  000006  000000
057220/ 000000  146260  125014  000000  000000  000006  020000  026503
057240/ 000000  006307  006200  044400  000000  000200  000002  025400
057260/ 000000  000000  000001  000000  000000  000012  050521  000000
057300/ 000000  000006  000000  000000  146260  125414  000000  000000
057320/ 000000  020000  033503  000000  006307  006200  044400  000000
057340/ 000000  000002  026000  000000  002000  000001  000000  000000
057360/ 000012  050521  000000  000000  000006  000000  146260  126414
057400/ 126014  000000  000000  000006  020000  033503  000000  006307
057420/ 006200  044400  000000  000000  000002  026400  000000  000000
057440/ 000001  000000  000000  000012  050521  000000  000000  000006
057460/ 000000  000000  146260  126414  000000  000000  000005  020000
057500/ 033503  000000  006307  006200  044400  000000  000000  000002
057520/ 027000  000000  000000  000001  000000  000000  000012  050521
057540/ 000000  000000  000006  000000  000000  146264  127014  000000
057560/ 000000  000036  020000  027303  000000  006307  006200  044400
057600/ 000000  000000  000002  027400  000000  000000  000001  000000
057620/ 000000  000012  050521  000000  000000  000006  000000  000000
057640/ 146260  127414  000000  000000  000000  000005  020000  026503
057660/ 006307  006200  044400  000000  000000  000002  030000  000000
057700/ 000000  000001  000000  000000  000012  044511  000000  000000
057720/ 000000  000006  000000  000000  006200  120014  000000  000006
057740/ 020000  006307  000000  006307  006200  044400  000000  000000
057760/ 000002  030400  000000  000000  000001  000000  000000  000212
060000/ 044511  000000  000000  000006  000000  000000  006200  120414
060020/ 000000  000000  000006  020000  006307  000000  006307  006200
060040/ 044400  000000  000000  000002  031000  000000  000000  000001
060060/ 000000  000000  000000  000012  044511  000000  000000  000206
060100/ 000000  006200  121014  000000  000000  000306  020000  006307
060120/ 000000  006307  006200  044400  000000  000000  000002  031000
060140/ 000000  000000  000001  000000  000000  000012  044511  000000
060160/ 000000  000006  000000  000000  006200  121414  000000  000000
060200/ 000006  020000  006307  000000  006307  006200  044400  000000
060220/ 000000  000002  032000  000000  000000  000001  000000  000000
```

Figure MON-77. RSTS Dump (Cont.)

```
060240/ 000012 044511 000000 000000 000006 000000 000000 006204
060260/ 122014 000020 000000 000006 020000 006307 000300 006307
060300/ 006200 044400 000003 000030 000002 032400 000300 000330
060320/ 000031 003000 000000 000012 044511 000000 000000 000006
060340/ 003000 000000 006200 122414 000000 000000 000006 020000
060360/ 006307 000030 006307 006200 044400 000000 000000 000002
060400/ 033000 000020 000000 000001 000000 000000 000012 044511
060420/ 000000 000000 000006 000000 000000 006200 123014 000000
060440/ 000000 000026 020000 006307 000000 006307 006200 044400
060460/ 000000 000000 000002 033400 000000 000000 000001 000000
060500/ 000000 000012 044511 000000 000000 000006 000000 000000
060520/ 000200 123414 000000 000000 000006 020000 006307 000000
060540/ 006307 006200 044400 000000 000000 000002 034000 000000
060560/ 000020 000001 000000 000000 000012 044511 000000 000000
060600/ 000000 000000 000000 006200 124014 000000 000000 000006
060620/ 020000 006307 000000 006307 006200 044400 000000 000000
060640/ 000002 034400 000000 000000 000001 000000 000000 000012
060660/ 044511 000000 000000 000006 020000 006307 000000 006200
060700/ 000000 000000 000006 020000 006307 000000 006307 006200
060720/ 044400 000000 000000 000002 035000 000000 000000 000001
060740/ 000000 000000 000012 044511 000000 000000 000006 000000
060760/ 000000 006200 125014 000000 000000 000006 020000 006307
061000/ 000000 006307 006200 044400 000000 000000 000002 035400
061020/ 000020 000000 000001 000000 000000 000012 044511 000000
061040/ 000000 000006 000000 000000 006200 125414 000000 000000
061060/ 000006 020000 006307 000000 006307 006200 044400 000000
061100/ 000000 000002 036000 000000 000000 000001 000000 000000
061120/ 000012 044511 000000 000000 000006 000000 000000 006200
061140/ 126014 000000 000000 000006 020000 006307 000000 006307
061160/ 006200 044400 000000 000000 000002 036400 000000 000000
061200/ 000001 000000 000020 000012 044511 000000 000000 000006
061220/ 000000 006200 126414 000000 000000 000006 020000 000002
061240/ 006307 000000 006307 006200 044400 000000 000000 000012
061260/ 037000 000000 000000 000001 000000 000000 000012 044511
061300/ 000000 000006 000000 000002 006200 127014 000000 000000
061320/ 000000 000006 020000 006307 000000 006307 006200 044400
061340/ 000000 000000 000002 037400 000000 000000 000001 000000
061360/ 000000 000012 044511 000000 000000 000006 000000 000000
061400/ 006200 127414 000000 000000 000006 020000 006307 000000
061420/ 006307 006200 044400 000000 000000 000002 040000 000000
061440/ 000000 000001 000000 000000 000012 044511 000000 000000
061460/ 000000 000000 000000 006200 120016 000000 000000 000006
061500/ 020000 011070 000000 011070 006200 044400 000000 000000
061520/ 000002 040400 000000 000000 000001 000000 000000 000012
061540/ 102605 000000 000003 000006 000000 000000 166220 120416
061560/ 000000 000000 000006 020000 014031 000000 011071 006200
061600/ 044400 000003 000000 000002 041030 000150 000001 000031
061620/ 015142 015142 000011 050521 017402 017402 000005 000000
061640/ 033034 156260 121016 000000 000000 000006 020000 017032
061660/ 000200 011072 006200 044400 000000 000000 000002 041400
```

Figure MON-77. RSTS Dump (Cont.)


```
062100/ 000000 000000 000000 146260 122416 000000 000000 000002
062120/ 017035 000000 011075 006200 044400 000000 000000 000002
062140/ 043004 000151 000001 000001 013242 013242 000011 050521
062160/ 015302 015302 000005 000000 000014 146260 123016 015442
062200/ 015442 000005 120204 017036 000000 011076 006200 044400
062220/ 000000 000000 000002 043400 000155 000000 000001 000000
062240/ 000000 000012 050521 000000 000000 000006 000000 000000
062260/ 167620 123416 000000 000000 000006 020000 012727 000000
062300/ 011077 006200 044400 000000 000000 000002 044034 002114
062320/ 020001 000000 021442 021442 000011 044511 013602 013602
062340/ 000005 000000 000000 046260 160016 000000 000000 000006
062360/ 020000 012430 000000 012430 006200 050400 000000 000000
062400/ 000002 044400 000000 000000 000000 000000 000000 000012
062420/ 050511 000000 000000 000006 000000 000000 006200 160416
062440/ 000000 000000 000006 020000 012431 100000 012431 000200
062460/ 050400 000000 000000 000002 045000 000000 000000 000000
062500/ 000000 000000 000000 000012 050511 000000 000000 000006
062520/ 000000 006200 161016 000000 000000 000006 020000 012432
062540/ 100000 012432 006200 050400 000000 000000 000000 000000
062560/ 000000 000000 000000 000000 000000 000012 050511 000000
062600/ 000000 000006 000000 000000 006200 161416 000000 000000
062620/ 000006 020000 012433 100000 012433 006200 050400 000000
062640/ 000000 000002 046000 000000 000000 000000 000000 000000
062660/ 000012 050511 000000 000000 000006 000000 000000 006200
062700/ 162016 000000 000000 000006 020000 013434 100000 013434
062720/ 006200 050400 000000 000000 000002 046400 000300 000300
062740/ 000000 000000 000000 000012 050511 000000 000000 000205
062760/ 000000 000000 000000 006200 162416 000000 000000 020000
063000/ 013435 100000 013435 006200 050400 000000 000000 000002
063020/ 047000 000000 000000 000000 000000 000000 000012 050511
063040/ 000000 000000 000000 000006 000000 000000 006200 163016
063060/ 000000 000006 020000 012436 100000 012436 006200 050400
063100/ 000000 000000 000000 000002 047400 000200 000200 000000
063120/ 000000 000012 050511 000000 000000 000005 200000 000000
063140/ 006200 163416 000000 000000 000006 020000 012437 100000
063160/ 012437 006200 050400 000000 000000 000002 056000 020000
063200/ 000000 000001 000000 000000 000012 044511 000000 000000
063220/ 000006 000000 000000 006200 120016 000000 000000 000000
063240/ 020000 011070 000000 011070 006200 044400 000000 000020
063260/ 000002 050400 000000 000000 000001 000000 000000 000012
063300/ 044511 000000 000000 000006 000000 000000 006200 120416
063320/ 000000 000000 000006 020000 011071 000000 011071 006200
063340/ 044400 000000 000000 000002 051000 000000 000000 000001
```

Figure MON-77. RSTS Dump (Cont.)

```
063360/ 000000 000000 000012 044511 000000 000000 000006 000000
063400/ 000030 006200 121016 000000 000000 000006 020000 011072
063420/ 000200 011072 006200 044400 000000 000000 000002 051400
063440/ 000200 030000 000001 000300 000000 000012 044511 000000
063460/ 000300 000005 000000 000000 006200 121416 000000 000000
063500/ 000006 020000 011073 000000 011073 006200 044400 000000
063520/ 000000 000002 052000 000000 000000 000001 000000 000000
063540/ 000012 044511 000000 000000 000006 000000 000000 006200
063560/ 122016 000000 000000 000006 020000 011074 000000 011074
063600/ 006200 044400 000000 000000 000002 052400 000000 000000
063620/ 000001 000000 000000 000012 044511 000000 000000 000006
063640/ 000200 000000 006200 122416 000000 000000 000006 020000
063660/ 011075 000300 011075 006200 044400 000000 000000 000002
063700/ 053000 000000 000000 000001 000000 000000 000012 044511
063720/ 000000 000000 000006 000000 000000 006200 123016 000000
063740/ 000000 000006 020000 011076 000000 011076 006200 044400
063760/ 000000 000000 000002 053400 000000 000000 000001 000000
064000/ 000000 000012 044511 000000 000000 000006 000000 000000
064020/ 006200 123416 000000 000000 000006 020000 011077 000000
064040/ 011077 006200 044400 000000 000000 000004 000000 000117
064060/ 000001 000006 000000 000124 000000 000002 000000 000006
064100/ 002400 000150 000030 000004 000000 000006 001000 000000
064120/ 000300 000006 000000 000006 001400 000000 000000 000010
064140/ 000020 002036 002000 000000 000000 000012 000000 000006
064160/ 002402 000000 000000 000014 000000 000006 003000 000000
064200/ 000020 000016 000000 000006 003400 000000 000000 000020
064220/ 000030 000006 004000 000000 000000 000022 000000 000006
064240/ 004400 000000 000000 000024 000000 000006 005000 000000
064260/ 000030 000026 000000 000006 005400 000000 000000 000030
064300/ 000000 000006 006000 000000 000000 000032 000000 000006
064320/ 006400 000000 000000 000034 000000 000006 007000 000000
064340/ 000000 000036 000030 000010 000003 000000 040001 102000
064360/ 001035 000000 015340 136210 000000 000000 000000 077104
064400/ 132004 015240 000000 000000 000000 000000 001400 000000
064420/ 000000 000000 000000 000010 000403 000000 040001 102000
064440/ 001035 000000 020000 134410 000000 000000 000000 077104
064460/ 132004 017700 000000 000000 000000 000000 001400 000000
064500/ 000000 000000 000000 000010 001003 000000 040001 102000
064520/ 001035 000000 020440 132610 000000 000000 000000 077104
064540/ 132004 013500 000000 000000 000000 000000 001400 000000
064560/ 200000 000000 000300 000010 001403 000000 040001 102000
064600/ 001035 000000 021000 131010 000000 000000 000000 077104
064620/ 132004 020700 000000 000000 000000 000000 001400 000000
064640/ 000000 000000 000000 000010 002000 000000 000000 000000
064660/ 000000 000000 000000 000000 000000 000000 000000 000000
064700/ 000000 000000 000000 000000 000000 000000 000000 000000
064720/ 000200 000000 000000 000010 002400 000000 000000 000000
064740/ 000000 000000 000000 000000 000000 000000 000000 000000
064760/ 000000 000000 000000 000000 000000 000000 000000 000000
065000/ 000000 000000 000000 001012 000000 000117 000000 064014
065020/ 013202 013202 000023 102605 041102 004300 000026 014400
```

Figure MON-77. RSTS Dump (Cont.)


```
065240/ 000000 000000 000000 000014 000000 000123 000000 000001
065260/ 000000 000000 000000 077206 017312 000001 000000 000361
065300/ 000004 000700 000000 000000 021614 021540 000014 000426
065320/ 000125 100000 000001 000000 000000 000000 077206 016156
065340/ 000001 000001 000361 001402 050664 000514 000000 014754
065360/ 014600 000014 001000 000000 000400 000000 000000 000000
065400/ 000000 000000 000000 000000 000000 000000 000000 000000
065420/ 000000 000000 000000 000000 000000 000000 000000 000000
065440/ 000000 000000 000000 000000 000000 000000 000000 000000
065460/ 000000 000000 000000 000000 000000 000000 000000 000000
065500/ 000000 000000 000000 000000 000000 000000 000000 000000
065520/ 000000 000000 000000 000020 000401 000000 000000 000000
065540/ 000000 000000 000000 000000 000000 000000 000000 000000
065560/ 000000 000000 000000 000000 000000 000000 000000 000000
065600/ 000000 000000 000000 000000 000000 000000 000000 000000
065620/ 000000 000000 000000 000000 001312 000000 000000 000000
065640/ 104004 000000 000000 000000 000000 000000 000000 000000
065660/ 000000 000000 000000 000000 002004 100716 000001 100000
065700/ 000000 120000 000000 000000 000000 000302 000000 000000
065720/ 020000 000000 000000 000454 000000 000001 013540 000000
065740/ 000000 000000 000000 000000 103112 065746 103014 000270
065760/ 000000 000000 100000 000000 000000 012040 000000 000000
066000/ 000001 000000 000000 035117 032462 000066 031000 040000
066020/ 033455 000071 000000 000000 000000 000000 000000 000000
066040/ 177750 001002 001002 001002 001002 001002 001002 001002
066060/ 001002 001034 001002 001002 001002 020040 001002 001002
066100/ 001002 013026 001002 001002 001002 001002 001002 001002
066120/ 001002 013026 001002 001002 001002 013026 001002 001002
        ****** 2. Duplicate Lines (066140 - 066160)          ******
066200/ 001002 013026 001002 001036 001002 013026 001002 001002
066220/ 001002 013026 001002 001002 001002 016034 016002 020040
066240/ 001002 016034 001002 021042 001002 001002 001002 001002
066260/ 001002 001002 001002 001002 001002 002026 001002 001002
066300/ 001002 001002 001002 001002 001002 001002 001002 001002
        ****** 4. Duplicate Lines (066320 - 066400)          ******
066420/ 001002 001002 001002 001002 001002 001032 001002 001002
066440/ 017002 012300 000006 165151 000070 000021 000000 000000
066460/ 000000 000000 000012 000341 000300 000020 000406 000361
066500/ 000310 000025 000000 000000 173440 105500 000006 166715
066520/ 005571 150000 150000 177404 000000 000000 001020 000574
066540/ 002000 127600 000000 177776 177777 177777 177777 000000
066560/ 000000 000000 000000 177776 000777 000000 000000 177776
066600/ 177777 177777 177777 000000 000000 000036 000000 000040
066620/ 177777 107740 010567 000400 000001 140706 140644 003740
```

Figure MON-77. RSTS Dump (Cont.)

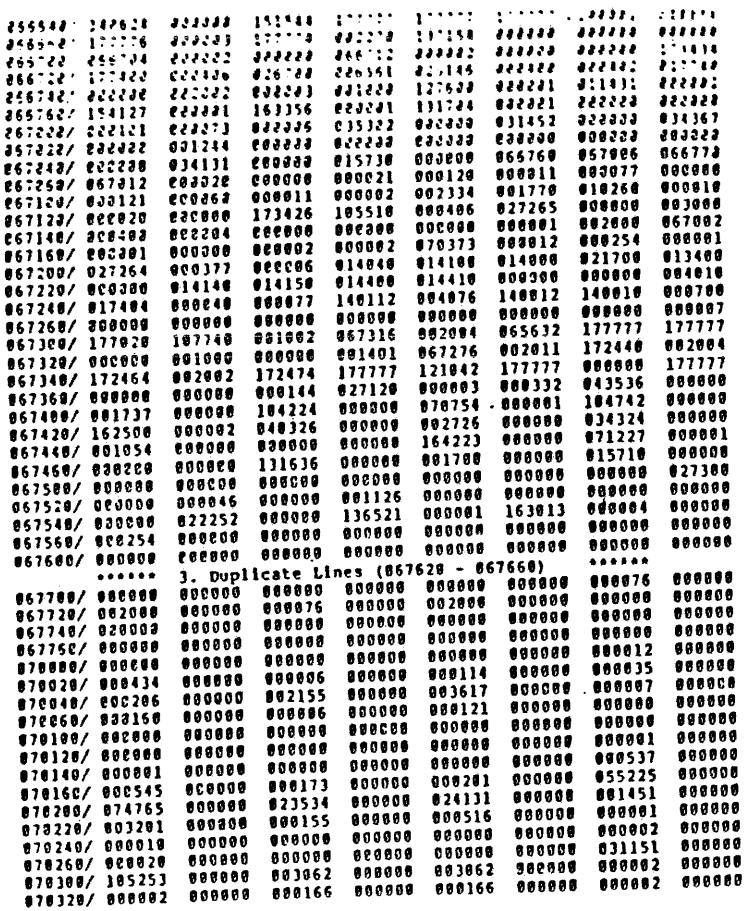


Figure MON-77. RSTS Dump (Cont.)

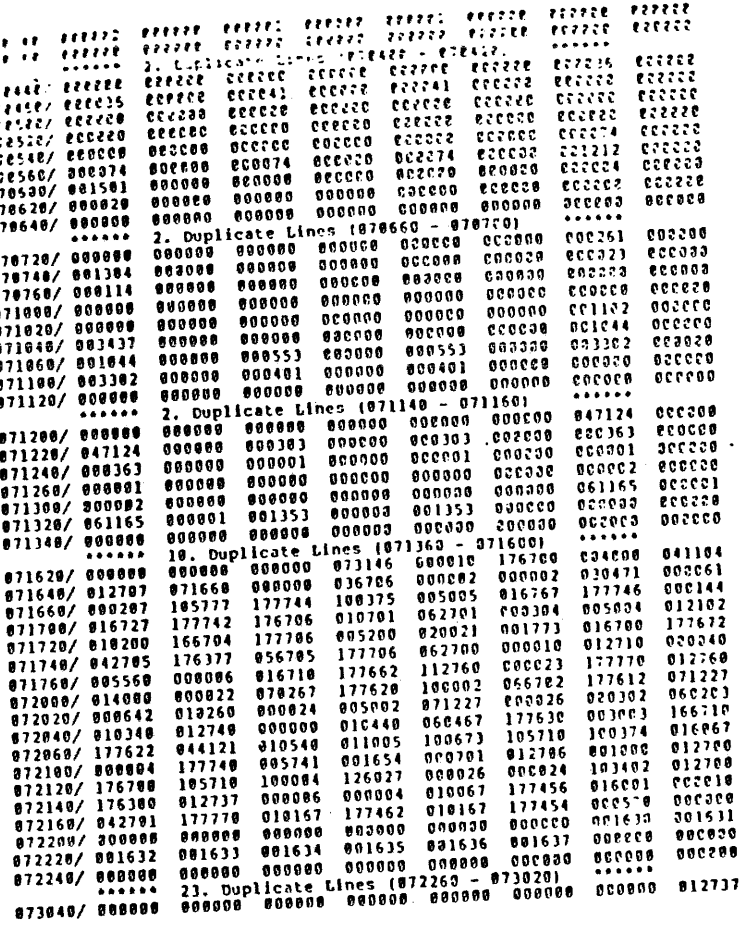


Figure MON-77. RSTS Dump (Cont.)

```
073060/ 030340  177776  005037  177572  005077  012370  012706  071630
073100/ 004767  000102  100037  000167  100742  012737  074340  177776
073120/ 005037  177572  005277  012330  012706  071630  005767  177854
073140/ 001402  004767  176514  027727  012310  177777  001354  016746
073160/ 124564  004767  000020  100775  026726  105606  001403  162737
073200/ 000004  000026  005007  116767  176442  176442  012767  000400
073220/ 176424  005067  176766  000167  176430  054523  041504  043104
073240/ 051504  045504  046104  046504  050104  051104  041104  041113
073260/ 046116  045520  046530  050114  046515  052104  054104  052124
073300/ 052115  054504  177777  000300  177777  177777  000001  000001
073320/ 177777  000001  177777  000001  000002  000127  000000  000016
073340/ 000005  000200  000002  000001  000001  073260  073272  073276
073360/ 052246  052222  052222  052222  052226  052232  052232  052236
073400/ 052236  052242  047320  047600  047602  047640  047654  047656
073420/ 047664  047670  000001  000002  000001  000004  000010  000020
073440/ 000001  000001  000001  075740  076642  075740  076642  077014
073460/ 077172  077202  077216  077236  075626  075646  075666  075706
073500/ 075726  000000  000000  000000  000000  000000  126140  000000
073520/ 131202  172040  000000  177440  176300  176700  177560  065536
073540/ 065536  065536  065536  065536  065536  065536  065536  065536
073560/ 065536  065536  065536  065536  065536  065536  160020  160020
073600/ 160020  160020  160020  160020  160020  160020  160020  160020
073620/ 160020  160020  160020  160020  160020  160020  160040  160040
073640/ 160040  160040  160040  160040  160040  160040  160040  160040
073660/ 160040  160040  160040  160040  160040  160040  160060  160060
073700/ 160060  160060  160060  160060  160060  160060  160060  160060
073720/ 160060  160060  160060  160060  160060  160060  160250  160250
073740/ 160250  160250  160250  160250  160250  160250  160260  160260
073760/ 160260  160260  160260  160260  160260  160260  160270  160270
074000/ 160270  160270  160270  160270  160270  160270  000000  000000
074020/ 000000  000000  000000  000000  000000  000000  000000  000000
074040/ 000000  000000  000000  000000  000000  000000  160150  160160
074060/ 160170  160200  160210  160220  177514  172440  172440  172440
074100/ 177342  177342  000000  000000  000000  000000  000002  000002
074120/ 000004  000004  000006  000006  000010  000010  000010  000002
074140/ 000003  177777  177777  000000  000000  020024  020024  020022
074160/ 024022  024022  004000  004000  000000  000000  064740  064740
074200/ 100220  100220  121510  121510  121510  000000  000000  000000
074220/ 000300  000000  000000  000000  000000  000000  000000  000000
074240/ 000000  000000  000000  000000  000000  130060  122614  122606
074260/ 121536  121720  122762  122730  076716  076716  076716  076716
074300/ 076716  121044  121174  120712  120610  120750  127750  121156
074320/ 121412  121252  120460  122464  120574  130476  127440  126332
074340/ 130176  127752  136176  075554  130634  000000  077100  126150
074360/ 126160  125102  124042  124736  131440  000000  000000  000000
074400/ 000300  000000  000000  000000  123676  125616  135432  000000
074420/ 101744  045436  133440  122530  000000  120412  000000  117644
074440/ 101210  135334  132010  101210  000000  077014  077202  077202
074460/ 077216  077216  077216  077236  000000  000000  000000  000000
074500/ 000000  000000  000000  101430  106144  075740  000000  075740
```

Figure MON-77. RSTS Dump (Cont.)


```
074520/ 075740  101430  105610  000000  077244  000020  075740  101430
074540/ 101430  101430  101430  107632  107624  107606  107624  107624
074560/ 107624  107500  107516  002000  000000  003203  125242  120052
074600/ 121010  000030  122372  124112  030000  076730  077230  002300
074620/ 125456  124174  131114  116466  123120  121026  121246  077034
074640/ 121454  126030  124240  131256  000001  124506  121024  121106
074660/ 077040  121254  177775  177777  131152  000301  124636  121024
074700/ 121172  077044  122260  177775  177777  131252  120314  133722
074720/ 121020  120774  077054  121252  125760  123506  130460  000000
074740/ 133500  121022  121104  077064  121434  125236  123510  131030
074760/ 000000  133500  121024  121104  077070  121404  125266  123542
075000/ 131030  000000  121734  121024  121104  077050  122150  125274
075020/ 124154  131034  000000  130754  000030  121054  077060  000000
075040/ 000000  000000  000000  100000  074402  000426  070420  030430
075060/ 035406  000016  000004  100422  000107  000000  000107  000107
075100/ 000107  000000  000107  000107  000107  001000  000200  000002
075120/ 000200  001000  000200  001000  000776  001000  177777  131177
075140/ 101163  177777  177777  177777  030007  000003  000001  177777  000007
075160/ 177777  177777  177777  101003  101163  101063  137372  130760
075200/ 134650  147000  147234  142420  144710  135172  137332  142642
075220/ 014000  014424  014520  134434  134634  133446  130742  130642
075240/ 140220  140556  141452  066000  143376  033310  062323  000301
075260/ 000001  000001  000001  145477  035001  024260  062225  033573
075300/ 024541  145325  145012  024263  000001  043155  145564  000001
075320/ 035631  065424  000001  144712  145507  145505  024001  024565
075340/ 026275  025165  033001  140221  034167  145215  022411  135172
075360/ 014000  014424  014520  023000  022000  142420  041000  142642
075400/ 000001  065000  145624  062001  000001  063007  145113  143276
075420/ 146330  000301  000000  000377  000000  177400  000000  000000
075440/ 177777  000377  177400  000377  177777  177400  177777  000000
075460/ 000000  000000  177400  000000  000000  000000  000000  000000
075500/ 177400  177777  000000  000377  000377  067552  067364  071614
075520/ 066754  125012  125446  133572  133506  125122  126126  126420
075540/ 123026  121736  122220  120736  121564  005000  013737  101634
075560/ 172352  000137  120720  054110  053020  053530  053000  046000
075600/ 047000  051266  052000  045032  050000  051000
```

Figure MON-77. RSTS Dump (Cont.)

FEEDBACK FOR MONITOR

Using the map and octal dump provided, answer the following questions.

1. Is 'JOB' attached to any resident libraries? No

   - Look at address 1026 in the dump. This is JOBWDB in low memory which contains a pointer to the WDB @ W.WIN1 for 'JOB'.
   - Since the value of JOBWDB is 0, the job has no resident libraries attached.

2. Have any resident libraries been added? Yes

   - Look at the map under the RAWMEM section and find the label LIBLST. The address associated with LIBLST is 50500.
   - Now look at address 50500 (LIBLST) in the dump. This address contains 5040. Since the value is non-zero, at least one resident library has been added on the system. The value 5040 is a pointer to the first resident library description block (LIB).

   a. How many resident libraries have been added?

   1

   - Look at address 5040 in the dump. The value at this address is the link word for the resident library description block. Since the word contains a 0, there is only one LIB block in the list.

   b. Answer the following questions about the library which is first in the list:

   Resident library name RMSRES

FEEDBACK

- Look at addresses 5042 and 5044. They contain 71233 and 70533 respectively, which is the resident library name (R.NAME), in RAD50. Using the RAD50 table, the library name is RMSRES. (Remember all subtraction is octal).

```
71233
70200 - R
 1033
 1010 - M
   23
   23 - S
    0

70533
70200 - R
  333
  310 - E
   23
   23 - S
    0
```

Its project, programmer number [0,1]

- Look at address 5046 (L.PPN) which contains 000001. This value has the project number in the high byte and the programmer number in the low byte.

```
00000000/00000001
   0        1
```

Its size in Kwords (decimal) 23kw

- Look at address 5056 (R.KS1Z) which contains 1027 in octal. The low byte of this word is the size of the resident library.

27 (in octal) = 23 (in decimal)

FEEDBACK

Its protection code 42

- Look at address 5074 which contains 025200. The library status flags are in the low byte and the library protection code is in the high byte.

L.PROT L.STAT
00101010/10000000

52 (octal) = 42 (decimal)

Which resident library flags are set. Save the load address.

- Look at address 5076 (R.FLAG) which contains 040000.

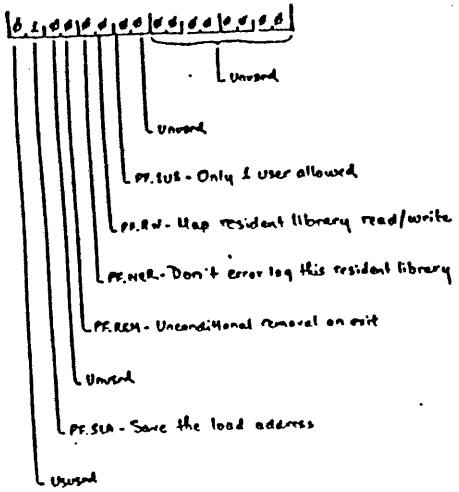


Figure MON-78.

FEEDBACK

Load address of library (octal) 1320000

- Look at address 5060 (M.PHYA) which contains 13200. This is the actual physical address at which the library will load divided by 100 octal.

3. Are large files sysgenned on the system? Yes

   - Look at the map. Since the section SYDLRG appears in the map, this is a large file system.

5. Considering the first file open on the first disk unit of the system:

> Note
> To find the address of the FCB for the first file:
>
> - Look at the map under the RAWMEM section and find the label FCBLST. The value 52352 is the address of the FCB table.
> - Starting at address 52352 in the dump, scan for the first non-zero word. This is at address 52366 which contains 13740.
> - The 13740 is the address of the FCB for the first file.

FEEDBACK

a. What is the file's PPN? (40,40)

- The file's PPN is the third word in the FCB. At address 13744 is the value 24050 which is the PPN.

PROJ	PROG
00101000	00101000
50	50

> Note
>
> 50 (octal) = 40 (decimal)

b. What is the filename and extension? DNDA14.TMP

- The filename and extension are stored in RAD50 format starting at address 13746.
- The values 15464, 5472 and 77430, converted from RAD50 using the table provided, produce DNDA14.TMP.

c. Is the file placed? No

- Look at address 13754 which contains 36024. The low byte of the word is the F$STAT byte.

FEEDBACK

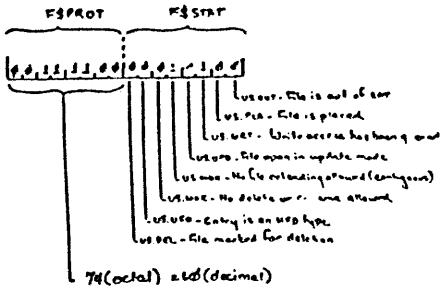


Figure MON-79.

d. Has write access been given out? Yes

e. Is the file open for update? No

f. Is the file contiguous? Yes

g. Is the file marked for deletion? No

h. What is the protection code? 60

- Look at the high byte shown in c (F$PROT) above. The 74 octal is 60 decimal.

i. What is the access count for opens in normal or update mode? 1

- Look at address 13756 which contains 000001. The low byte of the word (F$ACNT) contains the access count for normal or update mode.

j. What is the clustersize? 8

- Address 13744 contains the clustersize.

k. What is the octal address of the first WCB for the file? 17740

FEEDBACK

- Address 13776 contains the WCB address.

6. What disk types and how many units of each are sysgenned on this system?

Disk	Number
DS	2
DK	2
DM	2
DR	2
DB	3

There are two ways to find the answer.

a.
- Look at the .ABS. section of the map for symbols of the form CNT.xx where xx is a disk type. (i.e. CNT.DB)
- The value of these symbols is the number of units sysgenned.

b.
- Look at the map under the TBL section for the label DEVCNT which contains the address of the device count table 73310.
- Starting at address 73310 the first nine words contain: -1, -1, 1, 1, -1, 1, -1, 1, 2. These are the highest valid unit numbers on this system for each disk type supported by RSTS/E.
- Referring to the DEVNAM table in the module, the first nine words represent the disk types supported by RSTS/E (DC, DF, DS, DK, DL, DM, DP, DR, DB)

FEEDBACK

- Since the entries in DEVNAM and DEVCNT correspond directly to each other:

DEVNAM entry	DEVCNT entry	# of units
DC	-1	0
DF	-1	0
DS	1	2
DK	1	2
DL	-1	0
DM	1	2
DP	-1	0
DR	1	2
DB	2	3

7. Considering DB2:

To answer this question you must first get the UNTCNT entry for DB2: on this system. To get this information:

- Look at the map under the TBL section for the label DEVPTR which contains 73362, the address of the device pointer table.
- Starting at address 73362 count in 9 words. The value (52242) is the address in the UNTCNT table for DB0:
- Starting at address 52242 count in 3 words (this is the offset from DB0: to DB2: in the UNTCNT table). The value is 004017, which is the unit count status information for DB2:

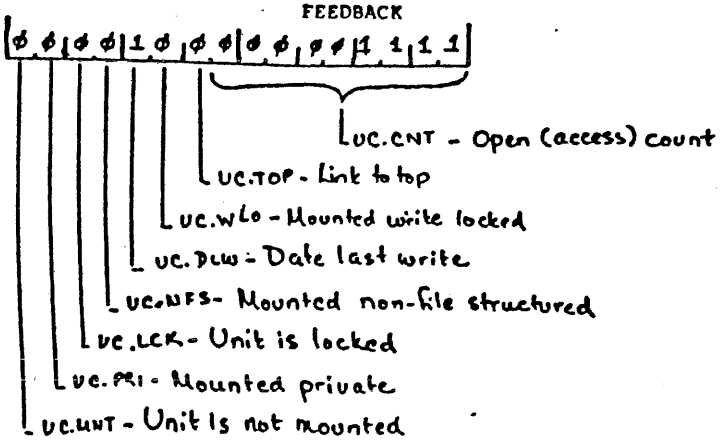


Figure MON-80.

a. Is it mounted? Yes

b. Is it a date of last write disk? Yes

c. Are files added at the top of the directory? No

d. What is the current open count (octal)? 17

e. What is the pack clustersize and error count? 8

0

To answer this question you must:

a. Determine the FIP unit number*2 for DB2: on this system.

   - Using the information in question 6, you know that the configured disks are:

FEEDBACK

Disk	FUN(octal)	FUN*2(octal)
DS0:	0	0
DS1:	1	2
DK0:	2	4
DK1:	3	6
DM0:	4	10
DM1:	5	12
DR0:	6	14
DR1:	7	16
DB0:	10	20
DB1:	11	22
DB2:	12	24

Note

The order of the disks is very important. They must be in the order in which they appear in the DEVNAM table. Also the FUN begins with 0.

   - Therefore, the FUN for DB2: is 12 and FUN*2 is 24.

b. Using the FUN*2 for DB2:, offset into the UNTCLU table to get the pack clustersize in the low byte and the error count in the high byte.

   - Look in the map under the RAWMEM section for the label UNTCLU which contains the value 52276. This is the address of the UNTCLU table.

   - To offset into the table by FUN*2, add 24 to 52276 (remember to do octal addition) to get the address 52322.

   - Look at address 52322 in the dump which contains 000010.

FEEDBACK

```
UNTERR  UNTCLU
00000000/00001000
   0       10
```

- The pack clustersize=10 (octal) which is 8 in decimal.
- The error count is 0.

8. What is the device designator of the system disk?

DB2:

To answer this question, you must remember that the word just before the DEVPTR table points to the entry in the UNTCNT table for the system disk (see the module).

- Since the address of the DEVPTR table is 73362 (see question 7), look at address 73360 which contains 52246.
- Starting an address 73362 (DEVPTR) scan the first nine words for the number which is the closest to but not greater than 52246.

  The ninth word (52242) is the one you're looking for.
- From question 7, this entry points from the DEVPTR table into the UNTCNT table for DB0:
- Therefore:

```
52242 ------> DB0:
52244 ------> DB1:
52246 ------> DB2:
```

RSTS/E V7.0 SWS UPDATE

SOF

Software and RSTS/E V7.0

Software and RSTS/E V7.0

CONTENTS

Introduction..... SOF-1

BASIC-PLUS and RSTS/E V7.0..... SOF-3
- General..... SOF-3
- Commands..... SOF-3
- Statements..... SOF-4
- Functions..... SOF-6
- System Function Calls..... SOF-19
  - File Name String Scan SYS Call..... SOF-19
  - Get Monitor Tables II SYS Call..... SOF-20
  - Change System Date/Time SYS Call..... SOF-21
  - Directory Lookup SYS Call..... SOF-21
  - ASSIGN/REASSIGN SYS Call..... SOF-22
  - DEASSIGN SYS Call..... SOF-26
  - Create a Job SYS Call..... SOF-28
  - Exit and Clear Program SYS Call..... SOF-30
  - Wild Card PPN SYS Call..... SOF-31
  - Return Job Status SYS Call..... SOF-33
  - Execute CCL SYS Call..... SOF-38
  - Data Caching SYS Calls..... SOF-40
  - File Utility SYS Call..... SOF-45
  - Resident Library SYS Calls..... SOF-53
- Implementing New RSTS/E Features Under BASIC-PLUS..... SOF-63

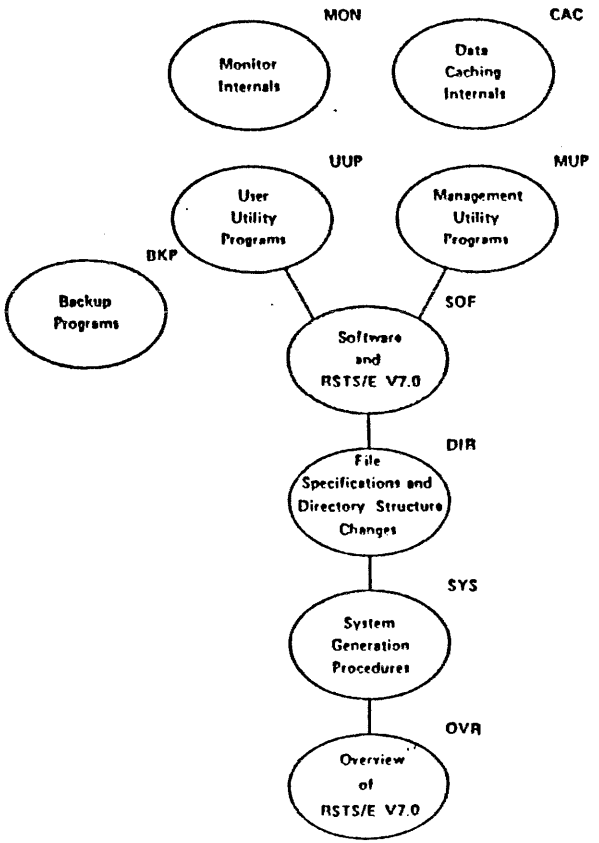
RMS V1.8 and RSTS/E V7.0..... SOF-80
- General..... SOF-80
- Implementing New RSTS/E Features Under RMS V1.8..... SOF-81

BASIC-PLUS-2 and RSTS/E V7.0..... SOF-82
- Commands..... SOF-82
- Statements..... SOF-84
- System Function Calls..... SOF-84
- Implementing New RSTS/E Features Under BASIC-PLUS-2 V1.60..... SOF-85
- Using the RMS Resident Library..... SOF-86

COBOL V4 and RSTS/E V7.0..... SOF-89
General..... SOF-89
Implementing New RSTS/E Features Under COBOL V4..... SOF-92
Using the RMS Resident Library..... SOF-93

MACRO and RSTS/E V7.0..... SOF-100
General..... SOF-100

COURSE MAP



Software and RSTS/E V7.0

INTRODUCTION

The RSTS/E Operating System supports a variety of languages. With this release of RSTS/E, each of the supported languages is affected in different ways. New features are implemented, enhancements are included to improve performance, or additional support is supplied.

This module covers the following software as supported on RSTS/E V7.0.

- BASIC-PLUS
- RMS-11 V1.8
- BASIC PLUS-2 V1.60
- COBOL V4
- MACRO-11

This page is for notes.

BASIC-PLUS and RSTS/E V7.8

General

Virtual Arrays

- I/O buffers larger than 512 bytes

Commands

COMPILE

- Defaults to conditionally contiguous .BAC files

ASSIGN/REASSIGN

- ASSIGN DEV:/MODE:1 Seizes the device (privileged)
- ASSIGN DEV:[ppn]LIB Associates the logical name LIB with a device and account number

> **Note**
>
> Making a user logical PPN assignment reduces the number of user logical assignments to three.

EXIT

- Returns to private default run-time system

DUMP

- Dumps user space
- Available as an optional patch

Statements

CLOSE -N%

- Device dependent close operations not performed
- Pending I/O buffers not written
- Close a tentative file without making it permanent
- Example

```
1000    ON ERROR GOTO 19000
1010    OPEN "VIRT.DAT" AS FILE 1%, RECORDSIZE 1024%    &
        \DIM #1%, P.ART$(3%)=256%                       &
        \PRINT "I/O BUFFERSIZE =";BUFSIZ(1%)            &
        \PRINT
1020    FOR I% = 0% TO 2% &
        \P.ART$(I%) = STRING$(256%,I%+65%) &
        \NEXT I%
32766   CLOSE -1%
32767   END

Ready

RUNNH
I/O BUFFERSIZE = 1024


Ready

DIR VIRT.DAT /
 Name .Ext    Size   Prot    Date        SY:[100,7]
VIRT  .DAT       0   < 60> 28-Nov-78

Total of 0 blocks in 1 file in SY:[100,7]

Ready
```

Figure SOF-1. CLOSE with Negative Channel

CHAIN

- Keyword LINE can be optionally included
- Example

```
1000   CHAIN "CCPOS" LINE 900
32767  END

Ready
```

Figure SOF-2. CHAIN

Functions

CCPOS

- o Equivalent to POS(N%)
- o Example

```
900     P.RINTSTRING$ = "THIS IS THE FIRST LINE OF THE HEADER"
1000    PRINT "...USING THE POS FUNCTION..."                  &
        \PRINT P.RINTSTRING$;POS(0%)                          &
        \PRINT                                                &
        \PRINT "...USING THE CCPOS FUNCTION..."               &
        \PRINT P.RINTSTRING$;CCPOS(0%)
32767   END

Ready
```

```
RUNNH
...USING THE POS FUNCTION...
THIS IS THE FIRST LINE OF THE HEADER 36

...USING THE CCPOS FUNCTION...
THIS IS THE FIRST LINE OF THE HEADER 36

Ready
```

Figure SOF-3. CCPOS

DEF*

- o Equivalent to DEF
- o Example

```
1000    DEF* FNA(X) = X**2 + 3*X + 25
2000    FOR I = 1 TO 4 STEP .5                                &
        \PRINT I,,FNA(I)                                      &
        \NEXT I
32767   END

Ready

RUNNH
 1                          29
 1.5                        31.75
 2                          35
 2.5                        38.75
 3                          43
 3.5                        47.75
 4                          53

Ready
```

Figure SOF-4. DEF*

SPEC%

- Performs special operations on disk, floppy disk, terminals and magnetic tape.
- Disk
  - Explicitly locks up to 7 blocks of a file open in update mode
  - Releases explicit and implicit locks on a file block(s)

For disk devices, the SPEC% function allows you to explicitly lock a maximum of seven disk block ranges on a file that is open for update. A locked range of blocks is one that cannot be accessed by another user or from another channel. Thus, SPEC% extends the use of guarded update, which locks the last block or blocks read on a file.

SPEC% also allows you to release explicit or implicit locks. Note that when you close a file, all explicit and implicit locks are released for that file.

The SPEC% function for disk files has the following format:

```
VALUE%=SPEC%(FUNCTION%, BLOCK, CHANNEL%, HANDLER INDEX%)
```

where:

VALUE% depends on the particular function code you specify in FUNCTION%. In most cases, VALUE% is equal to the BLOCK parameter.

FUNCTION% is a function code that specifies the desired operation. During normal I/O operations, a block, or range of blocks is implicitly locked when you read the file with a BASIC-PLUS GET statement. The SPEC% function allows you to convert implicit locks to explicit locks and to release selected locked blocks. The code specified in FUNCTION% determines the use of SPEC%.

The codes are as follows:

FUNCTION%=0 release all locked blocks

FUNCTION%=1 release the current implicit lock

FUNCTION%=2 convert the current implicit lock to an explicit lock

FUNCTION%=3 release the explicitly locked block specified in the BLOCK parameter. If BLOCK is 0, all explicitly locked blocks are released. However, implicitly locked blocks remain locked.

FUNCTION%=4 convert an implicit lock to an explicit lock and release the implicit lock

BLOCK specifies the starting block number for releasing an explicit lock. Note that BLOCK must be a floating-point number.

CHANNEL% is the I/O channel on which the operation is to be performed.

HANDLER INDEX% The handler index of the I/O device open on CHANNEL%. The index for disk devices is 0%.

If you open a file with a Recordsize greater than 512, SPEC% allows you to lock more than one block when you GET a range of blocks read into the buffer. For example, if you open the file with RECORDSIZE 1024%, each GET operation reads (and implicitly locks) two blocks. If you explicitly lock blocks 2 and 3 as follows:

```
GET #1%, RECORD 2%
VALUE%=SPEC%(2%,0%,1,0%)
```

you can then read blocks 3 and 4 (GET RECORD 3%) and cause implicit locks on these blocks. Note that if you attempt to lock a range of blocks that overlap an already explicitly locked range, the Monitor returns the ?DISK BLOCK IS INTERLOCKED error. Also, if a range of blocks is locked, an explicit release of those blocks must refer to the first block in the range.

The following errors are possible during a SPEC% operation:

Meaning	ERROR
?NO ROOM FOR USER ON DEVICE	4
There are too many locks pending on this device. You can lock a maximum of seven ranges of blocks on a file.	
?CAN'T FIND FILE OR ACCOUNT	5
You specified function code 3 in the FUNCTION% parameter and attempted to unlock a block that was not locked.	
?PROTECTION VIOLATION	10
You attempted to explicitly lock a block that had not been implicitly locked. An attempt to lock a block after a PUT or UNLOCK can cause this error.	
?DISK BLOCK IS INTERLOCKED	19
You attempted to explicitly lock a range of blocks that overlaps an already explicitly locked range of blocks.	

- Floppy Disk SPEC%
  - Returns density of a mounted floppy
  - Causes floppy driver to recompute density
  - Reformats a floppy to density specified (RX02 only)

For floppy disk devices, the SPEC% function allows you to obtain the density (single or double) of the current floppy disk, to mount a new floppy disk and recompute the density, and to reformat an RX02 floppy disk for a desired density. Because the RX02 floppy disk drive supports single and double density floppy disks, the SPEC% function is especially useful for programmed floppy disk operations. For example, SPEC% allows you to mount a series of single and double density floppy disks without having to close and reopen the device for each mount. That is, the driver computes density once; during the initial open. If you insert a second floppy that is incompatible with the initially computed density, read or write operations will fail. SPEC% permits you to include an instruction in your program that causes the driver to recompute the density. Also, for RX02 floppy disk drives, SPEC% permits you to specify a density reformat operation.

The SPEC% function for floppy disks has the following format:

VALUE%=SPEC%(FUNCTION%,PARAMETER%,CHANNEL%,HANDLER INDEX%)

where:

VALUE% depends on the particular function code you specify in FUNCTION%.

FUNCTION% is a function code that specifies the desired operation. The codes are as follows:

FUNCTION%=0% returns the density of the currently mounted floppy disk in the form; DENSITY%=VALUE% AND 255%. If DENSITY%=1%, the floppy is single density; if DENSITY%=2%, the floppy is double density.
Note that PARAMETER% must also be 0%.

FUNCTION%=1% causes the floppy disk driver to recompute density. If the diskette has been changed in the drive without closing and reopening the I/O channel, issue this code prior to any I/O operation on the floppy disk. This function also returns the computed density as described in FUNCTION%=0%. Note that PARAMETER% must be 0%.

FUNCTION%=2% reformats the current floppy disk to the density specified in PARAMETER%. PARAMETER% equals 1% for single density and 2% for double density. Note that this operation is allowed only on RX02 drives and that any data on the disk prior to the operation is lost.

PARAMETER% see the description of FUNCTION%.

CHANNEL% is the I/O channel on which the operation is to be performed.

HANDLER INDEX% the handler index on the I/O channel open on CHANNEL%. The index for floppy disks is 18%.

SPEC% can require as much as 20 seconds to reformat the density on an RX02 floppy disk and cannot be interrupted with CTRL/C. Note that if the operation is interrupted (by power failure or catastrophic error), the floppy disk is rendered unusable. That is, the floppy disk will contain both single and double density. To recover you must reformat the floppy disk.

The following errors are possible during a SPEC% operation:

ERROR	Meaning
14	?DEVICE HUNG OR WRITE LOCKED A hardware error occured. This can often be a transient condition. Retry the operation.
66	?MISSING SPECIAL FEATURE An attempt was made to reformat on an RX01 floppy disk drive. The use of SPEC% to reformat floppy disk density is allowed only on RX02 drives.

SPEC% can be used in floppy disk programming operations to ensure that sector opens are correctly handled. That is, the conflict between 128-byte single density buffersizes and 256-byte double density buffersizes can be resolved by the following method:

To FIELD the buffer;

```
FIELD #channel number, 128% *DENSITY% AS BUFFER RX02$
```

To PUT the buffer;

```
PUT #channel number, COUNT 128% *DENSITY%
```

When DENSITY% is;

```
DENSITY%=SPEC%(0%,0%,CHAN%,18%)and 255%
```

o Magnetic tape SPEC%

- Provides equivalent functions as the MAGTAPE function.

For magnetic tape, the SPEC% function allows you to rewind the tape, skip records on the tape, and set tape density and parity.

The SPEC% function for magnetic tape has the following format:

VALUE%=SPEC%(FUNCTION%,PARAMETER,CHANNEL%,HANDLER INDEX%)

where

VALUE% depends on the particular function code specified in FUNCTION%.

FUNCTION% is the function code.

PARAMETER depends on the particular function code specified in FUNCTION%.

CHANNEL% is the I/O channel on which the operation is to be performed.

HANDLER INDEX% is the handler index on the I/O device open on CHANNEL%. The index for magtape is 14%.

The SPEC% function performs various operations on magnetic tape as determined by the code you specify in FUNCTION%. These codes are as follows:

FUNCTION%=1 rewind the tape and take the unit off-line.

FUNCTION%=2 write an end of file marker.

FUNCTION%=3 rewind the tape but leave the unit on-line.

FUNCTION%=4 read the tape but skip specified records. You specify the number of records to be skipped in PARAMETER. Note that when FUNCTION% is 4, VALUE% returns the number of records that were read (i.e., not skipped).

FUNCTION%=5 backspace the tape over a specified number of records. You specify the number of records to be backspaced in PARAMETER. Note that when FUNCTION% is 5, VALUE% returns the number of records remaining (i.e., not backspaced).

FUNCTION%=6 set the tape's density and parity as specified in PARAMETER. The format for a PARAMETER specification, when FUNCTION% is 6, is E+D*4+P+S. The specifications E,D,P, and S have the following values:

E is phase encoded
256 = phase encoded mode, refer to value of D
0 = refer to values of D and P

D is density
if E=0

0 = 200 bpi (7-track only)
1 = 556 bpi (7-track only)
2 = 800 bpi (7-track only)
3 = 800 bpi (dump mode)

if E=256

0 = 1600 BPI
1 = Reserved for future use
2 = Reserved for future use
3 = Reserved for future use

P is parity
0 = odd parity
1 = even parity

S is stay
0 = the OPEN MODE value is released on CLOSE
8192 = the OPEN MODE value is retained on CLOSE

FUNCTION%=7 returns the tape status in VALUE%. The status returned is encoded as follows:

Bit	Decimal Value	Meaning
0-2	0	last issued command was off-line
	1	last issued command was read
	2	last issued command was write
	3	last issued command was write EOF
	4	last issued command was rewind
	5	last issued command was skip record
	6	last issued command was backspace record
3	0	See bits 13-14
	1	See bits 13-14
4	0	unit is TU10/TE10/TS03
	1	unit is capable of 1600 BPI
5	1	unit is off-line
6	1	the last issued command was a read on a record that was larger than the I/O buffer (part of the record is lost)
7	1	last command detected an EOF
8	1	tape is at its load point
9	1	tape is beyond the EOT marker
10	1	tape is write-locked
11	0	tape has odd parity
	1	tape has evin parity
12	0	9-track tape
	1	7-track tape
13-14		if bit 3 = 0 0 = 200 bpi 1 = 556 bpi 2 = 800 bpi 3 = 800 bpi, dump mode
		if bit 3 = 1 0 = 1600 bpi 1 = reserved 2 = reserved 3 = reserved
15	1	last command caused an error

FUNCTION%=10 returns the file characteristics in VALUE%. The characteristics returned in VALUE% are encoded as follows:

Word=0 DOS format tape
Word<>0 ANSI format tape. If the word is non-zero, the bytes have the following meaning:

Bit	Decimal Value	Meaning
0-11	-	For fixed-length ANSI format, the value is the record length in bytes. For variable-length ANSI format, the value is the maximum record length in bytes.
12-13	0	embedded carriage control
	4096	FORTRAN carriage control
	8192	implied line feed/carriage return before the record
14-15	16384	fixed-length format
	32768	variable-length format

FUNCTION%=11 rewind the tape on CLOSE

Note that the operation and values described for FUNCTION% are covered in detail in the RSTS/E Programming Manual.

- Terminal SPEC%

For terminals, the SPEC% function allows you to cancel CTRL/O and set modes for tape, echo, and ODT.

The SPEC% function for terminals has the following format:

VALUE%=SPEC%(FUNCTION%,PARAMETER,CHANNEL%,HANDLER INDEX%)

where

VALUE% depends on the particular function code specified in FUNCTION%.

FUNCTION% is the function code. The SPEC% functions performs various operations on terminals as determined by the FUNCTION% code. These codes are as follows:

FUNCTION%=0 Cancel CTRL/O
FUNCTION%=1 set tape mode
FUNCTION%=2 enable echo and clear tape mode
FUNCTION%=3 disable echo
FUNCTION%=4 set ODT mode
FUNCTION%=7 cancel all type ahead

PARAMETER specifies the terminal on which the operation is to take place. If PARAMETER is 0, the operation is performed on the currently open terminal. If you specify a keyboard number in PARAMETER, the operation is performed on that terminal. Note that the keyboard must be assigned to the calling job and must not be open.

CHANNEL% specifies the I/O channel for the terminal in PARAMETER.

HANDLER INDEX% the handler index of the I/O device open on CHANNEL%. The index for terminals is 2%.

System Function Calls

File Name String Scan SYS Call

- Suppresses logical device translation
- Example

```
ASSIGN MT:KB

Ready

LISTNH
1000    INPUT "FILE NAME"; F$ &
        \V$ = SYS(CHR$(6%)+CHR$(-10%)+"_"+F$) &
        \DEVICE$ = MID(V$,23%,2%) &
        \PRINT "DEVICE WITHOUT LOGICAL TRANSLATION IS "; DEVICE$
1010    V$ = SYS(CHR$(6%)+CHR$(-10%)+F$) &
        \DEVICE$ = MID(V$,23%,2%) &
        \PRINT "DEVICE WITH LOGICAL TRANSLATION IS ";DEVICE$
32767   END

Ready

RUNNH
FILE NAME? KB:
DEVICE WITHOUT LOGICAL TRANSLATION IS KB
DEVICE WITH LOGICAL TRANSLATION IS MT

Ready
```

Figure SOF-5. PNSC

Get Monitor Tables II SYS Call

- Returns information on whether or not a system is configured for large files
- Example

```
1000    ON ERROR GOTO 19000
2000    R.ETURN$ = SYS(CHR$(6%)+CHR$(-121%))     !MONITOR TABLES II &
        \IF SWAP%(CVT$%(RIGHT(R.ETURN$,29%)))                       &
              THEN PRINT "LARGE FILE SYSTEM"                        &
         ELSE PRINT "LARGE FILES NOT AVAILABLE"                     &
                !PRINT WHETHER THIS IS A LARGE FILE SYSTEM
2050    GOTO 32767
19000   PRINT ERR                                                   &
        \RESUME 32767
32767   END

Ready

RUNNH
LARGE FILE SYSTEM

Ready
```

Figure SOF-6. Get Monitor Tables

Change System Date/Time SYS Call

- Accepts zeroes for date/time
- Example

```
VS=SYS(CHR$(6%)+CHR$(-14%)+CHR$(0%)    &
   +CHR$(SWAP%(0%))+CHR$(0%)           &
   +CHR$(SWAP%(0%)))
```

RSTS/E V6C - ?Illegal mode mixing

RSTS/E V7.0 - No error returned
No change to date/time made

Directory Lookup Sys Call

- Always returns the correct file size (large file systems only)

ASSIGN/REASSIGN Sys Call

- Assigns/reassigns user logical names, user logical PPNs and user default protection codes
- Seizes a device

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(10%), the assign/reassign and enter user logical code.
3-4	Not used
5-6	For user logical assignment, the project-programmer number to be assigned.
7-10	For assign device, bytes 7 through 10 must be 0. For reassign device, byte 7 is the job number to which the device is reassigned. Bytes 8 through 10 must be 0. For user logical assignment, bytes 7 through 10 contain the logical device name (in Radix-50 format) to be assigned.
11-12+	For assign/reassign device, either DOS or ANS (in Radix-50 format) to specify DOS or ANSI label format for the magnetic tape drive. For user logical assignment, not used.

Byte(s)	Meaning
13-16	Not used
17-18	For assign/reassign device, CVT%$(SWAP%(-32767%)) to assign a device that is currently assigned to another user. This usage is privileged and the target device must not be open. If this operation is not desired, bytes 17 and 18 are 0. For user logical assignment, not used.
19-20	Not used
21	For user logical assignment,CHR$(255%) to enable protection code assignment (see BYTE 22). For assign/reassign device, must be 0.
22	For user logical assignment, the protection code to be assigned. Byte 21 must be 255. For assign/reassign device, must be 0.
23-24+	Device name for assign/reassign and for user logical assignment.
25+	Unit number for assign/reassign and for user logical assignment.
26+	Unit number flag for assign/reassign and for user logical assignment.
27-30	Not used

Data Returned: No meaningful data is returned.

Possible Errors:

Meaning	ERR Value
?ACCOUNT OR DEVICE IN USE	3
The specified device is currently open or has an open file. This error can only occur during an assign call.	
?NOT A VALID DEVICE	6
The device name specified in bytes 23 and 24 is a logical device name for which a physical device is currently not assigned.	
?DEVICE NOT AVAILABLE	8
The device specified in bytes 23 through 26 exists on the system but the operation fails for one of the following reasons:	
1. The device is currently reserved by another job (see bytes 17 and 18).	
2. Ownership of the device requires privilege that the user does not have.	
3. The device or its controller is disabled.	
4. The device is a keyboard line for a pseudo keyboard use only.	
?PROTECTION VIOLATION	10
A program with temporary privilege assigned a device, dropped its privilege, and attempted to reassign the device.	
?ILLEGAL NUMBER	52
An attempt is made to transfer control to a non-existent job. This error can only occur during a reassign call.	

Discussion:

The user logical assignment call is used to enter logical device names, logical project-programmer numbers, and default output protection codes. To assign a user logical device name, bytes 7 through 10 must contain the logical device name and bytes 23 through 26 must contain a physical device name and unit number. To assign a user logical project-programmer number, specify the number in bytes 5 and 6. To assign a user default protection code, specify the code in bytes 21 and 22.

The assign/reassign call used bytes 17 and 18 to assign or reassign a device which is currently assigned. For the call to be successful, it must originate from a privileged account, the target device must not be open, and the current owner can not be performing a directory on that device.

The assign call reserves a physical device to a job* or transfers assignment of a currently owned device to another job. The actions are equivalent to the ASSIGN and REASSIGN Monitor commands. System logical names are assigned either by a SYS call 21 or by UTILTY program commands.

Example:

```
10   A$ = SYS (CHR$(6%)+CHR$(10%)+STRING$(20%,0%)+
          "LP"+ CHR$ (1%)+CHR$(255%))
          !ASSIGN LP1: TO CURRENT JOB
20    INPUT "ASSIGN LP1: TO WHICH JOB"; X%
30    A$=SYS(CHR$(6%)+CHR$(10%)+STRING$(14%,0$)+
        CHR$(X%)+CHR$(0%)+STRING+(14%,0%)
         "LP"+CHR$(1%)+CHR$(255%))
          ! REASSIGN LP1: TO JOB # X%.
```

*The system manager, through an initialization option, can designate that certain devices require privilege to be assigned.

Deassign SYS Call

- Removes user logical assignments

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(11%), the deassign device and remove user logical code
3-4	Not used
5-6	For user logical removal, the project-programmer number to be removed. For device deassignment, must be 0.
7-10	For user logical removal, the logical device name (in Radix-50 format) to be removed. For device deassignment, must be 0.
11-20	Not used
21	For user logical removal, CHR$(255%) to enable protection code removal. For device deassignment, must be 0.
22	For user logical removal, the protection code to be removed. Byte 21 must be 255. For device deassignment, must be 0.
23-24	For device deassignment, the device name to be deassigned. For user logical removal, must be 0.

26	For device deassignment, the unit number flag. For user logical removal, must be 0.
27-30	Not used.

Data Returned: No meaningful data is returned

Possible Errors:

Meaning	ERR Value
?NOT A VALID DEVICE	6

The device or device type specified in bytes 23 through 26 is not configured on the system. This error can occur only on device deassignment calls.

Discussion:

The user logical removal call is used to deassign logical device names, logical project-programmer numbers, and default output protection codes. To deassign a logical device name, specify the name in bytes 7 through 10. To deassign a project-programmer number, specify the number in bytes 5 and 6. To deassign a user logical protection code, specify the code in bytes 21 and 22. Note that if these bytes do not contain specific deassignments, the call deassigns all user logical device name, PPN, and protection code assignments.

The device deassignment call performs the same action as the DEASSIGN system command described in the RSTS/E System User's Guide. For example, the following statement deassigns line printer unit 1, which is assigned to the current job.

```
10 A$ = SYS(CHR$(6%)+CHR$(11%)+STRING$(20%,0%)+
     "LP"+CHR$(1%)+CHR$(255%))
   !DEASSIGN LP1:
```

Create a Job SYS Call

- Spawns a job as long as there is swap space available

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS Call to FIP
2	CHR$(24%), the job creation code
3	CHR$(N%), where N% = 0% Spawn if LOGINs are enabled 128% Spawn regardless Note (N% AND 127%) must be 0
4	Not used, must be 0
5-6+	The project-programmer number of the program to be run.
7-10+	Program name of the program to be run in RADIX-50 format.
11-12+	Extension of the program to be run in RADIX-50 format.
13-22	Up to 10 bytes of information; placed into the created job's Core Common area.
23-26+	The device name and unit number of the program to be run.
27-28	The parameter word to be passed to the program to be run. The parameter word has exactly the same format and functions as the CCL command parameter word.
29-30	Not used

Data Returned:

Byte(s)	Meaning
3	The job number times 2 of the job just created.

Possible Errors:

Meaning	ERR Value
NO ROOM FOR USER ON DEVICE The new job cannot be created. Probable causes are: a) Further logins are disabled b) The system's job or swap slots are (currently) full	4

Discussion:

The job creating the new job must be running under control of the BASIC-PLUS Run-Time System. The job created can run under control of any available run-time system.

Exit and Clear Program SYS Call

- Allows a temporary switch to the specified run-time system.

Data Passed:

Byte(s)	Meaning
1	CHR$(9%), the exit and set up NONAME code.
2-3	First 3 characters of the name of the run-time system, in RADIX-50 format, to which control is passed. If bytes 2-5 are not specified, or if they are 0, your private default run-time system is used.
4-5	Last 3 characters of the name of the run-time system, in RADIX-50 format, to which control is passed.
6	If unspecified, establishes the named run-time system as the job's private default run-time system. Otherwise, CHR$(N%), where N% is 255% to establish the run-time system as the job's private default run-time system. 0% to enter the specified run-time system without establishing it as the job's default run-time system.

Wildcard PPN Lookup SYS Call

- New with V7.0

Wildcard PPN Lookup - Not Privileged (F=25)

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to PIP
2	CHR$(25%),the code for account number lookup on index
3-4	CHR$(N%)+CHR$(SWAP%(N%)), where N% is the index of the desired project-programmer number. If N% is 0, the call returns the project-programmer number of the first account on the disk that matches the wildcard specification. If N% is non-zero, the call returns the project-programmer number of the N+1 account on the specified disk that matches the wildcard specification.
5-6	The desired project-programmer number. A specification of 255 in either field represents a wildcard. A specification of 0 for N% in bytes 3-4 and a non-255 value in bytes 5-6 (no wildcard) verifies the existence of the specified account on the disk. If both fields are non-zero and neither field is 255, the call returns error 5 if the account does not exist.
7-22	Not used
23-24+	The device name, which must be a disk. If bytes 23 and 24 are both 0, SY0:(the system disk, not the entire public structure) is used.
25+	The device unit number

26+ The unit number flag. If byte 26 is 0, SY0: (the system disk) is used.

Data Returned:

Byte(s)	Meaning
3-4	Internal code
5-6+	The project-programmer number located by the call
7-30	Same as passed

Possible Errors:

Meaning	ERR Value
?CAN'T FIND FILE OR ACCOUNT	5
The specified device in bytes 23-24+ is not a disk or no match exists for the specified index value in bytes 3-4(see also bytes 5-6).	
?DISK PACK IS LOCKED OUT	22
The disk is in the locked state. Execute the call under a privileged account to override this condition.	

Discussion:

This call allows you to specify a wildcard account number and increment an index value to determine a matching project-programmer number. The wildcard account specification is as returned from the filename string scan call.

Return Job Status SYS Call

- New with V7.0

Return Job Status - Privileged and Not Privileged (F0=26)

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(26%), the return job status code
3	CHR$(J%), where J% is the number of the job for which status is desired. If J% is 0%, information on the caller's job is returned. If the caller is not privileged, J% is forced to 0%.
4	CHR$(S%), where S% is 0% or 1%. The value of S% determines the information returned on the job (see Data Returned).
5-30	Not used

Date Returned:

If S% is 0%:

Byte(s)	Meaning
1	The calling job's job number times two.
2	Not used
3	Job number times two of the job for which data is being returned. If the caller is not privileged, this byte is equal to byte 1.

4	Keyboard number of the job's console. If the number is negative, the job is detached and the number is the complement of the keyboard number.
5-6	If the job is attached to a pseudo keyboard, byte 5 contains the controlling job's job number times two; otherwise byte 5 is 0. If the job is swapped out, byte 6 contains the job's swap slot location; otherwise byte 6 is 0.
7-8	The job's logged in CPU time (least significant word) for the current session in tenths of a second.
9-10	The job's current connect time in minutes.
11-12	The job's current KCTs (least significant word) for this session.
13-14	The job's accumulated device time for the current session in minutes.
15	The most significant byte of the job's KCT.
16	The most significant byte of the job's CPU time.
17-20	The job's name in two Radix-50 words.
21-22	The job's project-programmer number.
23-26	The name of the job's default run-time system in two Radix-50 words.
27-30	The name of the job's current run-time system in two Radix-50 words.

If S% is 1:

Byte(s)	Meaning
1	The calling job's job number times two.
2	Not used
3	Job number times two of the job for which data is being returned. If the caller is not privileged, this byte is equal to byte 1.
4	Keyboard number of the job's console. If the number is negative, the job is detached and the number is the complement of the keyboard number.
5-6	The job's current flag word.
7	The job's current IOSTS byte.
8	The job's current information posting byte.
9-10	The job's current JBSTAT word.
11-12	The job's current JBWAIT word.
13	The size of the job's current user memory area in K words.
14	The job's current word from its memory control sub-block.
15-16	The job's current physical address in 32-word increments.
17	The job's priority.
18	The job's allotted run burst in tenths of a second.
19	The job's maximum allowable memory size in K words.

20 The value at offset 6 in the job's work block. This value is usually the channel number (times two) on which the job is performing an I/O operation.

21-22 If bytes 9 through 12 indicate that the job is in a keyboard wait state, bytes 21 and 22 contain the value at offset 12 (octal) in the job's work block. This value is the timeout parameter for input from the terminal. If the value is negative, it implies that the terminal is in a keyboard monitor (CTRL/C) input wait state.

If bytes 9 through 12 indicate that the job is in an I/O stall for a non-keyboard device, bytes 21 and 22 contain the generic name (in ASCII) of the device for which the job is stalled.

If bytes 9 through 12 indicate that the job is in a FIP wait state, byte 21 contains the byte value corresponding to the currently executing FIP function and byte 22 has no meaning.

23-24 The value at offset 16 (octal) in the job's work block. This value is usually an internal code that specifies whether the job is reading or writing on the current I/O channel. If the job is in an I/O wait state and this value is 2, the I/O operation is a read; if this value is 4, the I/O operation is a write.

25-26 A pointer to the beginning of the job's Job Data Block.

27-28 A pointer to the beginning of the job's second Job Data Block.

29-30 If the job is a receiver, these bytes contain a pointer to the job's receiver identification block; otherwise, these bytes are 0.

Possible Errors:

Meaning	ERR Value
?PROTECTION VIOLATION The calling job is not logged in.	10
?ILLEGAL SYS() USAGE The calling job number is less than zero or greater than JOBMAX.	10

Execute CCL SYS Call

- New with V7.0
- Allows execution of CCL commands while a program is running

Data Passed:

Byte(s)	Meaning
1	CHR$(14%), the execute CCL command Call
2-128	The string to be executed

Data Return: The target string is equivalent to the passed string.

Discussion:

This call scans the string that is passed to determine if it is a valid CCL command. If it is, the command is executed as though it had been typed directly to a keyboard monitor.

If the string passed is not a CCL command, execution of the program continues at the next statement.

Possible errors:

Meaning	ERR Value
?LINE TOO LONG The string that was passed is too long to be executed as a CCL command.	47
?ILLEGAL NUMBER An illegal number was specified as an argument to one of the CCL switches, such a specifying "/SIZE:A"	52

67

?ILLEGAL SWITCH USAGE

An illegal switch was specified for the CCL command, such as requesting a size that is larger than the System's SWAP MAX.

> **Note**
>
> Other errors will not be detected until the current program has been removed and execution of the invoked program is attempted. (See the discussion of .CCL in the RSTS/E System Directives Manual.)

Data Caching Sys Calls

- Enabling and Disabling Disk Caching

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(19%), the enable and disable cache code
3	CHR$(N%), where N% is: 0 to enable directory and data caching. In addition to this byte, data caching requires a value setting in byte 11. Note that bytes 4 through 12 are used only if this byte equals 0. 128 to return the current caching parameters. A 128 value in this byte does not enable or disable data caching. 1 to disable all caching
4	CHR$(C%), where C% is the cache clustersize. If C% is 0, the current clustersize is used (see Discussion). Cache clustersize can be specified as 1,2,4, or 8 blocks. If C% is greater than 8, 8 is used.
5-6	CHR$ (L%)+CHR$(SWAP%(L%)), where L% sets a limit on the total number of cache clusters that can be used. If L% is 0, the current limit is used (see Discussion). If L% is non-zero, it specifies an upper limit on the number of clusters in the cache. Note that if the amount of XBUF available to the cache is less than L%, the cache will not exceed XBUF.

7-8	CHR$(D%)+CHR$(SWAP%(D%)), where D% sets a limit on the total number of cache clusters allocated for directory caching. If D% is 0, the current limit is used (see Discussion). If D% is non-zero, it specifies an upper limit for the number of clusters in the cache that are available for directory caching. Note that the number of clusters allocated for directory caching during a particular operation can be less than D%.
9-10	CHR$(U%)+CHR$(SWAP%(U%)), where U% sets a limit on the total of cache clusters allocated to user data caching. If U% is non-zero, it specifies an upper limit for the number of clusters in the cache that are available for user data caching. Note that the number of clusters allocated for data caching during a particular operation can be less than U%.
11	CHR$(E%), where E% modifies the enabling/disabling of data caching as follows: E%=0 use the current setting E%=1 enable data caching as specified in file OPEN MODE or directory flag E%=128 disable all data caching E%=64 cache all data transfers regardless of file OPEN MODE
12	CHR$(M%), where M% controls the cache's use of the small buffer pool as follows: M%=0 use the current setting M%=1 use the small buffer pool M%=128 do not use small buffer pool

Data Returned:

Byte(s)	Meaning
1-2	Internal coding
3	Current cache setting and available options
	0 Cache disabled, user data caching is not available
	1 Cache enabled, user data caching is not available
	128 Cache disabled, user data caching is available
	129 Cache enabled, user data caching is available
4-12	Current settings of cache parameters, as described for passed data. Note that these bytes have meaning only if the system manager has installed user data caching during system generation.

Possible Errors:

Meaning	ERR Value
?ACCOUNT OR DEVICE IN USE	3
All of the clusters allotted to the cache are in use.	
?NO ROOM FOR USER ON DEVICE	4
An attempt was made to enable data caching without sufficient XBUF space allocated to the cache. The system manager must allocate at least 2K words of memory to XBUF for caching.	

?DEVICE NOT AVAILABLE	8
An attempt was made to change the cache clustersize (see Byte 4) while a cached file disk transfer was in progress. Retry the operation.	
?PROTECTION VIOLATION	10
The current user is not privileged.	
?MISSING SPECIAL FEATURE	
Caching was not configured for the system or data caching was configured and use of the small buffer pool was requested.	

Discussion:

Bytes 1,2, and 3 of this call enable or disable the FIP buffering module that controls directory caching. The ENABLE CACHE and DISABLE CACHE commands of UTILTY use these bytes.

If user data caching was configured on the system during system generation, bytes 1 through 12 enable or disable user data caching and set the parameters of the cache. The system manager defines the total size of XBUF during system initialization and some portion of this space is, in turn, used by the cache. The disk caching SYS call defines the size of the directory portion or the data portion of the cache. The sizes defined in this call set upper limits, not fixed sizes. Thus, if a 40K word XBUF were defined at system initialization, the SYS call could define the directory and data portions of the cache as 25K each. That is, data can use the space in the cache up to a maximum of 25K, which leaves a minimum of 15K for the directory. The reverse is also true. In this manner, data and directory caching are guaranteed a minimum allocation and are allowed to overlap, which permits the cache to dynamically adjust to system and program requirements.

This SYS call is also used to limit the size of the total cache. Because XBUF is used by the cache and DECnet/E, limiting the cache guarantees that space is always available in XBUF for DECnet/E. Note that the amount of memory allocated to the cache is freed for other use when caching is not being performed.

Byte 4 of the call sets the cache clustersize. This parameter controls the number of contiguous blocks that are copied from the disk to the cache whenever file data or directory data are cached. The cache clustersize should be small enough to contain a reasonable number of clusters, but large enough to reduce the number of disk accesses. That is, you must anticipate data requests and ensure that the cache is equal to the file clustersize of the most often accessed file. If you specify a cache clustersize of 1, only random caching is allowed. Refer to the RSTS/E System Manager's Guide for cache clustersize guidelines.

Note that the parameters for cache clustersize and cluster allocation (Bytes 4 through 10) have default settings at system startup. The default settings are a cache clustersize of 4 and no limits on directory, data, or total cache size. The system manager can reset these defaults with an INIT option as described in the RSTS/E System Manager's Guide.

File Utility SYS Call

- New with V7.0
- Sets a file's directory data cache characteristics
- Makes a contiguous file non-contiguous
- Changes a file's placed bit setting
- Obtains file's starting DCN
- Changes a file's run-time system name
- Changes a file's backup characteristics
- Returns an unopened file's run-time system name and attributes

File Utility Functions - Privileged

Date Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP.
2	CHR$(-26%), the file utility code.
3	CHR$(N%), where N% is the internal channel number (in the range of 1 to 12) on which the file is open. If N% is 0, specify the target file by project-programmer number, filename, and extension in bytes 5 through 12.
4	CHR$(F%), where F% specifies the file utility function. The function F% is one (or the sum) of the following codes:

1% set or reset the file's placed bit (can not be used with code 8%). See byte 15.

2% modify code 16% to return 0 as the device cluster number (DCN) if the file's placed bit is not set.

4% change the file's backup statistics.

8% change the file's run-time system name field.

16% return the file's retrieval information. That is, 16% causes the Monitor to map the virtual block number (VBN) of the file into the disk DCN. This code can be used to obtain an existing file's DCN in order to place a new file near it (see Discussion)

32% unset the file's contiguous bit. This code allows you to extend a contiguous file, however, the file is made non-contiguous.

64% enable/disable sequential mode caching if the file is cached (can not be used with code 8%). See byte 15.

128% enable/disable data caching on the file (can not be used with code 8%). See byte 15.

5-6+ If N% in byte 3 is 0, specify the project-programmer number of the file you wish to modify.

If N% is non-zero, these bytes are ignored.

7-10+ If N% in byte 3 is 0, specify the filename (in Radix-50 format) of the file you wish to modify.

If N% is non-zero, these bytes are ignored.

11-12+ If N% in byte 3 is 0, specify the file extension (in Radix-50 format) of the file you wish to modify.

If N% in non-zero, these bytes are ignored.

13-16 The specifications that you make in these bytes are dependent on the function code specified in byte 4, as follows:

If byte 4 AND 8%<>0%, then bytes 13 through 16 contain the new run-time system name field in Radix-50 format.

Otherwise;

If byte 4 AND 16%<>0%, then bytes 13 and 14 contain the low order word of the VBN you wish to locate, byte 15 contains 0% or is used by another operation, and byte 16 contains the high order word of the VBN you wish to locate.

If byte 4 AND 1%+64%+128%, then bytes 13, 14 and 16 contain zeros or are used by another operation, byte 15 contains flags for the following operations:

2% new value for the placed bit if byte 4 AND 1%<>0%.

4% new value for sequential bit if byte 4 AND 64%<>0%.

128% new value for cached bit if byte 4.

17-18 If the change file backup statistics function is selected in byte 4 (code 4), these bytes specify a new date of last access for the file. If no change of date is desired, specify 0.

If the statistics function is not selected, these bytes are ignored.

19-20 If the change file backup statistics function is selected in byte 4 (code 4), these bytes specify a new date of creation for the file. If no change of date is desired, specify 0. If the statistics function is not selected, these bytes are ignored.

21-22 If the change file backup statistics function is selected in byte 4 (code 4), these bytes specify a new time of creation for the file. If no change of time is desired, specify 0. If the statistics function is not selected, these bytes are ignored.

23-24+ If N% in byte 3 is 0, specify the name of the device that contains the file you wish to modify. The device must be a disk and a specification of 0 in bytes 23 and 24 indicates the public disk structure.

If N% is non-zero, these bytes are ignored.

25-26+ If N% in byte 3 is 0, specify the unit number and unit number flag associated with the file you wish to modify.

If N% is non-zero, these bytes are ignored.

27-30 Not used.

The file's characteristics are returned as follows:

Byte(s)	Meaning
1	Not used
2 byte 1 = 2	file is placed
byte 1 = 4	file will be cached sequentially if at all
byte 1 = 16	file is contiguous
byte 1 = 128	file will be cached on open

3-4 If the file's VBN was passed in byte 16 and file retrieval information (code 16) was requested in byte 4 (see Data Passed), these bytes contain the DCN of the file's VBN. Note that these bytes return 0 if the specified VBN is larger than the file size or if the file was not placed and function code 2 was not passed in byte 4.

5-26 File attribute data; unused words are filled with zeroes.

27-30 The file's run-time system name in Radix-50 format.

Possible Errors:

Meaning	ERR Value
?CAN'T FIND FILE OR ACCOUNT The file or account specified in bytes 5 through 12 is not present on the disk.	5

?I/O CHANNEL NOT OPEN 9

The channel specified in byte 3 is not open.

?PROTECTION VIOLATION 10

The file open on the channel specified in byte 3 is not a disk file or the job lacks the privilege required for the specified operation.

?ILLEGAL SYS() USAGE 18

The file open on the specified channel is not a disk file or is a user file directory.

Discussion:

This call supplements the functions of the change file run-time system name call (FIP SYS Call -17) and the change file statistics call (FIP SYS Call -11) to support large files and to add support for file placement. You can also use this call to obtain a file's run-time system name and attribute data without opening the file.

The run-time system name field (see Data Passed, bytes 4 and 27 through 30) in the accounting entry of the file's User File Directory contains file size information for large files. The two-word run-time system name field is decoded as follows:

1. If the first word is non-zero, the data in both words is the run-time system name. The file size is limited to 65535 blocks.

2. If the first word is 0, the low order byte of the second word contains the most significant bits of the file size. The file size is limited to 2**23-1 blocks. The high order byte of the second word is reserved and must be 0.

The following restrictions apply to large files:

1. Because a runnable file can not have both a run-time system name and a most significant byte indication in the same field, large files are not runnable.

2. You can not extend a compiled file beyond block 65535. An attempt to extend a compiled file past block 65535 results in a ?PROTECTION VIOLATION error (ERR=10).

3. You can not rename a file that is larger than 65535 blocks with the intent of assigning a compiled protection code. The attempt is rejected with no error and the compiled bit remains off.

4. When you extend a file past block 65535, it loses its run-time system name.

5. You can not change the run-time system name of a file that is larger than 65535 blocks. The attempt results in a ?PROTECTION VIOLATION error.

6. You can not change the run-time system name of a compiled file to two words of zeroes. The attempt results in a ?PROTECTION VIOLATION error. Note that you can perform this operation on a non-compiled file.

7. You can not change the run-time system name of an file to a zero word followed by a non-zero word.

To place a file in a particular position of the disk, you specify the desired disk DCN (Device Cluster Number) as returned in bytes 3 and 4 of this call in the file specification /POSITION switch (see the RSTS/E System User's Guide). The Monitor attempts to place the first block of the file at or after the specified DCN. If the file placement is successful, the placed bit (bit 1, mask value 2) in the file's UFD entry is set (see SYS calls -10 and -23). If the file placement is not successful, the first block of the file is placed at the lowest free block on the disk, the UFD placed bit is not set, and no error is returned.

This page is for notes.

Resident Library SYS Calls

Add a Resident Library - Privileged (F0=-18)

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(-18%), the resident library manipulation code
3	CHR$(16%), add a resident library
4	Not used
5-6+	The project-programmer number of the file to add
7-10+	The resident library name in Radix-50 format
11-12	CHR$(A%)+CHR$(SWAP%(A%)), where A% is the 1K-word section of memory at which the resident library is to be loaded. The numbering begins at the first available 1K-word section and ends at n-1 (where n is the total number of 1K-word sections of memory on the system). A% must be specified and cannot be zero.
13-17	Not used
18	CHR$(S%), where S% is the stay flag. If S% is 128% (the high bit set), the resident library is made permanently resident. If S% is 0%, the memory occupied by the library can be freed for user space whenever the usage count of the library is zero (no active task is accessing the library).

19-20 CHR$(F%)+CHR$(SWAP%(F%)), where F% is the flag word that defines the characteristics of the library. Only the high byte is used for flag bits. F% is the sum of the bits set, as follows:

F%=256% Not used

F%=512% The resident library is available to only one user. It is not shared by multiple users.

F%=1024% The resident library allows read/write access to its memory, rather than read only access.

F%=2048% The resident library does not record errors in its code in the system error log.

F%=4096% The resident library is immediately removed from memory when its usage count equals zero.

F%=8192% Not used

F%=16384% Not used

F%=32767%+1% Not used

21-22+ Protection code for the installed resident library. To specify a protection code, byte 21 is non-zero and byte 22 contains the desired protection code. To accept the default protection code, byte 21 is 0. The default protection code is <42>, which means that the Monitor grants read access to all users but denies write access.

23-24+ The name of the disk device on which the resident library is to be stored. If no name is specified, SY: is used.

25+ Unit number

26+ Unit number flag

27-30 Not used

Data Returned: No meaningful data is returned.

Possible Errors:

Meaning	ERR Value
?NO ROOM FOR USER ON DEVICE You specified an address in bytes 11 and 12 that would cause the Monitor to load the library in such a manner that memory would be fragmented and a swapping violation would occur. Refer to the RSTS/E System Generation Manual for guidelines on avoiding memory fragmentation.	4
?CAN'T FIND FILE OR ACCOUNT You specified a filename in bytes 7 through 10 that cannot be found in the account specified in bytes 5 and 6 on the device specified in bytes 23 through 26. Ensure that the filename you specify has a .LIB extension and is located in the specified account and device.	5
?PROTECTION VIOLATION The file you wish to add is in improper format. For example, this error occurs if you specify a file that is not contiguous or has illegal entries in the SIL index.	10

?NAME OR ACCOUNT NOW EXISTS 16

You specified the filename of a resident library that already exists.

?ILLEGAL BYTE COUNT OF I/O 31

You did not specify a load address in bytes 11 and 12 or the address you specified is not available. Refer to the memory status report of a display program to determine an available range of memory.

?NO BUFFER SPACE AVAILABLE 32

A small buffer is required for the description block of an added resident library. This error is returned if a small buffer is not available.

Discussion:

The Monitor SYS call to add a resident library is privileged and allows you to add a specified library to the Monitor's list of resident libraries. This call is similar to that used to add a run-time system except that you must specify a load address for the library. That is, unlike run-time systems, the Monitor does not automatically decide where to load a resident library. Also, the library file does not have to reside in account [0,1]; however, the file extension must be .LIB.

For additional information on the creation and use of resident libraries, refer to the RSTS/E Task Builder Reference Manual.

Remove a Resident Library - Privileged (F0=-18)

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(-18%), the resident library manipulation code
3	CHR$(20%), remove a resident library
4-6	Not used
7-10+	The resident library name in Radix-50 format
11-30	Not used

Data Returned: No meaningful data is returned

Possible Errors:

Meaning	ERR Value
?ACCOUNT OR DEVICE IN USE	3
You attempted to remove a library that is being loaded into memory or is in use by the currently running job. A resident library cannot be removed while a job is still attached to it.	
?CAN'T FIND FILE OR ACCOUNT	5
You specified a resident library name in bytes 7 through 10 that is not currently defined.	

Discussion:

The Monitor SYS call to remove a resident library is privileged and allows you to remove the library from physical memory, delete the Monitor structure that defines the library, and close the library file.

Load a Resident Library - Privileged (F0--18)

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(-18%), the resident library manipulation code
3	CHR$(18%), load a resident library
4-6	Not used
7-10+	The resident library name in Radix-50 format
11-12	CHR$(A%)+CHR$(SWAP%(A%)), where A% is the 1K-word section of memory at which the library is to be loaded. The numbering begins at the first available 1K-word section and ends at n-1 (where n is the total number of 1K-word sections of memory on the system). If A% is 0%, the Monitor uses the load address that was initially set for the library when it was added. If A% is non-zero, the value specified replaces that initially set for the library.
13-17	Not used
18	CHR$(S%), where S% is the stay flag. If S% is 128% (the high bit set), the library is made permanently resident (i.e., the usage count is always non-zero). If S% is 0%, the memory occupied by the library is freed for user space whenever the usage count is zero.
19-30	Not used

Data returned: No meaningful data is returned

Possible Errors:

Meaning	ERR Value
?NO ROOM FOR USER ON DEVICE	4
You specified an address in bytes 11 and 12 that would cause the Monitor to load the library in a position where memory would be fragmented and a swapping violation would occur. Refer to the RSTS/E System Generation Manual for guidelines on assigning and allocating memory to avoid fragmentaton.	
?CAN'T FIND FILE OR ACCOUNT	5
You specified a filename in bytes 7 through 10 for a resident library that is not currently defined.	
?ILLEGAL BYTE COUNT FOR I/O	31
The load address was not specified in bytes 11 and 12 or the address specified is not available. Refer to the memory status report of a display program to determine an available range of memory.	

Discussion:

The Monitor SYS call to load a resident library is privileged and allows you to specify a position in memory for the library. That is, you can use this call to move a resident library from its current position in memory to a position of your choice. Note that if you wish to move a library to a permanent position in memory, you must set the stay bit (see format description). The stay bit ensures that the library will remain in the specified location for the duration of the current time-sharing session.

Unload a Resident Library - Privileged (F0=-18)

Data Passed:

Byte(s)	Meaning
1	CHR$(6%), the SYS call to FIP
2	CHR$(-18%), the resident library manipulation code
3	CHR$(22%), to unload a resident library
4-6	Not used
7-10+	The resident library name in Radix-50 format
11-30	Not uesd

Data Returned: No meaningful data is returned.

Possible Errors:

Meaning	ERR Value
?ACCOUNT OR DEVICE IN USE	3
You attempted to unload a resident library that is in the process of being loaded or is in use by the currently running job. A library cannot be unloaded while a job is still attached to it.	
?CAN'T FIND FILE OR ACCOUNT	5
You specified a filename in bytes 7 through 10 for a library that is not currently defined.	

Discussion:

The Monitor SYS call to unload a resident library is privileged and allows you to remove the library from memory and free that portion of memory for use by other jobs.

Implementing New RSTS/E Features under BASIC-PLUS

Creating and Accessing Large Files

- Pre-extending a large file

```
1000 OPEN "file.ext/S1:70000 "FOR OUTPUT.....
```

> Note
>
> The switch on the file specification must be used since the FILESIZE option of the OPEN statement accepts only integers.

- Reading from and writing to a large file

```
GET #N%, BLOCK X

PUT #N%, BLOCK X
```

> Note
>
> The BLOCK modifier accepts floating point numbers. RECORD modifier accepts integers.

Specifying Data Caching

> **Note**
>
> Data caching must be configured into the monitor.

Data Caching

When a user job executes a read request, the monitor performs a disk access and transfers the requested data from the disk to the user job's buffer. On systems that have many jobs requiring large amounts of data, the large number of disk data transfers can have an impact on response time.

Data caching is a method whereby a privileged user can reduce data transfers from disk. When caching is enabled, the monitor stores the most recently read (accessed) data blocks in an area of memory called the cache. If the user job then requests a data block that is present in the cache, the monitor copies the requested data directly from the cache into the job's buffer. thus, the monitor avoids a physical disk access operation.

Data caching is most useful for read operations in that it can minimize disk transfers. In a write operation that modifies existing data, the data is updated in the cache and on disk but no new data is installed in the cache.

The system manager configures caching on the system and optionally sets its parameters during system initialization. A privileged user enables or disables caching and determines the size of the cache by means of the disk cache enable SYS call or the UTILTY system program. The SYS call or UTILTY are also used to specify caching for a particular file by marking the file's UFD entry. If a new file is to be created and accesses to it are to be cached, use the MODE values in the OPEN statement. However, because the use of caching MODE values is privileged, a privileged user should designate a file for caching by means of the file's UFD entry (i.e., use the file utility SYS call or UTILTY).

Once a file's UFD is marked for caching, it will be cached on OPEN regardless of the user's privilege as long as caching is enabled on the system.

When caching is enabled the cache receives all data transfer requests that are directed to the disk driver. Because the cache contains data, read operations on data that is in the cache occur without placing a load on the disk driver. The monitor constantly updates the cache such that it contains the most recently requested data.

During system initialization (or with the disk cache enable sys call), the system manager sets the cache cluster size. The cluster size (1, 2, 4, or 8 blocks) determines the smallest amount of data transferred into the cache on a disk access. For example, when the cache cluster size is set at 8 blocks, a read operation that installs data in the cache causes the monitor to determine the number of cache clusters needed to fulfill the request. If a single cache cluster is required, 8 physically contiguous blocks (including the requested blocks) are installed. If more than one new cache cluster is required to full fill the request, a separate disk access is performed to install the data into each of the required cache clusters. However, if the number of new cache clusters required is more than two, the data is not installed in the cache but is transferred directly into the user buffer with a single disk access. Note that the cache cluster size should not usually be greater than the pack cluster size that was set during disk initialization. The Monitor allocates space for the cache from the Extended Buffer Pool (XBUF).

User Data Caching (Random Mode) - MODE 256%

A privileged user can specify MODE 256% in an OPEN statement to cause data transfers from a file to be cached in random mode. MODE 256% has effect only if data caching is enabled on the system.

The monitor schedules data transfers into the cache on the basis of time since last access. That is, new data replaces data currently in the cache if the currently cached data:

1. is the data with the longest time since last access, and

2. has been in the cache for more than the minimum residency as established by the system manager.

If a data block meets these requirements, it is replaced when the monitor seeks to place a new data block in the cache.

As an example of random cache operation, consider a get statement that is executed on a file opened with mode 256%. When a read on the cached file occurs, the monitor examines the contents of the cache to determine if the requested data item is present. If the data is in the cache, the data is copied from the cache cluster that contains it. The data is made available to the program and the cache cluster is linked to the beginning of the list of cache clusters. The list of cache clusters is maintained in order of increasing time since last access.

If the requested data item is not present in the cache, the monitor examines the list of cache clusters to determine the time of last access for the oldest item in the cache. If the time is less than the minimum residency, the requested data cannot be installed in the cache and a normal disk read is performed. If the time is greater than the minimum residency, the requested data is installed in that cache cluster and made available to the program.

User Data Caching (Sequential Mode) - MODE 2048%

MODE 2048% has effect only if the file is being cached. That is, MODE 256% is set, the file's UFD entry is marked for caching, or caching is set for all data on the system. Note that sequential mode data caching requires a cluster size greater than 1, although no error is returned if the cluster size is 1.

When MODE 2048% is specified in an OPEN statement, data blocks from that file, placed in the cache as a result of that OPEN, are not subject to the minimum residency. However, MODE 2048% overrides the minimum residency only for that file's data blocks, not for the entire cache. That is, when data blocks from a sequentially cached file are placed in the cache, a read on the last block of the cluster causes it to become the "oldest" data block in the cache. It is the first to be replaced when the monitor seeks to place new data blocks in the cache. MODE 2048% has the effect of speeding the replacement process in the cache and thus minimizes the space required by the cache.

As an example of sequential cache operation, consider a GET statement that is executed on a file opened with MODE 2304%(256% + 2048%). When a read on the cached file occurs, the monitor examines the contents of the cache to determine if the requested data item is present. If the data is in the cache, the data is copied from the cache cluster that contains it. The data is made available to the program. If the data is in the last block of a cache cluster, that cluster is made available for replacement in the cache.

When a read operation is performed on the first block of a cache cluster, the monitor installs a full cluster of data in the cache. That is, if the system manager sets a clustersize of 8 for the cache, the requested data block plus the next 7 contiguous blocks are read from the disk into the cache. When the last block of the cache buffer is read, the cluster is immediately made available for the installation of new data as requested by other read operations.

If the GET statement causes more than one cache cluster to be read, all of the requested data blocks are made available to the program but only the last cache cluster is installed. However, if the last data block read is the last block in a cache cluster, no data is installed in the cache. Thus, if the cache cluster size is defined as 1 and sequential mode is specified, no data blocks will be installed in the cache (i.e., every data block is the last block in a cache cluster).

Table SOF-1. (Page 1 of 2)

Manager Specification through UTILTY	Non-Privileged Programming Options	
	Random Cache	Sequential Cache
DISABLE CACHE	No data caching * available	
ENABLE CACHE/ALL	Open file with no data caching modes	Open file with MODE 2048
ENABLE CACHE/FILE	No data caching * available	
FLAG file/CACHE**	Open file with no data caching modes	Open file with MODE 2048
FLAG file/ NOCACHE**	No data caching * available	
FLAG file/ CACHE/SEQ**	No random cache available	Open file with No data cache mode
FLAG file/ CACHE/RAN	Open file with no data caching modes	Open file with mode 2048

* If data caching modes are specified, no error is issued.

** These commands set and unset data caching bits in the disk directory structure for the file.

Table SOF-1. (Page 2 of 2)

Manager Specification through UTILTY	Privileged Programming Options	
	Random Cache	Sequential Cache
DISABLE * CACHE	1. Enable cache SYS call ** 2. Open file with MODE 256 ***	1. Enable cache SYS call ** 2. Open file with MODE 256 + 2048 ****
ENABLE CACHE/ALL	Open file with no data caching mode	Open file with MODE 2048
ENABLE CACHE/FILE	Open file with MODE 256 ***	Open file with MODE 256 + 2048 ****
FLAG file/ CACHE	Open file with no data caching mode	Open file with MODE 2048
Flag file/ NOCACHE	Open file with MODE 256	Open file with MODE 256+2048
FLAG file/ CACHE/SEQ	No random caching available *****	Open file with no data caching mode.
FLAG file/ CACHE/RAN	Open file with no data caching mode	Open file with MODE 2048

* USE LIST CACHE command to check if disabled.

** Enables data caching for the system.

*** The FLAG file/CACHE UTILTY command or the file utility SYS call can be used, but this makes random and sequential caching of the file available to all users.

**** The FLAG file/CACHE/SEQ UTILTY command or the file utility SYS call can be used, but the file will be cached sequentially for anyone using the file.

***** The FLAG file/RAN UTILTY command or the file utility SYS call can be used.

Extending Files Opened in Update Mode

- To extend the file:
  a. Open the file in update mode (MODE 1 or MODE 5)
  b. Read in the first block of the file.
  c. Explicitly lock the first block using the SPEC% function.
  d. Extend the file by writing beyond the end of file.
  e. Release the explicit lock on block 1 using the SPEC% function, the UNLOCK statement or the CLOSE statement.

> **Note**
>
> As extended blocks are written, they are made available to other users.
>
> Programs attempting to read block 1 during this process receive ?Disk Block is Interlocked errors.

- Example

Create the file

```
1000    OPEN "UPDATE.DAT<48>" FOR OUTPUT AS FILE 1% &
        \FIELD #1%, 20% AS N.AME$
1010    FOR I% = 1% TO 2% &
        \INPUT "RECORD"; N$ &
        \LSET N.AME$ = N$ &
        \PUT #1% &
        \NEXT I%
32767   CLOSE 1% &
        \END

Ready
```

Figure SOF-7. Create the File

Look at Directory

```
DIR UPDATE.DAT/$
 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
ST:[100,7]
UPDATE.DAT       2   < 48> 28-Nov-78 28-Nov-78 12:32 PM   8 BAS4P  23121  8/8

Total of 2 blocks in 1 file in ST:[100,7]

Ready
```

Figure SOF-8. Look at Directory

Extend the file

```
1000    ON ERROR GOTO 19000
1010    OPEN "UPDATE.DAT" AS FILE 1%, MODE 5%
1015    FIELD #1%, 20% AS N.AME$
1020    GET #1%, BLOCK 1 &
        \I% = SPEC%(4%,0%,1%,0%)        !EXPLICIT LOCK ON BLOCK 1&

1030    LSET N.AME$ = "NEW RECORD" &
        \PUT #1%, BLOCK 17%
1040    I% = SPEC%(3%,1%,1%,0%)         !RELEASE BLOCK 1 &
        \GOTO 32767
19000   PRINT ERR &
        \RESUME 32767
32767   CLOSE 1% &
        \END

Ready
```

Figure SOF-9. Extend the File

Look at Directory

```
DIR UPDATE.DAT/$
 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
ST:[100,7]
UPDATE.DAT      17   < 48> 28-Nov-78 28-Nov-78 12:32 PM   8 BAS4P  23121  1/0U

Total of 17 blocks in 1 file in ST:[100,7]

Ready
```

Figure SOF-10. Check Directory

Placed Files

- Specify the device cluster number on the /POSITION switch of the file specification.

```
LISTNH
1000    EXTEND                                  ! EXTEND MODE       &
        \ ON ERROR GOTO 19000                   ! ERROR TRAPPER     &

1010    INPUT "FILENAME.EXT";FIL$               ! GET NAME.EXT      &
        \ INPUT "FILESIZE(BLOCKS)";SIZ%         ! GET SIZE          &
        \ INPUT "DISK POSITION";DIP$            ! GET POSITION      &
        \ SPE$=FIL$+"<48>"+"/POSITION:"+DIP$    ! SET UP SPEC       &
        \ OPEN SPE$ FOR OUTPUT AS FILE 1%,      ! OPEN SPEC         &
        FILESIZE SIZ%,MODE 16%                  ! MODE AND FILESIZE&
        \ CLOSE 1%                              ! CLOSE IT          &

1020    PRINT "FILE HAS BEEN CREATED!"          ! DONE              &
        \ GOTO 32767                            ! EXIT              &

1030    PRINT "NO ROOM ON DISK FOR FILESIZE "+  ! NOT ENOUGH ROOM &
              NUM1$(SIZ%)+" TRY AGAIN"          ! TELL USER        &
        \ GOTO 1010                             ! GO TRY AGAIN     &

19000   IF ERR=4% THEN RESUME 1030%             ! NO DISK SPACE    &

19010   ON ERROR GOTO 0                         ! DISABLE TRAP     &

32767   END

Ready
```

Figure SOF-11. Placed File

Using Tentative Files

- Specify MODE 32 in an OPEN FOR OUTPUT statement.

```
OPEN "file.ext" FOR OUTPUT AS FILE 1%, &
     MODE 32%
```

- System Actions

  If the file specified already exists, the tentative file:

  - is "marked for deletion"
  - shows only in a slow listing of DIRECT
  - Example

```
DIR UPDATE.*
 Name .Ext    Size   Prot    Date        ST:[100,7]
UPDATE.DA1      17   < 40> 28-Nov-78

Total of 17 blocks in 1 file in ST:[100,7]

Ready

DIR UPDATE.*/S
 Name .Ext    Size   Prot    Access      Date      Time    Clu  RTS    Pos  Op/rr
ST:[100,7]
UPDATE.DA1      17   < 40> 28-Nov-78 28-Nov-78 12:32 PM    0 BAS4P  23131  0/0
UPDATE.DAT*      0   < 40> 28-Nov-78 28-Nov-78 02:54 PM    0 BAS4P  -----  1/0W

Total of 17 blocks in 2 files in ST:[100,7]

Ready
```

Figure SOF-12. Tentative Files

When the tentative file is explicitly CLOSEd, the original file is deleted and the tentative file is renamed.

> **Note**
>
> If the tentative file is not explicitly CLOSEd, is CLOSEd with a negative channel number or if the system should crash, the original file remains intact and the tentative file is deleted.

- Example

```
1000    OPEN "BOOK.LST" FOR OUTPUT AS FILE 1%, MODE 32% &
        \OPEN "BOOK.LST" AS FILE 2%, RECORDSIZE 32767%+1%+2%
1010    FIELD #1%, 30% AS BOOK.TITLE$ &
        \GET #2% + SWAP%(1%) &
        \LSET BOOK.TITLE$ = "PDP-11 COBOL REFERENCE MANUAL" &
        \PUT #1%
1020    CLOSE 1%
32767   END
```

Figure SOF-12A. Tenative File Program

Making Contiguous Files Non-Contiguous

```
OLD NONCNT

Ready

LISTNH
100     EXTEND
500     ON ERROR GOTO 19000 &
        \E% = -1%
1000    OPEN "CONT.DAT" AS FILE 1% &
        \FIELD #1%, 20% AS N.AME$
1010    INPUT "NAME"; N$ &
        \LSET N.AME$ = N$
1020    PUT #1%,  BLOCK 20 &
        \GOTO 32766
2000    V$ = SYS(CHR$(6%)+CHR$(-26%)+CHR$(1%)+CHR$(32%)) &
        \GOTO 1020
19000   IF ERL = 1020 &
             THEN PRINT "ERROR NO."; ERR; "WITH E% = "; E% &
                  \E% = 0% &
                  \RESUME 2000
19010   CLOSE 1% &
        \ON ERROR GOTO 0
32766   CLOSE 1%
32767   END

Ready

DIR CONT.DAT/S
 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
ST:[1,235]
CONT  .DAT      16C  < 50> 22-Mar-79 22-Mar-79 11:31 AM   8 BAS*F  13702  0/0
Total of 16 blocks in 1 file in ST:[1,235]

Ready

RUN NONCNT
NAME? HARRY JAMES
ERROR NO. 10 WITH E% = -1

Ready

DIR CONT.DAT/S
 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
ST:[1,235]
CONT  .DAT      20   < 60> 22-Mar-79 22-Mar-79 11:31 AM   8 BAS*F  13702  0/0
Total of 20 blocks in 1 file in ST:[1,235]

Ready
```

Figure SOF-13. Contiguous Files

Using Non-File Structured Block Mode

- Available on large file systems only
- Specify MODE 128 on OPEN
- Each GET/PUT using the BLOCK modifier accesses a logical block rather than a device cluster.

> **Note**
>
> Block number 1 is the first block of the MFD on RSTS/E disks.

```
1000  EXTEND                                    ! EXTEND MODE      &
      \ ON ERROR GOTO 19000                     ! ERROR TRAPPER    &

1025  FOR I%=1% TO 9%                           ! CAR IDENT'S      &
      \ CARS(I%)=CARS(I%)+NUM1$(I%) FOR I%=1% TO 3%  ! THREE EACH  &
      \ NEXT I%                                 ! 9 CAR IDENT'S    &
      \ CARS(10%)="000"                         ! AND THE TENTH    &
      \ FORMLINES="### \"+SPACE$(30%)+"\"+      ! FORMATS          &
            "  ####  $$##.##  ##  $$#,###.##"   ! REST OF IT       &
      \ HEDS="CAR  MAKE & MODEL & STYLE"+       ! HEADER           &
      SPACE$(12%)+"  PART   PRICE   QH  EXTENDED VALUE"! MORE OF SAME &
      \VALFORS="TOTAL VALUE"+SPACE$(43%)+"$$#,###,###.##"! VALUE FORMAT &

1010  INPUT "READ FROM WHICH DISK(DDN:)";DSK$   ! GET INPUT DISK   &
      \ OPEN DSK$ AS FILE 1%,MODE 128%          ! BLOCK MODE NFS   &

1020  FIELD #1%,2% AS CAR.ID$,                  ! KEY FIELD        &
            2% AS PART.NO$,                     ! PART CODE        &
            30% AS DESC$,                       ! DESCRIPTION      &
            8% AS UNIT.PRI$,                    ! PRICE EACH       &
            2% AS QUAN.AVA$,                    ! QUANTITY ON HAND &
            464% AS REST$,                      ! REST OF RECORD   &
            2% AS LST.BLK$,                     ! LAST BLK THIS ID &
            2% AS NXT.BLK$                      ! NEXT BLK THIS ID &

1030  INPUT "CAR ID CODE";CARID$                ! GET WHAT TO READ &
      \ FOR X%=1% TO 10%                        ! START VALIDATION &
      \ GOTO 1040 IF CARS(X%)=CARID$            ! FOUND IT         &
      \ NEXT X%                                 ! TRY AGAIN        &
      \ PRINT "? INVALID CAR ID CODE."          ! WRONGO           &
      \ GOTO 1030                               ! TRY AGAIN        &

1040  BLK=X%+1%                                 ! STARTING BLOCK   &
      \ INPUT "NUMBER OF RECORDS TO PRINT ";N.REC%  ! PRINT COUNT  &
      \ INPUT "PRINT REPORT TO ";REPO$          ! OUTPUT DEV/FILE  &
      \ OPEN REPO$ AS FILE 2%                   ! REPORT FILE      &
      \ PRINT #2%,HEDS                          ! PRINT HEADER     &
      \ PRINT #2%                               ! BLANK LINE       &

1080  FOR CT%=1% TO N.REC%                      ! LOOP FOR #REC    &
      \ GET #1%,BLOCK BLK                       ! GET TYPE 1ST BLK &
      \ GOTO 1090 IF CVT%$(NXT.BLK$)= -1%       ! END OF BLOCKS    &
      \ Q%=CVT%$(QUAN.AVA$)                     ! CONVERT QUANTITY &
      \ P=CVT$F(UNIT.PRI$)                      ! CONVERT PRICE    &
      \ V=P*Q%                                  ! COMPUTE VALUE    &
      \ PRINT #2% USING FORMLINES$,             ! FORMAT           &
       CVT%$(CAR.ID$),                          ! PRINT CAR TYPE   &
       DESC$+"  ",                              ! DESCRIPTION      &
       CVT%$(PART.NO$),                         ! PART NUMBER      &
       P,                                       ! UNIT PRICE       &
       Q%,                                      ! QUANTITY         &
       V                                        ! COMPUTED VALUE   &
      \ CNT%=CNT%+1%                            ! BUMP BLOCK COUNT &
      \ VALUE=VALUE+V                           ! SUM UP VALUES    &
      \ BLK=CVT%$(NXT.BLK$)                     ! NEXT BLOCK       &
      \ NEXT CT%                                ! NEXT PLEASE      &
                                                ! BLANK LINE       &
```

Figure SOF-14. NFS Block Mode (Page 1 of 2)


```
1090  PRINT #2%,SPACE$(55%)+"----------"        ! PRINT DIVIDER    &
      PRINT #2% USING "TOTAL RECORDS: ##";CNT%  ! FORMAT           &
      PRINT #2% "TOTAL RECORDS: ";CNT%          ! NUMBER RECS      &
      CLOSE 1%,2%                               ! CLOSE FILES      &
      \ GOTO 32767                              ! END IT           &

19222  ERROR HANDLING                                               &

19230  IF ERR=11% AND ERL=1030% THEN RESUME 1090      ! ABORT JOB   &

19999  ON ERROR GOTO 0                            ! JUST IN CASE    &

32767  END

Ready

RUNNH
READ FROM WHICH DISK(DDN:)? DK1:
CAR ID CODE ? 888
NUMBER OF RECORDS TO PRINT ? 15
PRINT REPORT TO ? KB:
```

CAR	MAKE & MODEL & STYLE	PART	PRICE	QH	EXTENDED VALUE
888	FIAT 124 4DR SEDAN PART	4511	$78.89	67	$5,285.63
888	FIAT 124 4DR SEDAN PART	4397	$72.48	39	$2,826.72
888	FIAT 124 4DR SEDAN PART	4211	$61.36	89	$5,461.04
888	FIAT 124 4DR SEDAN PART	2066	$21.74	44	$956.56
888	FIAT 124 4DR SEDAN PART	3785	$98.63	51	$5,030.13
888	FIAT 124 4DR SEDAN PART	7165	$33.75	57	$1,923.75
888	FIAT 124 4DR SEDAN PART	4156	$30.90	11	$339.90
888	FIAT 124 4DR SEDAN PART	9598	$17.88	44	$786.72
888	FIAT 124 4DR SEDAN PART	5555	$71.59	47	$3,364.73
888	FIAT 124 4DR SEDAN PART	2223	$73.09	38	$2,777.42
888	FIAT 124 4DR SEDAN PART	3087	$4.40	48	$211.20
888	FIAT 124 4DR SEDAN PART	5899	$36.94	90	$3,324.60
888	FIAT 124 4DR SEDAN PART	7418	$43.52	94	$4,090.88
888	FIAT 124 4DR SEDAN PART	4605	$41.55	34	$1,412.70
888	FIAT 124 4DR SEDAN PART	4769	$94.81	39	$3,697.59
TOTAL VALUE					$41,489.57

TOTAL RECORDS: 15

Ready

Figure SOF-14. NFS Block Mode (Page 2 of 2)

RMS-11 V1.0 and RSTS/E V7.0

General

DEFERRED WRITE

- Available to user through MACRO and COBOL
- Used by RMSCNV utility automatically

MASS INSERT

- Available to user through MACRO and COBOL
- Used by RMSCNV utility through a switch

NEW DATA TYPES FOR KEYS

- Packed decimal
- 2 and 4 byte signed and unsigned binary

FILE SPECIFICATION

- Standard RSTS/E file specification without switches supported

OVERLAY DESCRIPTION FILES

- RMSRLS.ODL — Resident Library for sequential files only
- RMSRLX.ODL — Resident Library for sequential/relative/indexed files
- RMS11S.ODL — Disk Library for sequential/relative files

- RMS11X.ODL — Disk Library for sequential/relative/indexed files. (9Kb)
- RMS12X.ODL — Disk Library for sequential/relative/indexed files. (11Kb)

Implementing New RSTS/E Features under RMS-11 V1.0

Table SOF-2.

Feature	How Supported
Large files	o Pre-extension through RMSDEF. o Explicit or implicit extension of non-contiguous files.
Data Caching	Through UTILTY
Extending non-contiguous files Opened in update mode	RMS-11 automatically handles
Placed files	Through RMSDEF
Tentative files	Unsupported
Making contiguous files non-contiguous	Through UTILTY
Non-file structured Block Mode	Unsupported

BASIC-PLUS-2 V1.60 and RSTS/E V7.0

Commands

SHOW - V1.60

- o Lists the current compiler defaults
- o Example

```
SHOW

RSTS/E BASIC PLUS 2 V01.57 using FPU  - 27 -
Installed on 23-Mar-79 at 03:30PM
Hiseg for RUN or /TSK is FPUIC2
Hiseg for BUILD is LB:FPUCOM
BASIC+2 Resident library is NONE
Disk library is LB:FPUCOM
RMS ODL file is LB:RMSRLX
RMS Resident library is LB:RMSRES
Case extend size = 1024
Scale factor = 3
Precision = Single
Switch settings:
        NO:MAP
        NO:DUMP
          :CHAIN
          :LINE
        NO:CLOUD
    Output:TSK with override to OBJ
  File orgs:Basic Plus compatible only

FPUIC2
```

Figure SOF-15. Show Command

ASSIGN/REASSIGN

- o ASSIGN DEV:[ppn]LIB Associates the logical name LIB with a device and account number

BRLRES

- o Allows the BASIC-PLUS-2 resident library to be linked to your program during task building
- o Example

```
BRLRES
FILE SPEC [NONE] -- BASICS
```


DSKLIB

- o Allows the BASIC2 or BP2COM disk library to be linked to your program during task building
- o Example

```
DSKLIB
FILE SPEC [LB:BASIC2]--
```

ODLRMS

- o Allows selection of RMS-11 ODL files
- o Example

```
ODLRMS
FILE SPEC [LB:RMS11S] -- LB:RMSRLX
```

RMSRES

- o Allows an RMS-11 resident library to be linked to your program during task duilding
- o Example

```
RMSRES
FILE SPEC [NONE] -- RMSRES
```

BUILD

- o Accepts new switches

```
/BRLRES:filespec
/DSKLIB:filespec
/ODLRMS:filespec
/RMSRES:filespec
/EXTEND:n
```

Statements

PRINT, RECORD

- PRINT with RECORD option supported

EXTEND/NO EXTEND - V1.5

- Recognized by compiler
- No action taken
- Allows RSTS/E CUSP compatibility

CLOSE -N%

- Supported for files opened as tentative with a MODE switch on the file specification

System Function Calls

- All the following new and enhanced system function calls are supported
  - File Name String Scan
  - Get Monitor Tables II
  - Change System Date/Time
  - Directory Lookup
  - Assign/Reassign
  - Deassign
  - Wild card PPN Lookup
  - Enable/Disable Disk Caching
  - File Utility
  - Resident Library
  - Return Job Status
  - Create a Job
  - Exit and Clear Program
- Unsupported System Function Call
  - Execute CCL

Implementing New RSTS/E Features under BASIC-PLUS-2 V1.60

Table SOF-3.

Feature	How Supported	
	Native Mode *	RMS
Large Files	Same as BASIC-PLUS (Virtual arrays limited to 65535 blocks)	o Pre-extension through RMSDEF. o Explicit and implicit extension of non-contiguous files
Data Caching	Through UTILTY or SYS calls	
Extending non-contiguous files Opened for Update	Same as BASIC-PLUS	RMS automatically handles
Placed files	Same as BASIC-PLUS	Through RMSDEF
Tentative files	Same as BASIC-PLUS or TEMPORARY Keyword**	Unsupported
Making contiguous files non-contiguous	Through UTILTY or SYS call	
Non-file structured Block Mode	Same as BASIC-PLUS	Unsupported

* BASIC-PLUS compatible
** New handling for RSTS/E V7.0

Using the RMS Resident Library

- Preliminary requirements
  a. Resident library support must be configured into the monitor.
  b. The resident library must have the correct protection code.
  c. The resident library must be added to the system via UTILTY or the system function call.
  d. RSX directive emulation in the monitor is not required by preferred.
- Procedures for using the resident library
  a. BUILD and task build procedures


```
$LOG IIO2A

FPUIC2

LISTNM
100     MAP (ONLY) A$(12), B$(5), C$(5), D,F(1,8)*2A
300     OPEN "CONLT.MAP" FOR OUTPUT AS FILE 2 &
        ,ORGANIZATION INDEXED FIXED &
        ,MAP ONLY, ACCESS APPEND &
        ,PRIMARY 99, ALTERNATE AS DUPLICATES
450     INPUT "IDENTIFICATION"; A$ &
        \INPUT "CODE"; B$ &
        \INPUT "LOCATION"; C$ &
        \INPUT "AMOUNT"; D &
        \PUT #2%
500     INPUT "ANOTHER RECORD"; ANS$YES$ &
        \IF ANS$YES$ = "YES" THEN 450
32767   CLOSE 2% &
        \END

FPUIC2

MISEG
Name (FPUCOM)--NONE
Account (LB:)--

FPUIC2

RMSRES
File spec (LB:RMSRES)--

FPUIC2

ODLRMS
File spec (LB:RMSRLX)--

FPUIC2

BUILD IIO2A/IND

FPUIC2

COM/OBJ/DOU

FPUIC2

TKB @IIO2A

FPUIC2


RUN IIO2A

IDENTIFICATION? MAILS
CODE? 16P
LOCATION? CJS
AMOUNT? 33228
ANOTHER RECORD? YES
IDENTIFICATION? MAILS
CODE? 10P
LOCATION? 333
AMOUNT? 567A$
ANOTHER RECORD? NO
```

Figure SOF-16. BASIC-PLUS-2 (page 1 of 2)

```
FPUIC2

DSP ONLT.MAS


ST:[120,120]ONLT.MAS    FILE ORGANIZATION:      INDEXED
PROLOGUE VERSION NUMBER:        1
CREATED: 22-MAR-1979 09:53
FILE PROTECTION:        <60>
RECORD FORMAT:          FIXED+52
RECORD ATTRIBUTES:      CARRIAGE RETURN
FILE ATTRIBUTES:
        ALLOCATION= 7   EXTEND QUANTITY=0       BUCKET SIZE=1
NUMBER OF KEYS:         2

FPUIC2

PIP

*II02%.CMD
ST:II02%/FP=ST:II02%/MP
UNITS = 12
ASG = ST:5:6:7:8:9:10:11:12
RESLIB=LB:RMSRES/RO
EXITSK= 102%
//
*II02%.OBL
        .ROOT BASIC2-RMSFOT-USER,RMSALL
USER:   .FCTR ST:II02%-LIBR
LIBR:   .FCTR LB:FPUCOM/LB
@LB:FPUIC%
@LB:RMSRLE
        .END
*^Z

FPUIC2
```

Figure SOF-16. BASIC-PLUS-2 (page 2 of 2)

COBOL V4 and RSTS/E V7.0

General

PERFORMANCE

- Improvements for CPU bound tasks through COMPUTATIONAL data formats.

> **Note**
>
> I/O bound tasks are affected very little.

DATA FORMATS

- COMP -- PDP-11 binary format
- COMP-6 -- Called COMP under previous releases
- COMP-3 -- Packed Decimal

Removal of Data Item Length Limitation

Previous to this version, the maximum length of any data item (group item, elementary item, or table) was 4095 characters. This limitation has been removed. Data item sizes are therefore limited only by the amount of memory available to the executable task and with the condition that all Data Division entries must be co-resident (non-overlayable).

Improved OTS Error Messages

COBOL-11, Version 4.6, includes a completely new object-time system. Error messages have been revised to provide clearer and more complete information. The new messages replace those contained in Appendix J of the PDP-11 COBOL User's Guide, Order No. AA-1757C-TC. The next revision of the User's Guide will include the new messages; however, they have been designed to require no additional reference.

Error message 15:

CBL -- 15: STOP RUN

no longer appears at program termination.

COBOL Interactive Debugger (CID)

The COBOL Interactive Debugger (CID) is an interactive debugging aid released with this version of PDP-11 COBOL. By using CID, which you include during the task-build process, you can:

- Examine and change data in your program during execution.
- Set and cancel breakpoints.

- Change program flow.

CID allows you to find program errors without recompiling or changing your program. When you want to use CID, you need only task-build the program with the CID module, then execute it.

Statements

ACCEPT

The ACCEPT identifier FROM DAY statements has been changed to agree with its description in the PDP-11 COBOL Language Reference Manual. Therefore, section 6.7.1 of the PDP-11 COBOL User's Guide will be revised to reflect this change.

Implementing New RSTS/E Features under COBOL V4

Table SOF-4.

Feature	How Supported
Large Files	o Pre-extension through RMSDEF. o Explicit or implicit extension of non-contiguous files.
Data Caching	Through UTILTY
Extending non-contiguous files Opened in Update Mode	RMS-11 automatically handles
Placed Files	Through RMSDEF
Tentative Files	Unsupported
Making Contiguous Files Non-Contiguous	Through UTILTY
Non-File Structured Block Mode	Unsupported

Using the RMS Resident Library

- o Preliminary requirements
  - a. Resident library support must be configured into the monitor.
  - b. The resident library must have the correct protection code.
  - c. The resident library must be added via UTILTY.
  - d. RSX directive emulation in the monitor is not required but preferred.
- o Procedures for using the resident library.
  - a. Merge and task build procedures

```
RUN   $PRO
PLEASE ENTER FILE SPECIFICATION FOR OUTPUT FILE
IALL1R.ODL
DO YOU WANT A DEFAULT MERGE?
PLEASE ANSWER Y(ES), N(O), OR H(ELP) N

DO YOU WANT AN ABBREVIATED OR MERGED ODL FILE?
PLEASE ANSWER A(BBREVIATED), M(ERGED), OR H(ELP) A

DO YOU WANT TO INCLUDE THE COBOL DEBUGGER (CID)?
PLEASE ANSWER Y(ES) OR N(O) N

DO YOU WANT TO OVERLAY I/O SUPPORT ROUTINES
OR USE THE RESIDENT I/O LIBRARY?
PLEASE ANSWER Y(ES), N(O), OR H(ELP) Y

DO YOU WANT TO USE THE RESIDENT I/O LIBRARY, THE DEC
SUPPLIED I/O OVERLAY OR YOUR OWN I/O ODL?
PLEASE ANSWER D(EC OVERLAY), O(WN OVERLAY),R(ESIDENT LIBRARY),
      OR H(ELP)R
PLEASE ENTER FILE SPECIFICATION FOR INPUT ODL FILE
IALL1C
OBJECT PROGRAM REFERENCED IN ODL FILE IS:
     IALL1C.OBJ
PLEASE ENTER OBJECT FILE DEVICE AND PPN IN THE FORMAT: DEV:[PROJECT,PROGRAMMER]

ANY MORE INPUT ODL FILES?
PLEASE ANSWER Y(ES) OR N(O) Y
PLEASE ENTER FILE SPECIFICATION FOR INPUT ODL FILE
SLPER
OBJECT PROGRAM REFERENCED IN ODL FILE IS:
     SLPER.OBJ
PLEASE ENTER OBJECT FILE DEVICE AND PPN IN THE FORMAT: DEV:[PROJECT,PROGRAMMER]

ANY MORE INPUT ODL FILES?
PLEASE ANSWER Y(ES) OR N(O) N
ODL FILE BUILD COMPLETE
MERGED ODL FILE IS: IALL1R.ODL

Ready
```

Figure SOF-17. COBOL Merge (Resident Library)


```
TY  IALL1R.ODL
;MERGED ODL FILE CREATED ON 23-MAR-79 AT 07:32:28
@IALL1C.ODL
@SLPER.ODL
CROBJ$: .FCTR IALL1C.OBJ-SLPER.OBJ-LB:COBOVR/LB:UIORSX
CROTS$: .FCTR LB:COBOVR/LB
        .NAME OTS$OV
CNVRT$: .FCTR LB:COBOVR/LB:CONVER
LDCVT$: .FCTR LB:COBOVR/LB:LDCVT
USRIO$: .FCTR LB:COBOVR/LB:USRSIO
OTSOV$: .FCTR OTS$OV-(CNVRT$-(*LDCVT$),*USRIO$)
@LB:RMSRLX.ODL
RMS$:   .FCTR  RMSROT-RMSALL
OBJRT$: .FCTR CROBJ$-CROTS$-RMS$
        .ROOT OBJRT$,OTSOV$
        .END

Ready



TKB
TKB>TKB  IALL1R,IALL1R=IALL1R/MP
ENTER OPTIONS:
TKB>LIBR=RMSRES:RO
TKB>//

Ready
```

Figure SOF-18. COBOL ODL and Task Build (Resident Library)

```
TKBIAL.TSK   MEMORY ALLOCATION MAP  TKB 07.005      PAGE 1
                    23-MAR-79   07:36


PARTITION NAME : GEN
IDENTIFICATION : V02072
TASK  UIC      : [1,201]
STACK    LIMITS: 001000 001777 001000 00512.
PRG XFR ADDRESS: 013172
TOTAL ADDRESS WINDOWS: 3.
TASK  IMAGE  SIZE  : 14208. WORDS
TASK ADDRESS LIMITS: 000000 067343


TKBIAL.TSK OVERLAY DESCRIPTION:

BASE    TOP         LENGTH
----    ---         ------
000000  057333  057334  24284.      IALLIC
057334  060567  001234  00668.           R3PUT
057334  061143  001810  00904.           R3UPDA
061144  061144  000000  00000.      OTSOV
061144  067337  006174  03196.           CONVER
067340  067343  000004  00004.                LDCVT
061144  066113  004750  02536.           USRSIO
```

Figure SOF-19. COBOL Map (Resident Library)

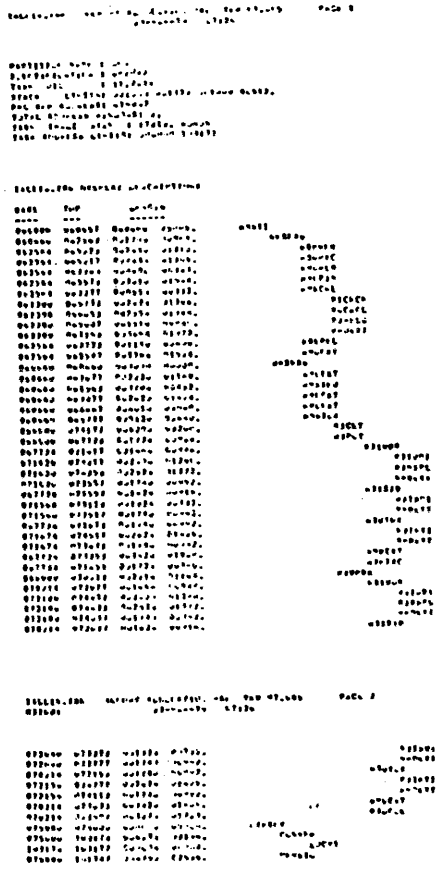


Figure SOF-20. COBOL Map (Without Resident Library)

```
TKBIAL.ISM   MEMORY ALLOCATION MAP  TKB 07.005      PAGE 13
                23-MAR-79   07:36

*** SEGMENT: USRSIO

R/W MEM  LIMITS: 061144 066113 004750 02536.
DISK BLK LIMITS: 000076 000102 000005 00005.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                          TITLE  IDENT  FILE
-------
. BLK.:(RW,I,LCL,REL,CON) 061144 000000 00000.
$USRIO:(RO,I,GBL,REL,OVR) 061144 004750 02536.
                          061144 004750 02536.  USRSIO 1A.18  COBOVR.OLB
$$ALVC:(RW,D,LCL,REL,CON) 061144 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 026406 000002 00002.


GLOBAL SYMBOLS:

$ALPHA 061764-R  $IOUTL 061144-R  $RSUTL 061202-R


*** TASK BUILDER STATISTICS:

     TOTAL WORK FILE REFERENCES: 338815.
     WORK  FILE  READS: 0.
     WORK  FILE WRITES: 0.
     SIZE OF CORE POOL: 21194. WORDS (82. PAGES)
     SIZE OF WORK FILE: 19944. WORDS (74. PAGES)

     ELAPSED TIME:00:01:13
```

Figure SOP-21. Task Build Statistics (Resident Library)


```
TKBIAL.ISM   MEMORY ALLOCATION MAP  TKB 07.005      PAGE 58
                23-MAR-79   07:26

*** SEGMENT: USRSIO

R/W MEM  LIMITS: 075000 101747 004750 02536.
DISK BLK LIMITS: 000266 000272 000005 00005.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                          TITLE  IDENT  FILE
-------
. BLK.:(RW,I,LCL,REL,CON) 075000 000000 00000.
$USRIO:(RO,I,GBL,REL,OVR) 075000 004750 02536.
                          075000 004750 02536.  USRSIO 1A.18  COBOVR.OLB
$$ALVC:(RW,D,LCL,REL,CON) 075000 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 030674 000002 00002.


GLOBAL SYMBOLS:

$ALPHA 075620-R  $IOUTL 075000-R  $RSUTL 075036-R


*** TASK BUILDER STATISTICS:

     TOTAL WORK FILE REFERENCES: 344509.
     WORK  FILE  READS: 919.
     WORK  FILE WRITES: 541.
     SIZE OF CORE POOL: 21194. WORDS (82. PAGES)
     SIZE OF WORK FILE: 37376. WORDS (146. PAGES)

     ELAPSED TIME:00:01:56
```

Figure SOP-22. Task Build Statistics (With Resident Library)

MACRO and RSTS/E V7.0

General

MONITOR CALLS

- All monitor calls are documented in the RSTS/E System Directives Manual
- All RSTS/E V7.0 features available through monitor calls

RMS Features

- Full RMS capability

> **Note**
>
> Use of RT11 directives are not documented and therefore are not supported.

RSTS/E V7.0 SWS Update

UUP

User Utility Programs

User Utility Programs

CONTENTS

Introduction..... UUP-1

FIT Program..... UUP-3
- Additional FIT Documentation..... UUP-3
- Transferring Files with FIT..... UUP-4
- Maintaining RT11 Format..... UUP-7
- DOS Disk Directory Listings..... UUP-11

RMSIFL Program..... UUP-12
- Additional RMSIFL Documentation..... UUP-12
- RSMIFL Utility Switches..... UUP-20

RMSCNV Program..... UUP-29

RMSDEF Program..... UUP-30

DIRECT Program..... UUP-32

BPCREF Program..... UUP-33

COPY Program..... UUP-37

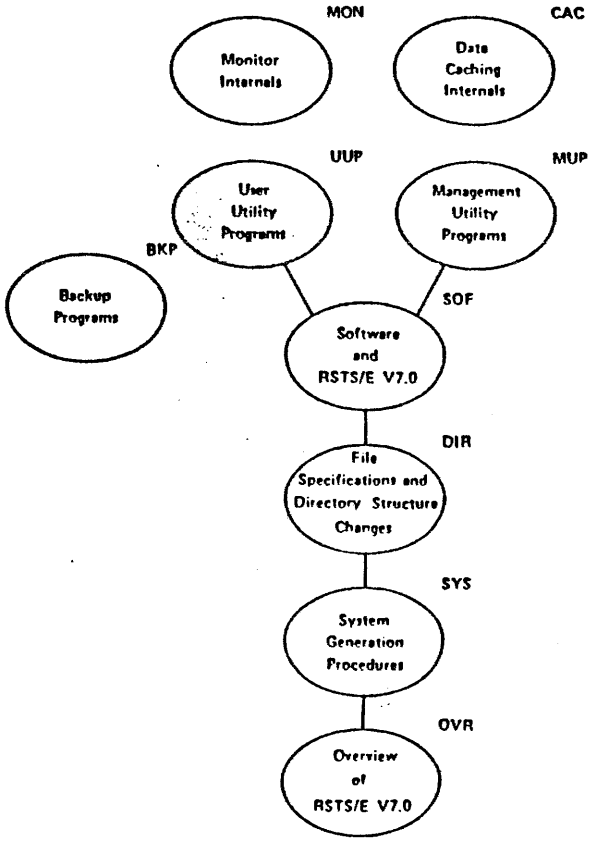
FLINT Program..... UUP-37

FILCOM Program..... UUP-38

GRIPE Program..... UUP-40

QUOLST Program..... UUP-40

COURSE MAP


MON
Monitor Internals
CAC
Data Caching Internals
UUP
User Utility Programs
MUP
Management Utility Programs
BKP
Backup Programs
SOF
Software and RSTS/E V7.0
DIR
File Specifications and Directory Structure Changes
SYS
System Generation Procedures
OVR
Overview of RSTS/E V7.0

User Utility Programs

INTRODUCTION

As RSTS/E users you know the wide range of services offered by the CUSPs supplied with a RSTS/E system. Some of these services include:

- o File manipulation
- o File transfer
- o Directory listing
- o Batch processing
- o File spooling

RSTS/E V7.0 includes new and revised utilities which provide you with greater flexibility in handling your day to day and long term needs.

In this module you will learn these new and revised non-privileged utility programs.

This page is for notes.

FIT PROGRAM

- Transfers files between

  Disk and TU58
  Disk and floppy
  RSTS/E disk and RT11 disk
  RSTS/E disk and RT11 DECtape

- Transfers files from a DOS disk to a RSTS/E disk

- Lists directory of a TU58 and floppy

- Lists directory of a DOS disk

Additional FIT Documentation

TU58 DECtape II and floppy disk files are maintained in the RT11 directory structure format. Thus, DECtape II and floppy disks are directly transferrable between RSTS/E and RT11 systems.

To invoke the FIT program, you must be logged into the system and type the following command:

```
RUN $FIT
```

If FIT is successfully invoked, it prints an identifying message followed by a FIT> prompt. The prompt indicates that FIT is prepared to accept a command line. To cause FIT to print a help text file, type the /HE switch in response to the prompt. To exit from FIT, type CTRL/Z in response to the prompt, for example:

```
FIT>^Z
Ready
```

Transferring Files with FIT

To transfer a file with FIT, type a command line of the following form in response to the FIT prompt:

```
FIT>[output[/sw]=]input[/sw]
```

where output and input are file specifications and /sw is an optional switch that specifies the structure of the device on which the file resides (RSTS/E, RT11, or DOS).

The input and output file specifications have the following form:

```
dev:[proj,prog]name.ext<prot>
```

where:

dev:	is a device specification. If none is specified, the default is the public structure. The output device must not be a DOS disk. The input device can be a RSTS/E, RT11, or DOS disk; or RT11 TU56 DECtape; or RT11 TU58 DECtape II; or RT11 floppy disk with RT11 file structure.
proj,prog	is a project-programmer number. If none is specified, the default is your current account. FIT ignores the project-programmer number if the device has an RT11 file structure.
name.ext	is a filename and extension. If none is specified for output, FIT uses the input filename and extension. An input filename must be specified.
<prot>	is a protection code. If none is specified on output, FIT assigns your current default protection code. FIT ignores any input file protection code specification.

The legal input device specifications for a FIT file transfer are as follows:

- RSTS/E disk
- floppy disk (RT11 format)
- TU58 DECtape II (RT11 format)
- RT11 disk (with pack clustersize = 1) or TU56 DECtape (RT11 format)
- DOS disk

The legal output device specifications (except for DOS input) for a FIT file transfer are as follows:

- RSTS/E disk
- floppy disk (RT11 format)
- TU58 DECtape II (RT11 format)
- RT11 disk (with pack clustersize = 1) or TU56 DECtape (RT11 format)
- line printer, keyboard, etc.
- DECtape (RSTS/E format)

If the input device is a DOS disk, the legal output devices are as follows:

- RSTS/E disk or DECtape
- line printer, keyboard

The switch specification /sw can be one of the following:

/RT11	The file is on an RT11 file structured device
/DOS	The file is on a DOS format disk (legal only for an input file specification)
/RSTS	The file is on a RSTS/E disk or device. This switch is illegal with floppy disk and TU58 DECtape II.

These switches can be abbreviated to the first two letters (for example, /RT, /DO, and /RS).

If you do not specify a switch on the input specification, FIT attempts to determine the file structure of the input file device. If the input device does not contain a valid RSTS/E, RT11, or DOS file structure, FIT prints the following error:

?INPUT DEVICE NOT FILE STRUCTURED

If the input device is a RSTS/E format DECtape, FIT prints the following error:

?ILLEGAL INPUT DEVICE

If the specified file does not exist on the device, FIT prints the following error:

?CAN'T FIND FILE OR ACCOUNT

If you do not specify a switch on the output specification, FIT assumes a default file structure based on the type of device. That is, if the device is TU58 DECtape II or floppy disk, FIT assumes the RT11 file structure; if the device is disk or TU56 DECtape, FIT assumes the RSTS/E file structure.

If FIT determines that the output device should have an RT11 file structure, the following errors are possible:

?NO RT11 STRUCTURE ON OUTPUT

The output device does not contain a valid RT11 file structure. Check that the right device was specified. An RT11 structure must already exist on the output device before files can be transferred to it. The /ZE switch can be used to create an RT11 structure on the device.

?DIRECTORY OVERFLOW

The output device has no space for directory entries. However, it may be possible for you to use the /SQ switch to free more space for directory. If this is not successful, you must delete some of the files on the device before more can be added. Note that if you intend to have many small files on the device, you should use the /N switch to allocate more than the default number of directory segments when the device is zeroed.

?DEVICE FULL

The output device has no free space for the file transfer. It may be possible for you to use the /SQ switch to create enough contiguous free space for the additional files. If this is not successful, you will have to delete enough files from the output device to add the files from the input device.

These errors are generated prior to the transfer of any data to the RT11 structured device. If an error is encountered during a transfer, a RSTS/E error is reported and the transfer is aborted. Because the directory entry for the RT11 file is not finalized until the transfer is complete, an error during transfer causes the space reserved for the file to be freed.

If an error is encountered during the open or transfer phase of a RSTS/E file operation, the appropriate RSTS/E error is generated and the transfer is aborted.

Maintaining RT11 Format

In addition to transferring files to and from an RT11 directory structured device, FIT can perform the following operations:

1. You can use the /ZE switch to zero and initialize a device as an RT11 device. This operation creates an RT11 directory structure on the device and must be performed prior to transferring files to and from the device.

2. You can use the /LI switch to obtain a directory listing of the RT11 device.

3. You can use the /DE switch to delete a file on an RT11 device.

4. You can use the /SQ switch to compress the data and directory of the device and allow more files to be stored on that device. This operation is useful if the available space on the device is fragmented.

File Deletion on an RT11 Format Device -- To delete a file on an RT11 directory structured device, enter the following command format in response to the FIT> prompt:

```
FIT>dev: filename.ext/DE
```

where dev: identifies the device and filename.ext specifies the file to be deleted. If the specified device is not RT11 directory structured or if the file does not exist, FIT prints an error message. If an error is encountered by FIT as it accesses the device, a RSTS/E error is reported.

RT11 Device Switches -- You can perform zeroing, compressing, and listing operations on RT11 directories by entering the following command format in response to the FIT> prompt:

```
FIT>dev:/sw[/sw][/sw]
```

where dev: indicates the device on which the operation is to be performed and /sw specifies the operation. The switches you specify can be one or more of the following:

/ZE:m — Zero the device and build an RT11 directory structure. If :m is specified, m extra words of directory information for each file are added to the minimal RT11 directory structure. If m is not specified, FIT adds 13 extra words, by default, to the directory structure. These words are used to store the run-time system name and file attributes.

Note that RT11 systems currently ignore these extra directory entries. However, if future releases of RT11 attach significance to these words, a specification of/ZE:0 may be required for RT11 files that are read by RT11 systems. FIT currently ignores extra directory information unless there are 13 extra words (this may change in future releases). Therefore, you should only specify values of 0 or 13 in the /ZE switch.

/N:n — Used with the /ZE switch to cause n directory segments on the initialized device. Each segment consists of two 512-byte blocks. The value of n must be in the range of 1 to 31 and determines the size of the directory on the device (and, therefore, the number of files that can be stored on the device). The number of files that can fit on a device is determined by the following formula:

```
n*INT(507/m+7)
```

where n is the number of directory segments and m is the number of extra words for each directory entry (that is, /N:n and /ZE:m, respectively).

Note that you may have to compress the device with the /SQ switch before the maximum number of files is reached.

If you do not specify n, FIT applies defaults based on the device type. These defaults are indicated in the following table. The table also lists the maximum number of files allowed on a particular device corresponding to your specification in /ZE:n.

Table UUP-1. Defaults by Device Type

Device	Default number of Directory segments	Max. number of files /ZE:0	Max. number of files /ZE:13
TU58 DECtape II	4	288	104
Floppy disk	4	288	104
RK05 disk	16	1152	416
RK06 disk	31	2232	806

/LI or /DI — List the directory of the device. The information listed includes filenames, extensions, file size, protection codes, dates of creation, and position on the device. If the device has 13 extra words for each directory entry (/ZE:13, the default value), the run-time system name and file attributes are also printed. Those sections of the device which are free space are marked <UNUSED> in the listing.

/RT11 — Used with the /LI or /DI switches. /RT11 indicates that the device must have RT11 directory structure. That is, if this switch is not present and the device is a DOS disk, a directory of the DOS disk is listed. If this switch is present and the device is a DOS disk, an error is returned.

/SQ — Compress (squeeze) the device. This switch allows you to compress the files that are currently on the output device and make room for files that are to be transferred. The space on RT11 devices that is allocated to the directory or files can become fragmented. If this is the case, an attempt to add another file to the device can result in a ?DIRECTORY OVERFLOW or ?DEVICE FULL error. You can use the /SQ switch to compress the contents of the device and provide enough room for the additional files.

Note that because of the manner in which RT11 directory entries are made, you may need to compress the device to use the directory space even if files are added contiguously without deletions.

All of these switches can be abbreviated to one letter (for example, /Z, /N, /L, /D, /R, and /S).

DOS Disk Directory Listings

To list the directory of a DOS disk, enter the following command format in response to the FIT prompt:

```
FIT>dev:[/DO[S]]/L[I]
```

where dev: is the device designator of the DOS disk. The switch /D or /DI can be used in place of /L or /LI. If the /DOS switch is omitted, FIT lists the directory of the device whether it is DOS or RT11 directory structured. If the /DOS switch is present, an error is returned if the device is not a DOS disk.

The listing generated by this operation includes the following information: account numbers (UICs), filenames, extensions, file lengths, protection codes, dates of creation, the first and last block numbers of the files, and whether or not the file is contiguous.

RMSIFL PROGRAM

- Provides fastest population of multi-key indexed files
- Input file can have any RMS file organization
- Output file must be previously created and must be empty

Additional RMSIFL Documentation

Purpose

The RMSIFL utility provides the fastest, most direct method of populating an empty RMS-11 Indexed file. The utility can read records from any type of RMS-11 file and load them into an Indexed file that you have already created. However, RMSIFL does not use the standard RMS-11 access methods to build the output file. It exploits the basic structure of Indexed files to insert data records and construct indexes as fast as possible.

Although RMSCNV may compete with this utility when populating a single-key indexed file, RMSIFL is definitely faster when it comes to multi-key file loading. However, you must use RMSCNV if the output file is not empty.

Effect

RMSIFL reads records from an input file and loads them into an output Indexed file in a series of phases.

1. The utility examines the command string, rejecting it for syntax and/or other errors. When it has a valid command string, RMSIFL determines if the input file must be sorted into ascending order by the output file's Primary Key.

   RMSIFL bypasses the sort phase (phase 2) and skips to phase 3 if:

   - You included the Infile Switch NOSO.

   - The key of reference specified for an Indexed input file is equal to or contains the output Primary Key, based on position only. Nonstring key types must match exactly, in starting position and length. String keys must start in the same position, but the Primary Key can be the same length as or shorter than the input key of reference.

     You specify key of reference with the KR Infile Switch.

     Example The ten-byte string input Alternate Key 2 starts at byte 0 and is specified as the key of reference. The eight-byte string output Primary Key starts at byte 0. RMSIFL does not sort.

     Example The ten-byte string input Primary Key starts at byte 0. The eight-byte string output Primary Key starts at byte 3. RMSIFL sorts the input file.

     Example The three-byte packed decimal input Alternate Key 1 starts at byte 43 and is specified as the key of reference. The three-byte packed decimal output Primary Key starts at byte 42. RMSIFL sorts the input file.

     RMSIFL notifies you that it has started processing by printing the following message:

     PRIMARY KEY:

2. RMSIFL reads the input file in Sequential Access Mode, depending on the organization of the input file:

INPUT FILE ORGANIZATION	PROCESSING
SEQUENTIAL	RMSIFL reads the records in the Sequential Access Mode, starting with the first record in the file.
RELATIVE	RMSIFL reads the records in the Sequential Access Mode, starting with record cell 1.
INDEXED	RMSIFL reads the records in the Sequential Access Mode, following the key of reference specified in the command string (the Primary Key is default).

RMSIFL passes each record to modules extracted from the SORT-11 utility. These modules order the input records by ascending value in the Primary Key field of the output file.

At the beginning of this phase, RMSIFL allocates disk space for three sort work files, each of which is the size of the 1.5 times the input file.

> **Note**
>
> Performance in this phase depends entirely on the SORT-11 utility. See the PDP-11 SORT Reference Manual for information on the space required for temporary sort files and guidance on the factors affecting sort performance.

RMSIFL notes this phase by printing the following messages:

```
SORT HAS STARTED
SORT MERGE PHASE HAS FINISHED
```


3. RMSIFL loads the ordered input records into the output file, constructing Level 0 of the Primary index one bucket at a time.

   The manner in which RMSIFL reads input records depends on whether the input file was sorted in phase 2:

   - If RMSIFL sorted the input file, the utility accepts the input records one at a time from the sort modules. This technique avoids the need for a temporary file containing the input records sorted in output Primary Key order.
   - If RMSIFL did not sort the input file, the utility reads the records from the input file.

   RMSIFL examines each record to ensure that it is compatible with the output file:

   - If the record is not compatible, RMSIFL processes it as an exception record according to the switch you have specified in the command string. See the discussion of the ER:[filespec] and NOER Infile switches for the causes of exception records and their handling.
   - If the record is compatible, RMSIFL inserts it into a Level 0 bucket being built in memory.

   As the utility places each record, the utility extracts the values for each of the Alternate Keys defined for the file, if any:

   a. RMSIFL converts nonstring key data types into a stringlike representation for sorting only.

   b. RMSIFL combines the Alternate Key value with the Record's File Address (RFA) of the user data record that supplied the key value. RFAs are six bytes long. The utility then appends a byte with the key number and the converted version of the key value.

c. These fixed-length Alternate Key records are written into a separate temporary file named IFL0.TMP. RMSIFL creates this file with an initial allocation of 100 blocks and a Default Extension Quantity of 100 blocks. You can calculate the final size of IFL0.TMP (ALQ) as follows:

ALQ = NDRF * NAK * (LKS + 7 + PAD)

where NDRF is the number of user data records in the input file,

NAK is the number of Alternate Keys defined for the output file

LKS is the size of the longest Alternate Key defined for the output file (and shown by RMSDSP), in bytes, and

PAD is the number of null bytes added to word align the record:

PAD = 0 if LKS is an even number

PAD = 1 if LKS is an odd number

Remember to round ALQ to the nearest multiple of 100.

During this process, the utility assembles the index levels of the Primary index as well. RMSIFL puts an index record into a Level 1 bucket for each Level 0 bucket it writes to the file. When it writes a Level 1 bucket to the file, the utility inserts an index record into the Level 2 bucket, and so on, up to and including the Root.

4. RMSIFL sorts the Alternate Key temporary file into order by key number and ascending key value.

At the beginning of this phase, RMSIFL allocates the disk space for sort work files, each of which is 1.5 times the size of IFL0.TMP.

The utility notes this phase by printing the following messages:

ALTERNATE KEY
SORT HAS STARTED
SORT MERGE PHASE HAS FINISHED

5. RMSIFL builds the Alternate indexes one at a time, from Level 0 up to the Root, using the sorted temporary file for input.

The utility notes the construction of each alternate index by printing the following message:

ALTERNATE KEY(s) nnn

where nnn starts with 1.

6. RMSIFL prints the following messages when the output file is completely built, then terminates the processing of the current command string.

NUMBER OF INPUT RECORDS: nnnnn
NUMBER OF OUTPUT RECORDS: nnnnn
NUMBER OF EXCEPTION RECORDS: nnnnn

> **Note**
>
> RMSIFL extends the output file if it cannot contain the input records and the index(es). The utility minimizes this overhead by extending the file a set quantity each time it requires space. This quantity is the integral multiple of a bucketsize greater than 50.

Call and Termination

Using the CCL command -

IFL [command string]

If you include a command string, the utility attempts to execute it and then returns control to your default run-time system.

If you do not specify a command string, RMSIFL assumes control and prints the prompt.

```
IFL>
```

You may enter a command string or CTRL/Z to terminate the utility. When RMSIFL has executed a command string, it prints the prompt.

Using the RUN command -

```
RUN $RMSIFL
```

The utility assumes control and prints the prompt:

```
IFL>
```

You may type a command string or CTRL/Z to terminate the utility. When RMSIFL has executed a command string, it reprints the prompt.

Command String

General Form -

outfile[/switch]..]=infile[/switch]..]

where outfile is the filespec of an existing Indexed file that is to receive the records of the input file. Wild card characters are not permitted in any field of the specification.

RMSIFL checks for the following conditions in the output file. If they are not met, the utility prints the indicated error message and terminates execution of the command string.

- The output file must exist before you run the utility, otherwise, RMSIFL prints the message:

  ?IFL--FILE NOT FOUND - filespec

- The output file must be empty except for the prologue written when the file was created. Otherwise, RMSIFL prints the message:

  ?IFL--INDEX OUTPUT FILE NOT EMPTY

- The output file must be an RMS-11 Indexed file. Otherwise, RMSIFL prints the message:

  ?IFL--OUTPUT FILE MUST BE AN INDEX FILE

- The output file must not have a bucket size greater than five blocks. Otherwise, RMSIFL prints the message:

  ?IFL--THERE IS NOT ENOUGH IFL MEMORY FOR THE CURRENT COMMAND LINE

- The output file cannot have more than ten keys defined. Otherwise, RMSIFL prints the message:

  ?IFL--TOO MANY KEYS - filespec

infile is the filespec of the input file that is the source of records to be written to the output file. Wild card characters cannot appear in any field of the filespec.

You can specify only one input file, but it can have any RMS-11 file organization. RMSIFL does not affect the input file in any way.

The utility opens the input file access read, allow no-write. Therefore, no writing task can access the input file while RMSIFL is operating.

The input file cannot have more than ten keys defined. Otherwise, RMSIFL prints the message:

?IFL--TOO MANY KEYS - filespec

You cannot use a back-up file produced by RMSBCK as input. RMSIFL terminates with the following error message:

?IFL--OPEN ERROR ON FILE - filespec
?IFL--FATAL RMS ERROR STS = -01424, STV = 00000

switch may be one of the codes shown in the table below and described on the following pages.

> **Note**
>
> A command string may also consist of the word "HELP" or a question mark (?). RMSIFL responds with a HELP message.

RMSIFL Utility Switches

Type	Switch	Description	Default Process
String	? or HELP	Print HELP message.	No help.
Global	/ID	Identify current version	No id.
Outfile	/ER	Write exception records on terminal.	Same.
	/ER:filespec	Write exception records into specified file.	Write exception records on terminal.
	/LO	Honor fill numbers	Fill buckets.
	/NOER	Stop processing immediately if input record is incompatible.	Write exception records on terminal.
	/PD[:[#]x]	Pad input records to output record length.	Handle input as exception record if different lengths.
	/TR	Truncate input records to output record length.	Handle input as exception record if different lengths.
Infile	/DE:dvn[:dvn [dvn[:]]]	Reassign devices for sort work files.	Create and use sort work files on SYSDEV.
	/KR:n	Key of reference number.	Primary Key (n=0).
	/NOSO	Do not sort before loading.	Sort input file before loading.

Global Switches -

- ID causes RMSIFL to print the version number of the utility that is operating in the form:

  IFL -- VERSION 1.8nn

  where nn is the patch level of the utility.

  This switch may appear alone on a command string.

Outfile Switches

- ER:[filespec] directs RMSIFL to continue processing if it encounters a record in the input file that cannot be written into the output file. Exception records can exist for one of the following reasons:

  - You specified the NOSO Infile Switch and the utility found a record that was not in ascending collating order by Primary Key value. The error message is:

    ?IFL--EXCEPTION RECORD: RECORD OUT OF SEQUENCE

- You did not allow duplicates in at least one key of the output file, but an input record contains a duplicate value in that key value. The error message is:

  IFL--EXCEPTION RECORD: DUPLICATE KEY WHERE NOT ALLOWED

- An input record is not long enough to contain the Primary Key of the output file. The error message is:

  IFL--EXCEPTION RECORD: RECORD TOO SHORT FOR PRIMARY KEY

- An input record is too long to fit in a bucket of the output file. The error message is:

  IFL--EXCEPTION RECORD: RECORD TOO LONG FOR BUCKET SIZE

- An input record is not compatible with the output file's fixed-length record format and you have not specified the PD and/or TR switches. The error message is:

  IFL--EXCEPTION RECORD: RECORD DOESN'T FIT FIXED LENGTH

- An input record is longer than the output file's variable-length Maximum Record Size and you have not specified the TR switch. The error message is:

  IFL--EXCEPTION RECORD: RECORD TOO LONG FOR MAXIMUM RECORD SIZE

If RMSIFL encounters an exception while it is processing the Alternate Index(es), it flags the data record in the Primary Level 0. The utility does not remove the record from the file. The record may be compressed during later use of the Indexed file.

You can add corrected exception records to the output file after RMSIFL terminates with either an application program or RMSCNV (possibly using your terminal as the input device).

If you do not specify an argument (filespec), RMSIFL prints the appropriate error message, then the exception record on your terminal.

If you specify a nondisk file or a terminal, the utility creates it as an RMS-11 Sequential file with variable length records when the first exception record is encountered. Then, RMSIFL writes both the error message and the exception record into the file. RMSIFL uses this procedure for every exception record until it finishes processing.

If you specify a disk file, the utility creates it as an RMS-11 Sequential file with VFC records when the first exception record is encountered. Then RMSIFL writes the exception record into the variable portion of the record and a four-byte exception code into the fixed control area. The codes are:

Code	Meaning
001	Record is out of Primary Key sequence.
002	Record contains an illegal duplicate value.
003	Record is too short to contain output Primary Key.
004	Record is too long to fit in output bucket.
005	Record is not correct size for output fixed length record.
006	Record is too long for output Maximum Record Size.

Also see the NOER switch.

Default If you do not include the ER switch, RMSIFL prints an appropriate error message first, then the exception record on your terminal.

- LO directs RMSIFL to write records into buckets according to the fill numbers established when the file was created.

  Default If you do not include the LO switch, RMSIFL inserts as many records as possible into each bucket.

- NOER[:S] directs RMSIFL to print an error message, delete the output file, and terminate processing immediately if it encounters an input record that cannot be written into the output file; see the ER:[filespec] switch for reasons and error messages.

RMSIFL deletes the output file unless you specify the SAVE argument (:S) in the NOER switch. Termination at this point leaves the output file incomplete: it is not a valid file organization and cannot be used for any RMS-11 purpose.

Default If you do not include the NOER switch, RMSIFL responds according to the version of the ER switch you used.

- PD[:[#]x] directs RMSIFL to pad records read from the input file to the output file's record length before writing them to the output file. Padding character is specified as follows:

Switch	Character
PD	NULL
PD:x	x is ASCII A-Z, 0-9, or special character except #, ?, and @
PD:#x	x is octal number 000-377 (40 for SPACE, 43 for #, 77 for ?, and 100 for @)

You use the PD switch only when the output file specifies fixed-length records.

Default If you do not include the PD switch, and the input records are shorter than the output file's records, RMSIFL treats them as exception records.

See the ER and NOER switches.

- TR directs RMSIFL to truncate records read from the input file to the output file's Maximum Record Size before writing them to the output file. The trailing bytes of the records are truncated.

  Default If you do not include the TR switch, and the input records are longer than the input file's record RMSIFL treats them as exception records.

See the ER and NOER Switches.

Infile Switches

- DE:dvn[:dvn[:dvn[:]]] directs RMSIFL to reassign the devices where it allocates the three sort work files. You can specify from one to three disk devices with the switch, where dvn is a physical device name and number or a logical name up to six alphanumeric characters. You separate device names by a colon (:); the final colon is optional.

If you include the DE switch in the command string, RMSIFL reassigns all three sort work files:

- If you specify one device name, the utility reassigns the three work files to that device.
- If you specify two device names, the utility reassigns one work file to the first device and two work files to the second.
- If you specify three device names, the utility reassigns one work file to each device.

For optimum performance, you should assign each sort work file to a different device. However, if you have only two devices available, assign the first and third files to one device and the second file to the other device.

Example /DE:DM0:DM1:DM2

Example /DE:DM0:DM1:DM0

Default RSMIFL allocates the sort work files on SY:

- KR:n directs RMSIFL to read an input Indexed file according to the key specified by n, where n is 0 for the Primary Key, 1 for the First Alternate Key, and so on.

The KR switch can eliminate the need for sorting the input file; see the "Effect" section.

If you use the KR switch with a Non-Indexed file, RMSIFL terminates with the following message:

?IFL--/KR NOT ALLOWED FOR SEQUENTIAL OR RELATIVE FILE

Default If you do not include the KR switch, and the input file is Indexed, RMSIFL reads the file via the Primary Index.

- NOSO directs RMSIFL to bypass its sort phase of processing because you have already sorted the input file. If the input file is not in order by ascending value of the output file's Primary Key, RMSIFL's response depends on whether you have included the NOER switch or a version of the ER switch.

Default If you do not include the NOSO switch, RMSIFL sorts all records from the input file into the proper order. See the "Effect" section.

Cautions

- RMSIFL ignores the fixed control area of VFC records in both the input and output files.

- When both the input and output files have fixed-length records, RMSIFL requires either the TR or PD switch if the fixed record lengths are not equal. If neither switch is specified, the utility terminates with the error message:

  ?IFL--INPUT AND OUTPUT RECORD SIZES DO NOT CORRESPOND.

- When the input file contains variable-length records and the output file contains fixed-length records, RMSIFL requires both the TR and PD switches. If both are not specified, the utility terminates with the error message:

?IFL--SWITCH /TR OR /PD OR BOTH ARE NEEDED FOR THIS RECORD

- Example

```
RUN $RMSIFL

IFL>?

                RMS-11 INDEXED FILE LOAD UTILITY
                --------------------------------
          FUNCTION: TO BUILD INDEXED FILES
          FORMAT  : IFL OUTFILE[/OPTIONS]=INFILE[/OPTIONS]

   GLOBAL OPTIONS:
       ? OR HELP              TYPE THIS MESSAGE
       /ID                    IDENTIFY CURRENT VERSION AND PATCH LEVEL

   OUTFILE OPTIONS:
       /ER[:FILESPEC]         WRITE EXCEPTION RECORDS TO TERMINAL
                              (OR TO FILESPEC)
       /LO                    HONOR FILL NUMBERS
       /NOER[:S]              STOP PROCESSING IF INPUT RECORD IS
                              INCOMPATIBLE AND DELETE THE OUTPUT FILE
                              (IF :S IS SPECIFIED SAVE OUTPUT FILE)
       /PD[:[#]X]             PAD INPUT RECORDS TO OUTFILE RECORD LENGTH
       /TR                    TRUNCATE INPUT RECORDS TO OUTFILE RECORD LENGTH

   INFILE OPTIONS:
       /DE:DVN[:DVN[:DVN[:]]] REASSIGN DEVICES FOR SORT WORK FILES
       /KR:NNN                KEY-OF-REFERENCE NUMBER
       /NOSO                  DO NOT SORT INPUT BEFORE LOADING
IFL>NEW.IDX=ONLY.MAS

           KEY CHECK
           PRIMARY KEY:
       SORT HAS STARTED
       SORT MERGE PHASE HAS FINISHED
       NUMBER OF INPUT RECORDS: 1
       NUMBER OF OUTPUT RECORDS: 1
       NUMBER OF EXCEPTION RECORDS: 0
IFL>^Z

Ready
```

Figure UUP-1. RMSIFL

RMSCNV PROGRAM

- Uses deferred write feature of RMS
- Allows a switch to specify mass insert feature of RMS
- Example

```
RUN $RMSCNV

CNV>ONLY1.MAS/MA/FO:IDX=ONLY.MAS

CNV>^Z
?Task exit status: SEVERE ERROR

Ready
```

Figure UUP-2. RMSCNV

RMSDEF PROGRAM

- Allows specification of new RMS key data types
- Allows placement of RMS files
- Example

```
RUN RRMSDEF

DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE(NO)?YES

ENTER FILE SPECIFICATION FOR COMMAND FILE:ONLY1.CMD

DO YOU WANT TO CREATE THE FILE YOU WILL BE DESCRIBING(NO)?YES

ENTER FILE SPECIFICATION:ONLY1.MAS

IF THE FILE ALREADY EXISTS, DO YOU WANT TO SUPERSEDE IT(NO)?YES

ENTER FILE ORGANIZATION(SEQ):IDX

ENTER RECORD FORMAT(VAR):FIX

ENTER MAXIMUM RECORD SIZE:62

DO YOU WANT CARRIAGE RETURN CONTROL(YES)?NO

DO YOU WANT FORTRAN CHARACTER CONTROL(NO)?

IT'S TIME TO DEFINE THE PRIMARY KEY
ENTER DATA TYPE(STR):

ENTER POSITION OF KEY:10

ENTER SIZE OF KEY:5

ENTER NAME OF KEY(NONE):

WILL YOU ALLOW DUPLICATE KEYS(NO)?YES

DO YOU WANT TO DEFINE MORE KEYS(NO)?

DO YOU WANT TO DEFINE AREAS(NO)?

DO YOU WANT PLACEMENT CONTROL(NO)?YES

ENTER LOCATION:500

ENTER INITIAL ALLOCATION IN BLOCKS(3):100

ENTER BUCKET SIZE(1):4

ENTER DEFAULT EXTENSION QUANTITY IN BLOCKS(3):4

DO YOU WANT A CONTIGUOUS FILE(NO)?YES

THESE QUESTIONS ARE FOR THE PRIMARY KEY:

THE BUCKET SIZE IS 4 BLOCKS.
ENTER FILL NUMBER FOR DATA BUCKETS(3):50

ENTER FILL NUMBER FOR INDEX BUCKETS(3):70

ENTER CLUSTERSIZE(3):125

ENTER PROTECTION CODE(48):48

YOUR FILE HAS BEEN OPERATED!!! -- SY:[1,235]ONLY1.MAS
```

Figure UUP-3. RMSDEF (Page 1 of 2)


```
YOUR FILE HAS BEEN PROCESSED AND A COMMAND FILE GENERATED!!! --SY:ONLY1.CMD

DO YOU WANT TO CLOSE THE COMMAND FILE(NO)?YES

DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE(NO)?^Z

Ready

DIR ONLY1.MAS/S
 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
SY:[1,235]
ONLY1 .MAS      100CL < 48> 19-Mar-79 19-Mar-79 01:06 PM 128 ...RSX 41425  0/0
 RF:FIX=62   FO:IDX   USED:101:0      RECSI:62
  BK:4

Total of 100 blocks in 1 file in SY:[1,235]

Ready
```

Figure UUP-3. RMSDEF (Page 2 of 2)

DIRECT PROGRAM

- New Switches

/SZ	List file names, extensions and sizes (same as /SI)
/OP	List open and read regardless file access counts
/MD	List files marked for deletion
/PO	List file's position on disk
/AT or /SA	List attributes symbolically
/OA	List attributes in octal

- Example

```
DIR ONLT.MAS/S,SEQ2.BAK/S
 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
SY:[1,235]
ONLT  .MAS      11   < 60> 15-Dec-78 15-Dec-78 02:19 PM   8 BASIC2 14128  0/0
 RF:FIX:62   FO:IDX   USED:12:0       RECSI:62      CC:IMP

Total of 11 blocks in 1 file in SY:[1,235]

 Name .Ext    Size   Prot   Access     Date      Time   Clu  RTS    Pos  Op/rr
SY:[1,235]
SEQ2  .BAK       1L  < 60> 14-Mar-79 14-Mar-79 10:45 AM   8 RT11   21998  0/0

Total of 1 block in 1 file in SY:[1,235]

Ready
```

Figure UUP-4. DIRECT

BPCREF PROGRAM

- Reports cross reference of local/global symbols
- BPCREF command format:
  "OUTFILE.CRF=INFILE.BAC[,INFILE.BAS]"
  Note the following switches:

/SOURCE	Include source code file
/QUEUE	Queue file for printing
/NODELETE	Don't delete file after queueing
/NOHEAD	No header lines (for FILCOM)
/WIDTH:132	Line width for cref output
/PAGE:60	Number of lines per page
/NOCREF	No standard CREF printout
/GLOW:0	Global low limit
/GHIGH:32767	Global high limit
/LLOW:0	Local low limit
/LHIGH:32767	Local high limit

  The command "FILE" defaults to "FILE.CRF=FILE.BAC"
  The command "FILE/S" defaults to
  "FILE.CRF=FILE.BAC,FILE.BAS"

- Example

```
RUN $BPCREF
BPCREF  X7.0-53 RSTS V7.0-53 The ARK / 5.3
BPCREF command? EVLNFS/WIDTH:80/LL:500/LH:2000/GL:2000/GH:5000

Ready
```

Figure UUP-5. BPCREF

```
        Cross Reference Listing of EVLNFS on 22-Dec-78 at 01:12 PM        Page 1

#     #          19999
#  1005          1005   1070   1090
#  1020          1035   1037
#  1037          1035
#  1040          1037
#  1060          1050   1060   1060
#  1070          1060
#  1090          1032
# 10100          1020   1045   1060
# 10200          1030
# 10250          1045
# 10300          1060
# 19000          1000
# 32700          1050  19010
BLK.OFF$       102100 102600 103100
BLD.MASK$       10150
BYTE.OFF$      102100 102600 103100
CLD.MASK$       10150 10110
CNTR$           1060   10600  1070   10700
DCN$            1030   1030   1030   1030   10450  1045   1045   1045   10500
                1050   1050   10600  1060   1060   1060
DCN.UFD$        1045  102100
DUMMY$          10150  10460
DUMOFF$         10150  10460
DUMUFD$         10500
END.MASK$       10150 10110
ERR             19030
FILCNT$         1060   10600  1060   10600
HLD.DCN$        10450  1050
I$              10150  1015   1015   10460  1046   1046
IGNORE$         10100
LINK$           10150  10350  10370  10450  10600 10110
LINK%          101100 10110  101100 10110  10110  10110
LNR.ACC$        1045  102100
LNR.ATT$       102600
LNR.ENT$        10100  1015
LNR.NAM$        1035   1037  102100
MCLU$           1045  102600
MCOM$          102600
MCPU$          102600
MDEV$          102600
MFD.DCN$(10)    10150  1030   1045
MRCT$          102600
MS595          102600
MQCO$          102600
OLDDCN$         1030   10300  1045   10450  10500  1060   10600
PACKID1$        10100  1010
PACKID2$        10100  1010
PASS$          102100
PCS$            10100  1010
PG$             10050  1005   1037   1050   1090
PJ$             10050  1005   1037   1050   1090
PROG$           1037   1040  102100
PROJ$           1037   1040  102100
STAT$           10100
```

```
        Cross Reference Listing of EVLNFS on 22-Dec-78 at 01:12 PM        Page 2

UACNT$         102100
```

Figure UUP-6. BPCREF Printout (Page 1 of 3)


```
UACNT$          :0:0  :0:00
UFD.CCL$        10450  1045
UFD.DCN$(10)    10450  1045
UFD.ENQ$        10150  1060  103100
UFD.PG$         10150  1050
UFD.PJ$         10500  1050
UFD.UFD$        10500  1050
UL.BLD$          1032   1040  101100 10210  10260  10310
UL.CLO$          1030   1032   1045   1060   1060  121100
UL.ENQ$          1032   1060  101100 10210  10260  10310
UNAM1$           1060  103100
UNAM2$           1060  103100
UPROT$         102100
USTAT$           1035  102100
WHO.CARE$      103100
WORD$          102100
```

```
Please check that the following variables have been referenced properly:
BLK.OFF$        BLD.MASK$       BYTE.OFF$       DUMMY$          DUMOFF$
DUMUFD$         IGNORE$         LNR.ATT$        MCOM$           MCPU$
MDEV$           MFD.DCN$(10)    MRCT$           MMS59$          MQCO$
PASS$           STAT$           UACNT$          UFD.DCN$(10)    UPROT$
WHO.CARE$       WORD$
```

```
        Cross Reference Listing of EVLNFS on 22-Dec-78 at 01:12 PM        Page 3

                Global range is 2000 to 5000
```

```
        Cross Reference Listing of EVLNFS on 22-Dec-78 at 01:12 PM        Page 4

                Local range is 500 to 2000

#  1005          1005   1070   1090
#  1020          1035   1037
#  1037          1035
#  1040          1037
#  1060          1050   1060   1060
#  1070          1060
#  1090          1032
# 10100          1020   1045   1060
# 10200          1030
# 10250          1045
# 10300          1060
# 19000          1000
BLD.MASK%       10150
CNTR$           1060   10600  1070   10700
DCN%           10300  1030   1030   1030   10450  1045   1045   1045   10500
                1050   1050   10600  1060   1060   1060
DUMMY$          10150  10460
DUMOFF$         10150  10460
DUMUFD$         10500
FILCNT$         1060   10600  1060   10600
HLD.DCN$        10450  1050
I$              10150  1015   1015   10460  1046   1046
IGNORE$         10100
```

Figure UUP-6. BPCREF Printout (Page 2 of 3)

```
LNR.EMT$          10100  1015
MFD.DCM$(10)      10150  1030  1045
OLDDCM$           1030   10300 1045   10450  10500  1060   10600
PACKID1$          10100  1010
PACKID2$          10100  1010
PCS$              10100  1010
PG$               10050  1005  1037   1050   1090
PJ$               10050  1005  1037   1050   1090
$TAT$             10100
UFD.CLS$          10450  1045
UFD.DCM$(10)      10460  1060
UFD.PG$           10500  1050
UFD.PJ$           10540  1050
UFD.UFD$          10500  1050

Please check that the following variables have been referenced properly:
BLO.MASK$       DUMMY$          DUMOFF$         DUMUFD$         IGNORE$
MFD.DCM$(10)    STAT$           UFD.DCM$(10)

   37 Variables    170 References    29 Statements
  865 Name bytes   589 Code bytes
    7 Total time     5 CPU time

R-Words Reserved      Used      Free
   Data      2.00     1.02      1.01
   Code      1.17      .79       .38
  Total      4.00     2.60      1.40

Ready
```

Figure UUP-6. BPCREF Printout (Page 3 of 3)

COPY PROGRAM

- Prints warning for disks with bad blocks
- Reports device causing an error
- Does not copy bad blocks
- Does not copy RP06 disks

FLINT PROGRAM

- Executes faster
- Zeros floppies
- Supports RX02 for single density only
- Accepts a /NH switch to create an output file without a without a header record

FILCOM PROGRAM

- Accepts wildcards
- Has new switches
- Includes defaults in its prompts
- Example

```
RUN $FILCOM
FILCOM  X7.0-53 RSTS V7.0-53 The ARK / 5.3
File Comparison Program  19-Mar-79 01:29 PM
Output to <KB:>?
Input File #1? FIX.CMD
Input File #2? FIX.BAK
How Many to Match <3>?
BASIC+ Lines <NO>?
Blank Lines <NO>?

Comparing: 1) [1,235]FIX.CMD to 2) [1,235]FIX.BAK

********************************
1) [1,235]FIX.CMD
SY:SEQFIX=SY:SEQFIXA/MP
HISEG=BASIC2
**********
2) [1,235]FIX.BAK
SY:SEQFIX=SY:SEQFIX/MP
HISEG=BASIC2


?1 Difference Found.


Output to <KB:>? ^Z

Ready
```

Figure UUP-7. FILCOM Defaults


```
RUN $FILCOM
FILCOM  X7.0-05 RSTS V7.0-04 Timesharing
File Comparison Program  20-Dec-78 09:52 AM
Output to <KB:>? *.BAK,*.EXP/MA:1/NOBA/BL

Comparing: 1) [100,7]RMSTST.BAK to 2) [100,7]RMSTST.EXP


?Input File #2 '[100,7]RMSTST.EXP' not found


Comparing: 1) [100,7]IIO24A.BAK to 2) [100,7]IIO24A.EXP


?Input File #2 '[100,7]IIO24A.EXP' not found


Comparing: 1) [100,7]FILCOM.BAK to 2) [100,7]FILCOM.EXP

********************************
1) [100,7]FILCOM.BAK
BASIC-PLUS CONTINUATION LINES WILL BE CONSIDERED PAR OF A
NUMBERED PROGRAM LINE (I.E., /MA:3/NOBL/NOBA). HOWEVER, IT
**********
2) [100,7]FILCOM.EXP
BASIC-PLUS CONTINUATION LINES WILL NOT BE CONSIDERED PART OF A
NUMBERED PROGRAM LINE (I.E., /MA:3/NOBL/NOBA). HOWEVER, IT
********************************
1) [100,7]FILCOM.BAK
YOUSPECIFY THE /PATCH SWITCH, FILCOM DOES COMPARE BLANK LINES
AND DOES CONSIDER BASIC-PLUS CONTINUATION LINES. THAT IS,
A /P SPECIFICATION OVERRIDES THE NORMAL DEFAULTS AND SETS /BA
**********
2) [100,7]FILCOM.EXP
YOU SPECIFY THE /PATCH SWITCH, FILCOM DOES COMPARE BLANK LINES
AND DOES NOT CONSIDER BASIC-PLUS CONTINUATION LINES. THAT IS,
A /P SPECIFICATION OVERRIDES THE NORMAL DEFAULTS AND SETS /BA

?2 Differences Found.


2 Differences Found in 1 File of 1 Total Files Compared.

Output to <KB:>? ^Z

Ready
```

Figure UUP-8. FILCOM Switches

GRIPE PROGRAM

- Allows a CTRL Z as well as ESC to end program

QUOLST PROGRAM

- Shows more than 65535 blocks as >=65K

UMOUNT PROGRAM

- Allows mounting of an unlabelled ANSI tape.

RSTS/E V7.0 SWS UPDATE

CAC

Data Caching Internals

Data Caching Internals

CONTENTS

Introduction..... CAC-1

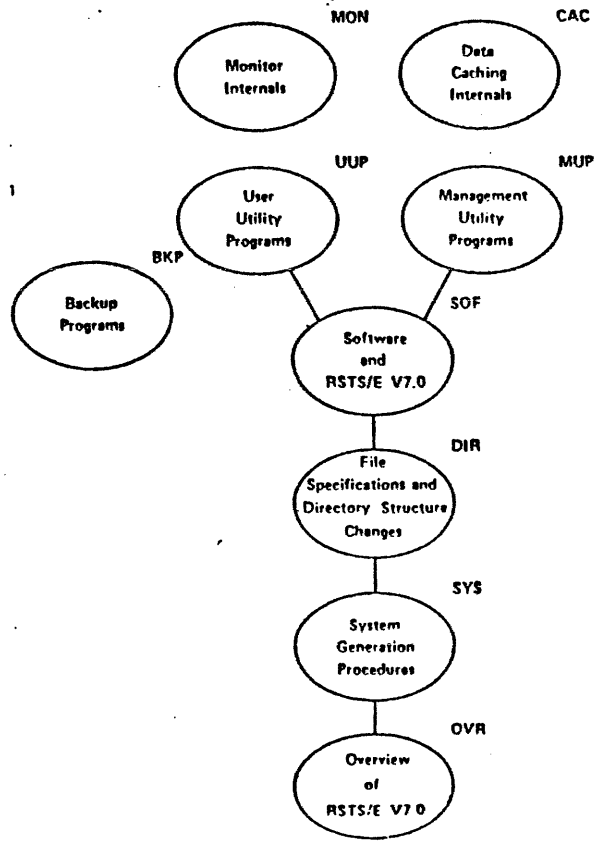
Monitor Structures - Without Sysgenning Data Caching..... CAC-3

Monitor Structures - Data Caching Sysgenned..... CAC-9

How Cache Views a Disk..... CAC-19

How Cache Moves Data into Memory..... CAC-22

COURSE MAP


MON
Monitor Internals
CAC
Data Caching Internals
UUP
User Utility Programs
MUP
Management Utility Programs
BKP
Backup Programs
SOF
Software and RSTS/E V7.0
DIR
File Specifications and Directory Structure Changes
SYS
System Generation Procedures
OVR
Overview of RSTS/E V7.0

Data Caching Internals

INTRODUCTION

The internal structures and operations of caching on RSTS/E aids in understanding what actions are taken when a user requests disk information.

Knowing how caching works enables you to determine:

- Whether a file should be cached
- How a file should be cached
- What the cache cluster size should be

This page is for notes.

MONITOR STRUCTURES - WITHOUT SYSGENNING DATA CACHING

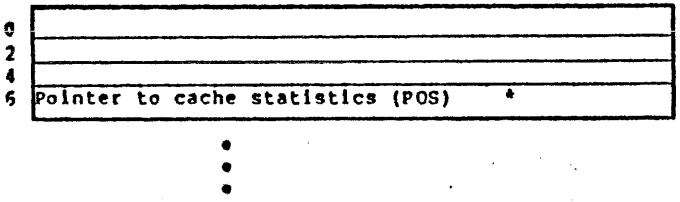
- Only directory caching (FIP buffering)
- Defaults at startup time
  - Cache enabled
  - Cache clustersize of 1
  - Use monitor buffer pool
  - Total number of cache clusters is 65535 (unlimited)
  - Total number of cache clusters for directories is 65535 (unlimited)

0	CHECTL	$$CACH::
2	CHETIC	
4	CHENXT	
6	CHEPRV	
10	CHEPAR*	
12	CHETAG*	
14	CHEUSM*	
16	(unused)	
20	CHELIM	
22	CHEFLM	
24	(unused)	
26	(unused) 0-no data caching	
30	(size of cache cluster in blocks)	
32	SAVKA6	
34	RSTRTN	
35	REFLAG	
40	CHEL3Q	
42		
44	CACHEQ	
46	POS **	
50	**	
52	**	
54	**	
56	HIT **	
60	**	
62	**	
64	**	
66	CHERST**	
70	CHENUM	
72	CHENUE	

* FILLED IN BY INIT
** ONLY IF STATS ARE GENNED

Figure CAC-1. CHECTL AREA OF MONITOR (no data caching)

CHECTL		400⟶cache enabled 0⟶cache disabled
CHETIC		1⟶charge cache time 0⟶no charge
CHENXT		Pointer to "first" cache tag in list @+2
CHEPRV		Pointer to "last" cache tag in list @ + 0
CHEPAR		PAR6 value for extended cache tags (INIT)
CHETAG		List of usable extended cache tage (INIT)
CHEUSM		-1⟶monitor pool (INIT) 0⟶no use of monitor pool
CHELIM		Limit of total numbers of extended cache tags
CHEFLM		Limit of number of extended cache directory tags
SAVKA6		Saved KISAR6
RSTRTN		"Restart" return address
REFLAG		"Restart" flag (SP stack value)
CHEL3Q	(word)	Pointer to cache completion queue root (CACHEQ)
	(word)	Caching level 3 queue bit (QCACHE)
CACHEQ		Caching completion queue root
POS	(2 words)	Total possible number of hits
	(2 words)	Total possible number of FIP hits
HIT	(2 words)	Total number of hits
	(2 words)	Total number of FIP hits
CHERST		Number of times cache restarted
CHENUM		Number of monitor buffer cache elements
CHENUE		Number of extended pool cache elements

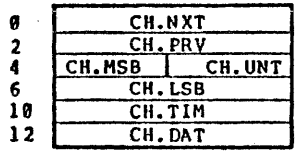


* Only if STATS are genned

Figure CAC-2. STSTBL entry for statistics (no data caching)

Cache tags

- Identify directory blocks currently in the cache clusters.
- Keep track of time of last reference.
- A doubly linked list with head and tail pointers in CHECTL.
- Exist in XBUF and/or monitor pool space.
- XBUF tags created by INIT at start up.



CH$SIZ - Size of cache tag in bytes

Figure CAC-3. Cache Tag

CH.NXT	Pointer to next @ CH.PRV	If two high bits set, tag in XBUF
CH.PRV	Pointer to previous @ CH.NXT	
CH.UNT	Unit number of block	
CH.MSB	Block's FBN MSB	
CH.LSB	Block's FBN LSB	
CH.TIM	Time of last reference	
CH.DAT	Pointer of cached data block	

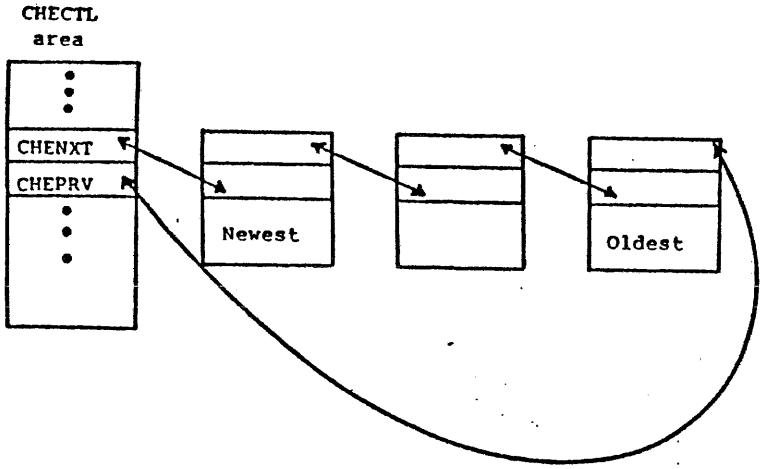


Figure CAC-4. Cache Tag List

MONITOR STRUCTURES - DATA CACHING SYSGENNED

- Directory and Data Caching available
- Defaults at startup time
  - Cache enabled
  - Data caching available through MODE
  - Cache cluster size of 4
  - Total number of cache clusters to be used is 65535 (unlimited)
  - Total number of cache clusters for directories is 65535 (unlimited)
  - Total number of cache clusters for data is 65535 (unlimited)
- All cache tags are in XBUF

0	CHECTL	
2	CHETIC	
4	CHENXT	
6	CHEPRV	
10	CHEPAR*	$$CACH==
12	CHETAG*	
14	0 - No monitor buffers used*	
16	CHEWTG*	
20	CHELIM	
22	CHEFLM	
24	CHEDLM	
26	200-Data caching | CHEMOD	
30	CHEBSZ*	$$CBSZ
32	CHESHF*	
34	*	
36	CHEBIC*	
40	SAVKA6	
42	RSTRTN	
44	REFLAG	
46	TMWQUE	
50	CHLFXB	
52	CHWTL3	
54		
56	CHWRTQ	
60	CHRDL3	
62		
64	CHREDQ	
66	FAKDSQ	
70		
72		
74		
76		
100		
102		
104		
106		
110		
112		
114		
116		
120		

* Filled in by INIT

Figure CAC-5. CHECTL Area of Monitor (data caching)

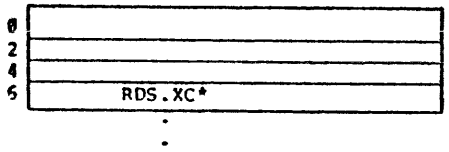
122	
124	
126	RDS.XC**
130	
132	DIR.CN**
134	
136	HRS.RH**
140	
142	DIR.RH**
144	
146	CHERST**
150	
152	CHENUE
154	CHECNT
156	CHDCNT
160	DAT.CN**
162	
164	DAT.RH**
166	
170	RDS.NC**
172	
174	RDS.CI**
176	
200	RDS.LX**
202	
204	RDS.RX**
206	
210	RDS.SR**
212	
214	WTS.NX**
216	
220	WTS.CH**
222	
224	CAN.CH**
226	
230	HIT.CH**
232	

** Only if STATS Genned

Figure CAC-5. CHECTL Area of Monitor (data caching) (cont.)

CHECTL		400→ cache enabled 0→ cache disabled
CHETIC		1→ charge cache time 0→ no charge
CHENXT		Pointer to first in list @+2
CHEPRV		Pointer to last in list @+0
CHEPAR		PAR6 value for extended cache tags
CHETAG		List of usable extended cache tags
CHEWTG		List of usable extended cache tags
CHELIM		Limit of total number of extended cache tags
CHEFLM		Limit of number extended cache directory tags
CHEDLM		Limit of number of extended cache data tags
CHEMOD		0→ No data caching 1→ open mode caching -1→ cache all data
CHEBSZ		Size of a cache cluster in blocks
CHESHF	(1 word)	Amount of shift to divide by cluster size
	(1 word)	Mask to round FBN down to cache cluster
CHEBIC		Complement of above mask
SAVKA6		Saved KISAR6
RSTRTN		"Restart" return address
REFLAG		"Restart" flag (SP stack value)
TMWQUE		Queue for writes awaiting free write tag
CHLFXB		Time of last failure to allocate buffer
CHWTL3	(1 word)	Pointer to caching queue root for writes (CHWRTQ)
	(1 word)	Level 3 queue bit for cache (QCACHE)
CHWRTQ		Caching queue root for writes
CHRDL3	(1 word)	Pointer to caching queue root for reads (CHREDQ)
	(1 word)	Level 3 queue bit for cache (QCACHE)
CHREDQ		Caching queue root for reads
FAKDSQ		Fake DSQ for handling cross cluster transfers
RDS.XC	(2 words)	Number of disk transfers which are cached
DIR.CN	(2 words)	Number of above which are directory transfers
HRS.RH	(2 words)	Number of cache hits
DIR.RH	(2 words)	Number of above which are directory hits
CHERST		Number of times cacher restarted
CHENUE		Number of extended pool cache elements

CHFCNT		Count of directory cache tags
CHDCNT		Count of data cache tags
DAT.CN	(2 words)	Data transfers which were cached
DAT.RH	(2 words)	Cache hits on data transfers
RDS.NC	(2 words)	Transfers which are not cached
RDS.CI	(2 words)	Fraction of above due to cache installs
RDS.LX	(2 words)	Large transfers cut up into segments
RDS.RX	(2 words)	Large transfers not cached because no tags
RDS.SR	(2 words)	Segment reads produced when cut up
WTS.NX	(2 words)	Total write transfers
WTS.CH	(2 words)	Tags hit by writes
CAN.CH	(1 word)	Pointer to STATS area for directory reads (DIR.CN)
	(1 word)	Pointer to STATS area for data reads (DAT.CN)
HIT.CH	(1 word)	Pointer to STATS area for directory hits (DIR.RH)
	(1 word)	Pointer to STATS area for data hits (DAT.RH)



* Only if STATS are genned

Figure CAC-6. STSTBL entry for statistics (data caching)

Cache Tags

- Identify directory and data blocks currently in cache clusters
- Keep track of time of last reference
- A doubly linked list with head and tail pointers in CHECTL
- Exist only in XBUF
- Created by INIT at startup

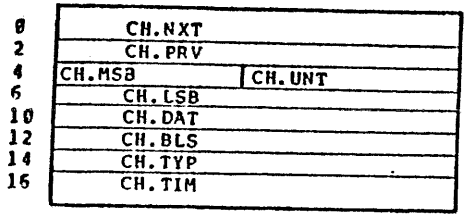


Figure CAC-7. Cache Tag

CH.NXT Pointer to next @ CH.PRV

CH.PRV Pointer to previous @ CH.NXT

CH.UNT Unit number of block

CH.MSB Block's FBN MSB

CH.LSB Block's FBN LSB

CH.DAT Pointer to the cached data block

CH.BLS Pointer to DSQ's waiting on tag

CH.TYP Type pointer (data or directory)

CH.TIM Time of last reference

Write Tags

- Used to store DSQ information while disk is transferring data to the cache

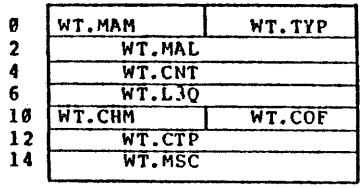


Figure CAC-8. Cache Write Tag

WT.TYP Write tag type (real or from tag list)
WT.MAM Saved DSQMAM
WT.MAL Saved DSQMAL
WT.CNT Saved DSQCNT
WT.L3Q Saved DSQL3Q
WT.COF Saved transfer's block offset into cluster
WT.CHM Saved cache mode of transfer
WT.CTP Saved pointer to cache tag
WT.MSC Saved DSQMSC

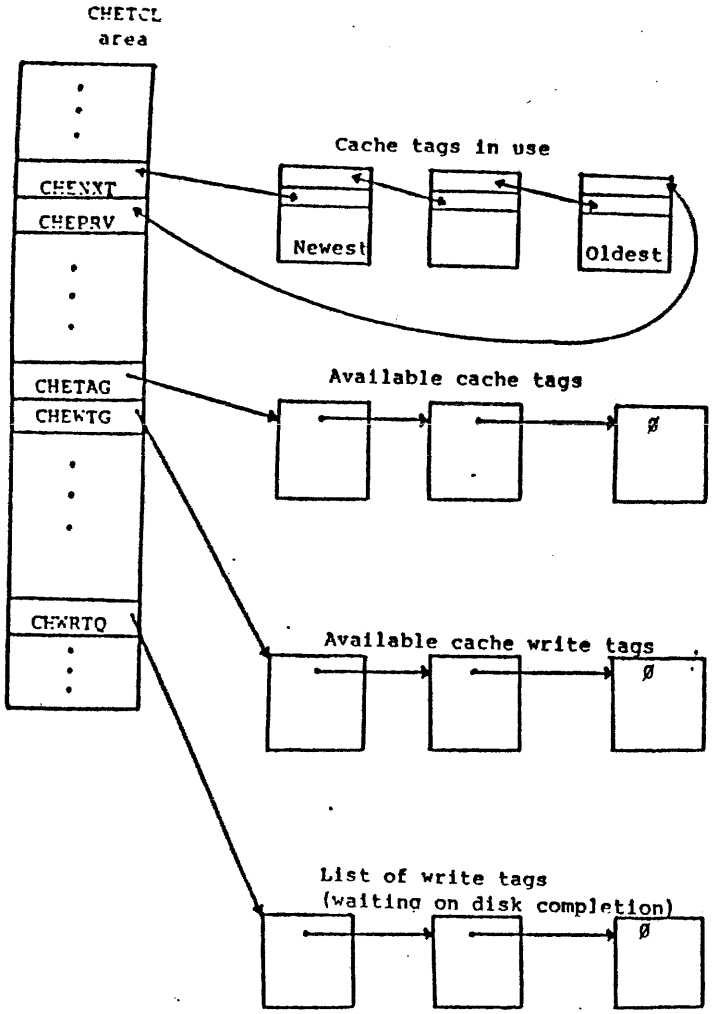
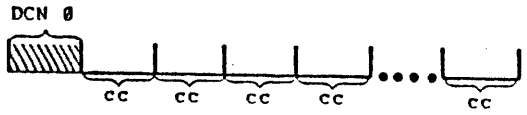


Figure CAC-9. Tag List (data caching)

HOW CACHE VIEWS A DISK

- Always divided into cache clusters (cc) starting at DCN 1.

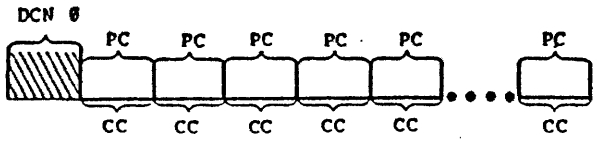


Note

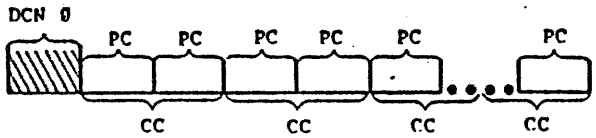
Cache will always transfer an entire cache cluster.

If there is no room in the cache a normal disk I/O to user buffer is done.

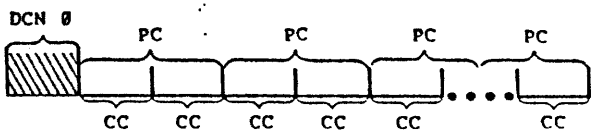
- When a cache cluster equals a pack cluster (pc)



- When a cache cluster is larger than a pack cluster



- When a cache cluster is smaller than pack cluster

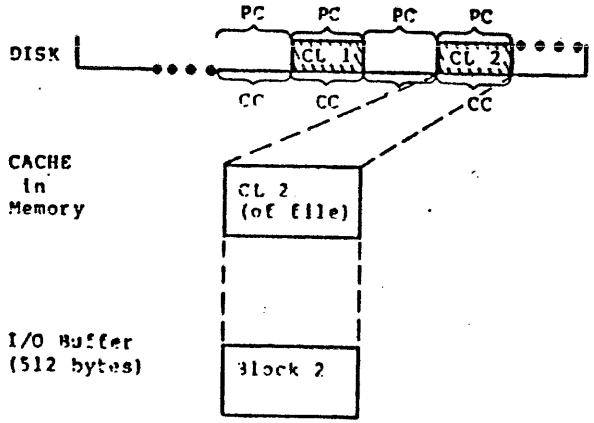


HOW CACHE MOVES DATA INTO MEMORY

Case 1: • Cache Cluster = Pack Cluster = File Cluster = 1

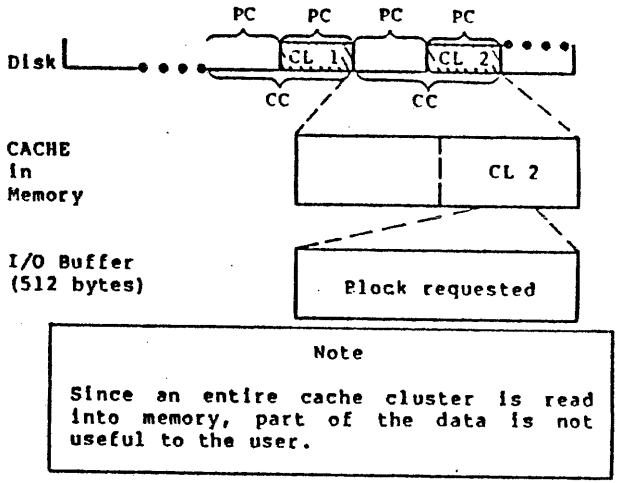
Note
No Sequential Caching

• User Request - Read block 2 of a file



Case 2: • Cache cluster > Pack Cluster = file cluster

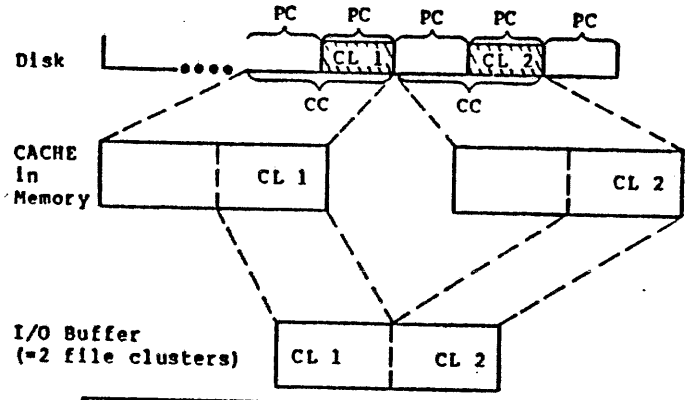
• User Request - Read Block 2 of a file in file cluster 2



Note
Since an entire cache cluster is read into memory, part of the data is not useful to the user.

Case 3: • Cache Cluster > Pack Cluster = File Cluster

• User Request - Read File Clusters 1 and 2



Note

Cache makes two disk reads to satisfy the request.

RSTS/E V7.0 SWS UPDATE

DIR

File Specification and Directory Structure Changes

File Specification and Directory Structure Changes

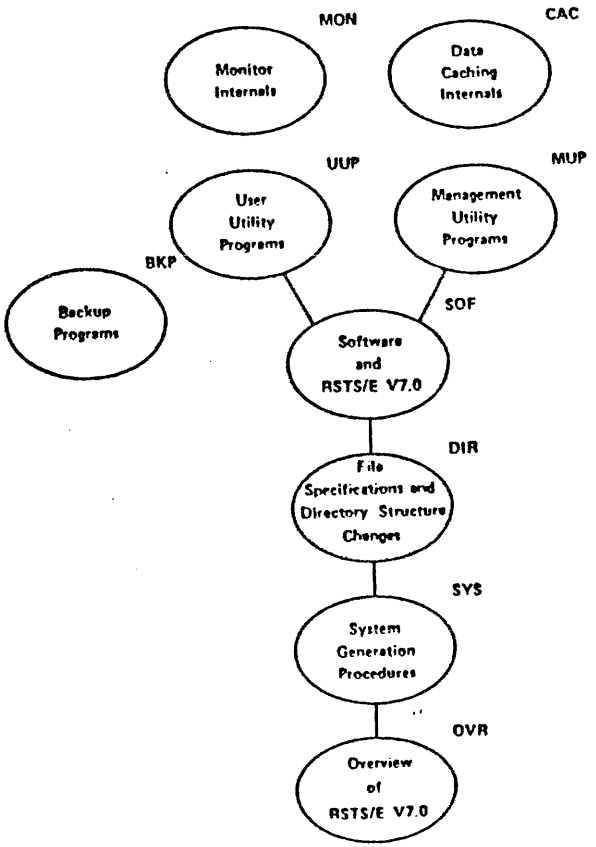
CONTENTS

Introduction..... DIR-1

File Specification Changes..... DIR-3
Device Designators..... DIR-3
Protection Codes..... DIR-4
File Specification Switches..... DIR-4

Directory Structure Changes..... DIR-6
MFD Changes..... DIR-6
UFD Changes..... DIR-8

COURSE MAP


MON
Monitor Internals
CAC
Data Caching Internals
UUP
User Utility Programs
MUP
Management Utility Programs
BKP
Backup Programs
SOF
Software and RSTS/E V7.0
DIR
File Specifications and Directory Structure Changes
SYS
System Generation Procedures
OVR
Overview of RSTS/E V7.0

File Specification and Directory Structure Changes

INTRODUCTION

The RSTS/E operating system supports a variety of high level languages: BASIC-PLUS, BASIC-PLUS-2, COBOL, DIBOL, RPGII, etc. Regardless of the language in which you code, the file specification is the way of telling the RSTS/E monitor exactly what actions you want taken when a file or device is opened. It is through the file specification that you can implement many of the new features of RSTS/E V7.0.

Furthermore, some of these new enhancements have required changes to or modifications of the current disk directory structure.

In this module, you will learn not only how to specify these new features using the RSTS/E file specification, but also the changes to the disk directory structure needed by RSTS/E to maintain your files.

This page is for notes.

File Specification and Directory Structure Changes

FILE SPECIFICATION CHANGES

Device Designators

Table DIR-1. Device Designators

Device Name	Device Designation	Valid Unit Numbers
RL02	DL	0-3
RX02	DX	0-7
TS04	MS	0-7
TU58	DD	0-7
KMC11	XK	0-15

The underscore character can precede the device designator. For example:

"_DL2:file.ext"

- Suppress logical device translation during file name string scan.

Protection Codes

Protection code - 128

- Settable by privileged users only.
- Write null characters over a compiled file or data file when implicitly or explicitly deleted by:
  - KILL a file
  - OPEN FOR OUTPUT
  - ZEROing an account

> **Note**
>
> On a large file this can be a time-consuming process.

File Specification Switches

1. /FILESIZE:n or /SIZE:n
   - The only way to pre-extend RSTS/E V7.0 large files without explicitly writing actual or dummy records. For example:

     "FILE.EXT/FILESIZE:200000"

     Allocate 200,000 blocks for the file
2. /POSITION:n
   - Place the first block of the file at or near device cluster n on the disk. For example:

     "FILE.EXT/POSITION:25000"

     Place the beginning of the file at or near device cluster 25000

Mode Value	Disk	Paper Tape Reader	Paper Tape Punch
2		Even parity checking	Punch even parity
2 + 1		Odd parity checking	Punch odd parity
32	Create a tentative file		
64	Create conditionally contiguous file		
128	File structured-No supersede		
	Non-file structured-NFS Block mode		
256	Random data cache the file		
256 + 2048	Sequential data cache the file.		
16384	Write access to a UFD if: 1. No one is logged into the account 2. The UFD is not open 3. No files in the UFD are opened 4. The account is not [0,1],[1,1] or [1,2]		

Figure DIR-1. /MODE:n

DIRECTORY STRUCTURE CHANGES

MFD Changes

1. MFD Label Entry (V6C change)

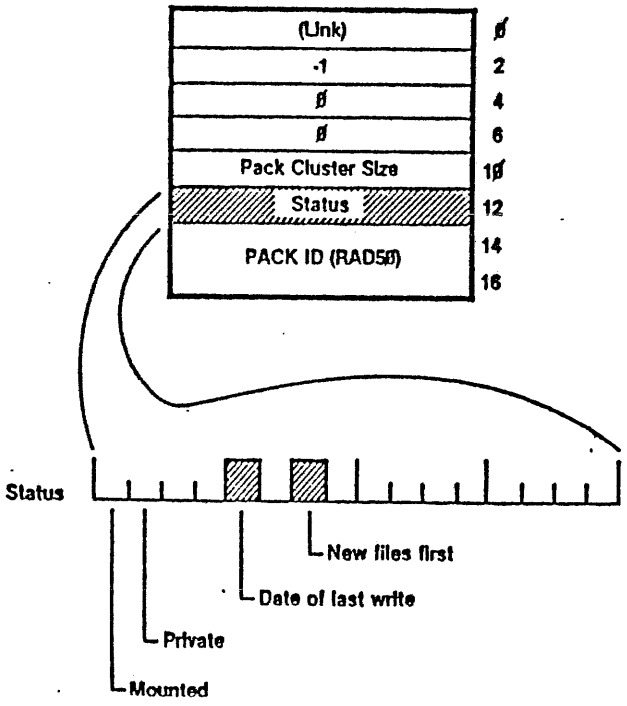


Figure DIR-2. MFD Label Entry (V6C change)

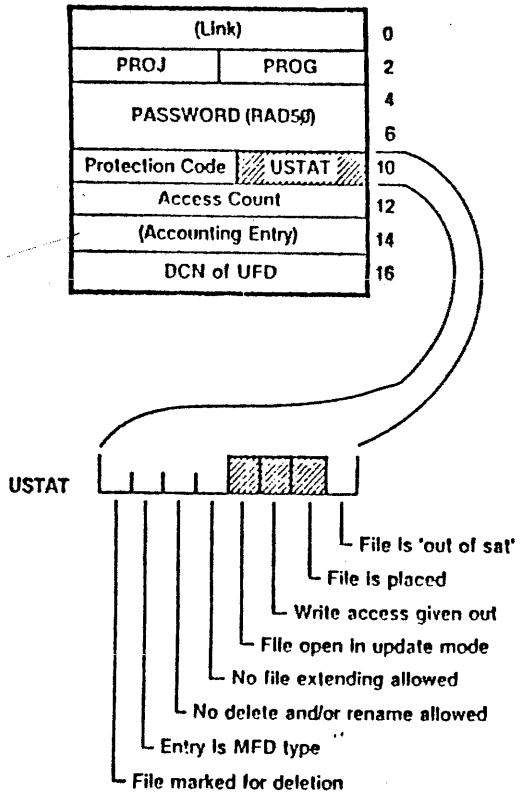


Figure DIR-3. MFD Name Entry

UFD Changes

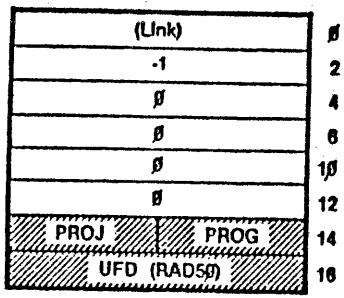


Figure DIR-4. UFD Label Entry (V6C change)

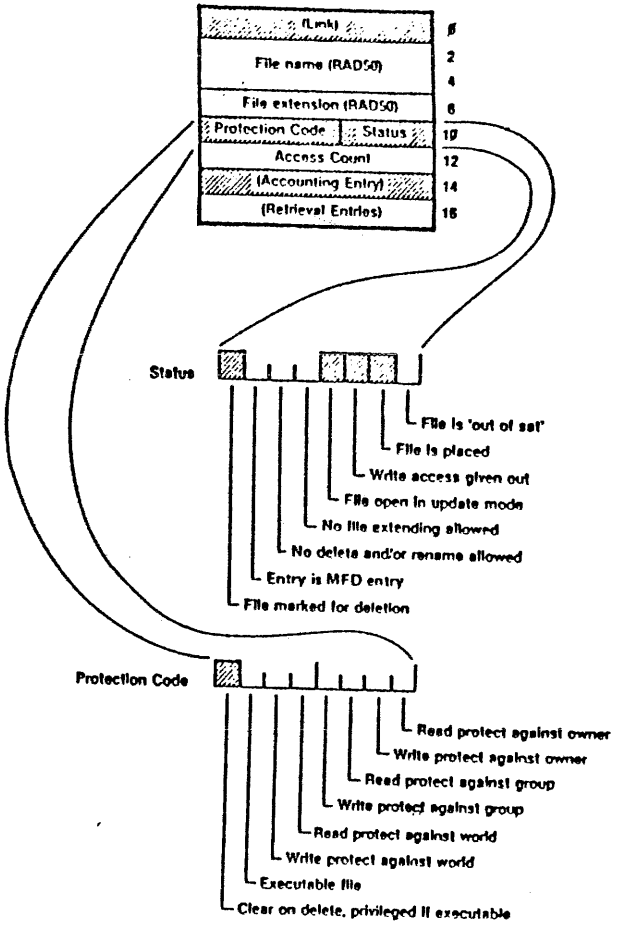


Figure DIR-5. UFD Name Entry

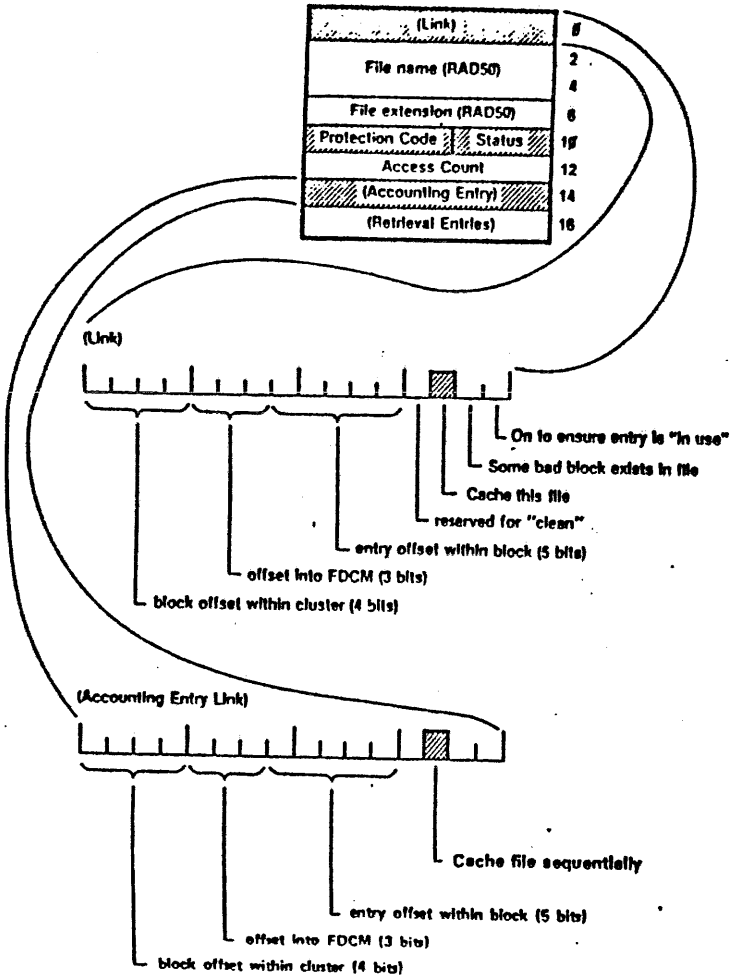


Figure DIR-6. UFD Name Entry (Cont.)

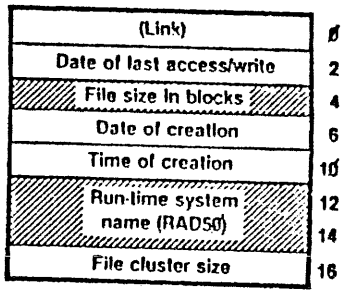


Figure DIR-7. UFD Accounting Entry on Small File Systems

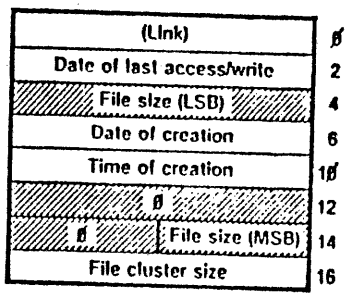


Figure DIR-8. UFD Accounting Entry on Large File Systems

Note

An executable file has run-time system name at offsets 12 and 14 on large file systems.

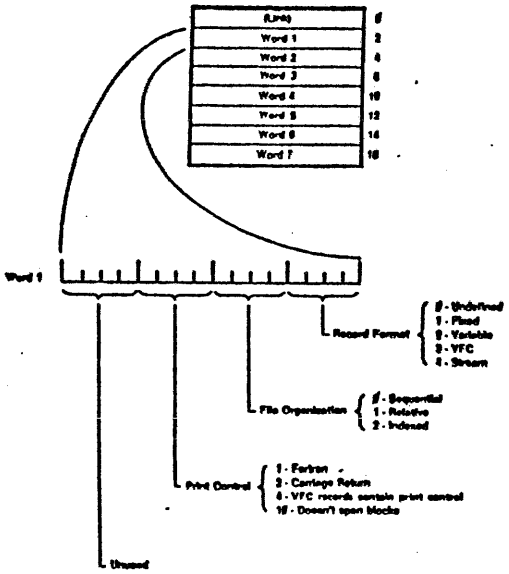


Figure DIR-9. UFD Attributes Entry #1 (V6C change)

Word 2 Record Size (Actual size for fixed length records or maximum size for variable length records)

Word 3 Highest virtual block number (MSB)

Word 4 Highest virtual block number (LSB) (corresponds to the file size accounting entry)

Word 5 EOF block number (MSB)

Word 6 EOF block number (LSB) (block that is the logical end of file)

Word 7 Offset to first usable byte in EOF block.

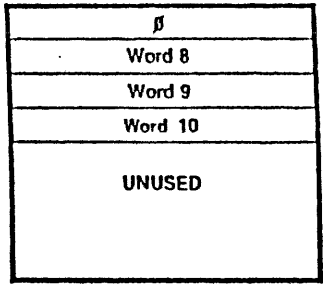


Figure DIR-1Ø. UFD Attributes Entry #2 (V6C change)

Word 8 Number of bytes in fixed control area (high byte) Bucketsize in blocks (low byte)

Word 9 Maximum length of record actually read by RMS

Word 1Ø Default extension quantity

This page is for notes.