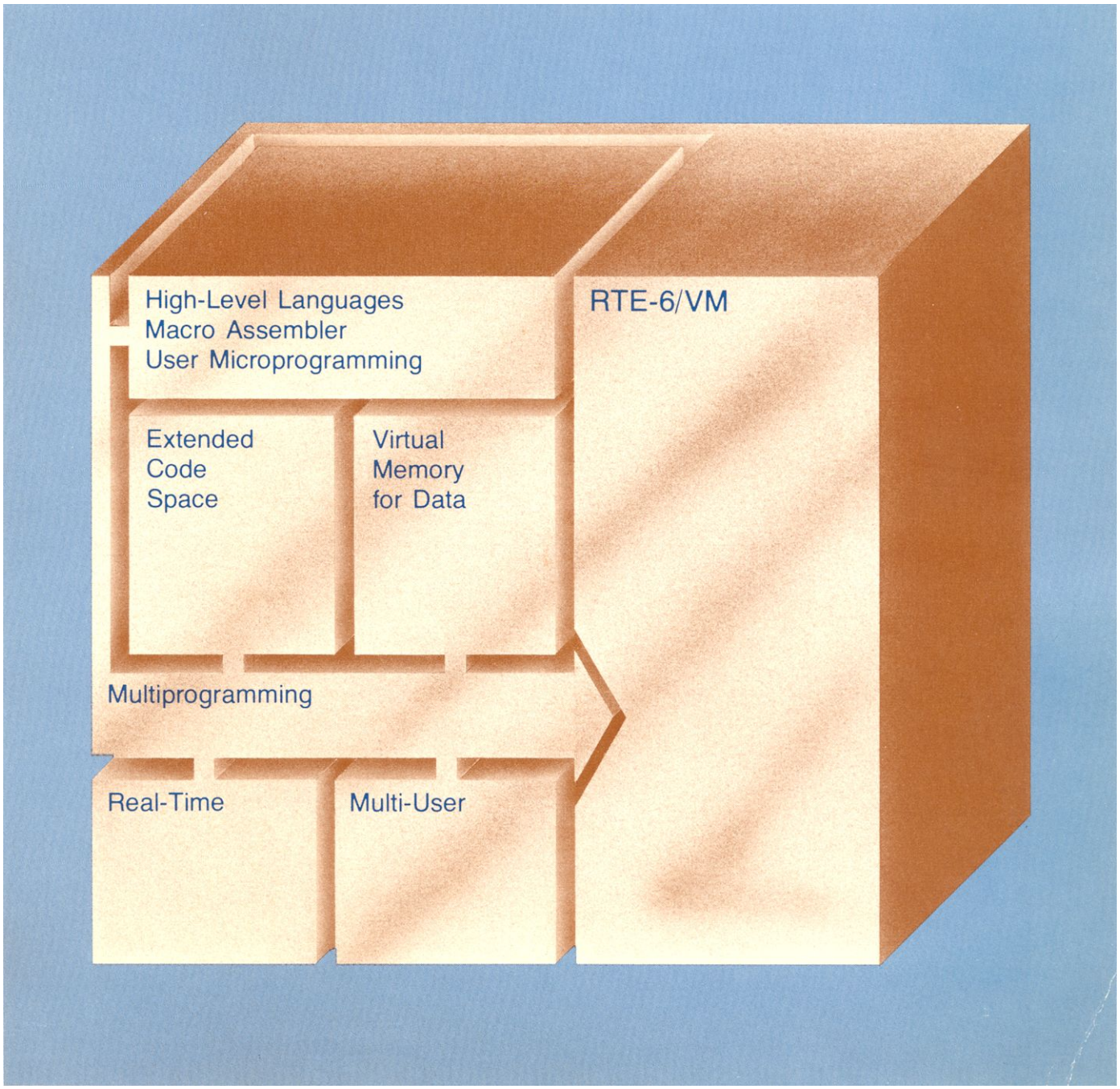
HEWLETT PACKARD

RTE-6/VM Utility Programs

Reference Manual



RTE-6/VM
Utility Programs

Reference Manual

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

MANUAL PART NO. 92084-90007
Printed in U.S.A. December 1981
U0283

PRINTING HISTORY

The Printing History below identifies the Edition of this Manual and any Updates that are included. Periodically, Update packages are distributed which contain replacement pages to be merged into the manual, including an updated copy of this Printing History page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past Updates, however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all Updates.

To determine what software manual edition and update is compatible with your current software revision code, refer to the appropriate Software Numbering Catalog, Software Product Catalog, or Diagnostic Configurator Manual.

First Edition	Dec 1981	
Update 1	Jul 1982	
Update 2	Jan 1983	Add LUPRN, enhancements to SCOM and Physical Backup Utilities.
Update 3	Feb 1983	Enhancements to FORMC

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1983 by HEWLETT-PACKARD COMPANY

Preface

This manual describes the format and use of the interactive utility programs available for use with the Hewlett-Packard RTE-6/VM Operating System. As new utilities are added to the system, they will be documented in this manual. Before attempting to use the utilities, you should be familiar with the RTE-6/VM operating system. The manuals associated with the operating system are defined in the Index to Operating System Manuals (92084-90001).

This manual is organized in nine chapters to facilitate references to the information:

Chapter 1 provides a general introduction to the operating system utilities and describes the syntax used in the utility runstrings.

Chapter 2 presents the general system utilities that allow you to call for the system status, to manipulate files, to call for help in defining error messages, to compile and load programs and to program the terminal function-key cluster.

Chapter 3 describes the utility provided to translate old-record formats to the formats used by RTE-6/VM.

Chapter 4 describes the utilities used to copy and restore files between tape and disc media.

Chapter 5 describes the utilities used to back up cartridges to tape and to restore the saved cartridges.

Chapter 6 defines the on-line and off-line backup utilities that allow you to selectively back up files, groups of files, or entire logical units (LU).

Chapter 7 describes the utility provided to translate files to and from the Logical Interchange Format (LIF) that is interchangeable with other HP systems.

Chapter 8 contains the disc formatting utilities for the HP MAC/ICD and CS80 discs, and the HP 9895 Flexible Disc.

Chapter 9 defines the on-line driver replacement utilities that allow you to replace device drivers without having to regenerate the system.

Table of Contents

Chapter 1 Introduction

Introduction 1-1
Operating Environment 1-1
Command Syntax 1-2

Chapter 2 General System Utilities

Introduction 2-1
Status Utility (WHZAT) 2-1
Program States 2-2
WHZAT Output, AL/SM Options 2-5
WHZAT Output, PA Option 2-8
WHZAT Output, PL Option 2-10
Log Track-Assignment Table Utility (LGTAT) 2-14
Abbreviated LGTAT Output 2-14
Complete LGTAT Output 2-15
System Configuration Utility LUPRN 2-16A
HELP Utilities (HELP, CMD, GENIX) 2-17
HELP 2-17
Calling HELP 2-17
Examples 2-18
Error Messages 2-20
CMD 2-20
Calling CMD 2-21
Examples 2-23
Error Messages 2-25
GENIX 2-26
Calling GENIX 2-26
Input File Format 2-27
Examples 2-29
Error Messages 2-30
Compile Utility (COMPL) 2-31
Calling COMPL 2-31
Examples 2-33
Compile and Load Utility (CLOAD) 2-34
Calling CLOAD 2-34
Examples 2-36
File Manipulation Utilities (MERGE, SCOM) 2-37
MERGE 2-37
SCOM 2-39
Scheduling SCOM Programmatically 2-41
Status Interrogation 2-43
Examples 2-43
Error Messages 2-51
Function-Key Manipulation Utilities (KEYS, KYDMP) 2-52

KEYS 2-52
Calling KEYS 2-53
Modifying an Existing Command-Set File 2-56
Error Messages 2-56
KYDMP 2-57
Calling KYDMP 2-57
Error Messages 2-59

Chapter 3 Record Reconfiguration Utility

Introduction 3-1
Running OLDRE 3-2
Translation Results 3-3
Program Restrictions 3-4
MACRO/1000 3-4
Pascal 3-5
FORTRAN 3-5

Chapter 4 File Copy Utility

Introduction 4-1
Running FC 4-1
FC Commands 4-2
Information Commands 4-4
Command Summary Function 4-4
FMGR Error Help Function 4-5
Configuration Commands 4-5
List Device: LL 4-5
Title Command: TI 4-6
Name Comment File: CF 4-6
Scratch Area Definition: SC 4-7
Echo Command: EC 4-7
Copy and Related Commands 4-7
Source, Destination Parameter Considerations 4-10
COpy Command Options 4-11
Default Command 4-15
CO Command Examples 4-16
Grouping CO Commands 4-18
Listing Commands 4-19
List Directory: DL 4-19
List Cartridges: CL and CLAL 4-22
List Comment/Header Files: LC,LH 4-22
Transfer Command 4-23
Tape Handling 4-23
Destination Disc Handling 4-25
Loading FC 4-26
Performance Considerations 4-26
Error Handling in Transfer Files 4-28
Messages Logged by FC 4-32

Errors Requiring Operator Action/Response 4-32
Information Messages and Warnings 4-33
Disc Data I/O Errors 4-37
Non-Fatal Tape Read Errors 4-37
Errors Affecting Single File 4-38
Loss of Unidentified Files on Copy from Tape 4-42
Disc-to-Tape Copy Verify Errors 4-43
Errors that Cause Rejection of Current Source Tape Volume 4-43
Command Syntax, Parameter Errors (Command is Skipped) 4-44
Command Out-of-Sequence Errors (Command is Skipped) . 4-47
Other Errors that Terminate Current Command 4-47
Other Errors 4-50
Errors that Terminate FC 4-50

Chapter 5 Cartridge Backup Utilities

Introduction 5-1
Write Tape Utility (WRITT) 5-1
Calling WRITT 5-1
Examples 5-3
Error Messages 5-4
Read Tape Utility (READT) 5-6
Calling READT 5-6
Examples 5-9
Error Messages 5-11

Chapter 6 On-Line, Off-Line Physical Backup Utilities

Introduction 6-1
Compatibility With Other Disc Backup Utilities 6-1
Compatibility Among Disc Drives 6-2
Tape Handling 6-4
Tape Positioning 6-5
Unit Save Tape Files 6-6
Multiple-Volume Tape Sets 6-6
Data Verification 6-6
Verification of Saves 6-7
Verification of Restores 6-7
Pushbutton Save/Restore Data Verification 6-7
Pusbutton (PB) Operations 6-8
Using the BReak Command 6-8B
On-Line Physical Backup Utilities 6-9
Loading the On-Line Utilities 6-10
PSAVE 6-11
Calling PSAVE Interactively 6-12
Examples 6-14
PRSTR 6-17
Calling PRSTR Interactively 6-19
Example 6-23

PCOPY 6-26
Calling PCOPY Interactively 6-27
Example 6-28
Off-Line Physical Backup Utility 6-29
Referencing Devices Off-Line 6-30
Loading the Off-Line Utility 6-30
Loading the Off-Line Utility from Cartridge Tape. . 6-31
Loading the Off-Line Utility from Magnetic Tape . . 6-32
Loading and Using the PBU I/O Reconfigurator . . . 6-34
Loading Utilities from Cartridge Tape 6-37
Loading Utilities from Magnetic Tape 6-38
Off-Line System Console Operations 6-38
Display I/O Configuration (IO) 6-38
Transfer to Input LU (TR) 6-39
Restore Tape File (RE) 6-40
Calling RE Interactively 6-43
Example 6-44
Disc to Disc Copy (CO) 6-46
Save Disc to Tape (SA) 6-49
Calling SA Interactively 6-51
Example 6-52
Tape Movement Functions 6-53
Help Function 6-54
Data Structures 6-55
PSAVE Format on Magnetic Tape (Reel-to-Reel) 6-55
PSAVE Format on CS80 Cartridge Tape 6-57
Pushbutton Image Format on CS80 CTD 6-59
Error Messages 6-61

Chapter 7 Logical Interchange Format Utility

Introduction 7-1
Utility Commands 7-2
Naming Conventions 7-3
Calling LIF Interactively 7-4
HE - Help 7-4
CO - Copy 7-5
DL - Directory List 7-6
IN - Initialize 7-7
LI - List 7-8
LL - Logical List 7-8
MC - Mount Cartridge 7-8
PK - Pack Cartridge 7-9
PU - Purge 7-9
RN - Rename 7-9
ST - Store 7-10
SV - Severity 7-11
TR - Transfer control 7-12
Exit Command 7-12
Errors 7-12

Chapter 8 Disc Formatting Utilities

INTRODUCTION 8-1
FORMT 8-1
Loading FORMT 8-2
Calling FORMT 8-2
Format A Flexible Disc (FO) 8-4
Using FO 8-5
Example 8-6
Initialize an LU (IN) 8-7
Using IN 8-8
Examples 8-9
Spare a Track (SP) 8-10
Using SP 8-11
Examples 8-12
Reformat a Disc (RE) 8-14
Verify a Disc (VE) 8-17
FORMT Utility Error Messages 8-18
FORMC 8-20
FORMC Options 8-21
Calling FORMC 8-22
Response to System BReak Command 8-23
Runstring Mode 8-23
Interactive Mode 8-23
Verify Command (VE) 8-24
Format Command (FO) 8-27
Tape Formatting. 8-29
Disc Formatting 8-31
Spare Command (SP) 8-33
Error Messages 8-36

Chapter 9 On-Line Driver Replacement Utilities

Introduction 9-1
Finding Space For Drivers 9-1
Overlaying an Existing Driver 9-1
Using Available Pages 9-2
Creating a Dummy Driver 9-2
Base Page Links 9-2
Loading the Driver Replacement Utilities 9-3
Driver Relocation Utility (DRREL) 9-4
Example 9-8
Error Messages 9-9
Driver Replacement Utility (DRRPL) 9-11
Replacement Driver Specification 9-15
Driver Replacement 9-21
Driver Replacement Example 9-22
Error Messages 9-25

Appendix A Initializing, Formatting, and Sparing Discs

Initializing and Sparing a Hard Disc. A-3
Formatting a Flexible Disc. A-4
Interleaving (Fill Number) A-4

List of Illustrations

Figure 2-1. Program Status Mode (AL) Output. 2-6
Figure 2-2. Program Status Mode (SM) Output. 2-7
Figure 2-3. Partition Status Mode (PA) Output. 2-9
Figure 2-4. Active Program Mode (PL) Output, Type 5 Programs Only. 2-12
Figure 2-5. Active Program Mode (PL) Output, Background Programs Only. 2-13
Figure 2-6. Active Program Mode (PL) Output, #----- Programs Only. 2-13
Figure 2-7. Complete LGTAT Output. 2-16
Figure 2-8. Soft Key Label Display Format. 2-58
Figure 4-1. Tape Directory List Format 4-21
Figure 9-1. DRREL Command Summary List Format. 9-7
Figure 9-2. DRRPL Command Summary List Format. 9-13
Figure 9-3. DRRPL DI/MI Command Listing Format 9-13
Figure 9-4. DRRPL ME/DE Command Listing Format 9-14
Figure 9-5. DRRPL MD/DD Command Listing Format 9-15
Figure A-1. Formatting Time vs. Fill Value A-6

Tables

Table 2-1. General Wait State Messages 2-4
Table 2-2. Track Assignment Table Entries. 2-15
Table 4-1. Summary of FC Commands. 4-3
Table 6-1. Disc Drive Source/Destination Compatibility . 6-3
Table 9-1. Driver Replacement Prompt Sequence. 9-16

Chapter 1
Introduction

Introduction

This Utility Programs Reference Manual details the format and use of the interactive programs available for use with the Hewlett-Packard RTE-6/VM Operating System. Utilities are provided to display system status, to manipulate files, save and restore files and LUs between discs and magnetic tape devices, to format discs, to replace drivers on-line, and to physically back-up discs off-line. If special loading considerations exist, these are included in the utility descriptions. Where applicable, examples are provided to illustrate the use of the utilities, and error messages are given as part of each free-standing utility description.

Operating Environment

The utilities defined in this manual operate in the following environment:

HP 1000 M/E/F Systems, Model 60 and 65

RTE-6/VM Operating System

One or more of the utilities support each of the following discs. Note that the referenced device drivers must be generated into the system.

- ICD (Integrated Controller Disc) - HP Models 9895 (flexible disc), 7906H, 7920H and 7925H with driver DVA32.
- MAC (Multiple Access Controller Disc) - HP Models 7905, 7906, 7920 and 7925 with driver DVR32.
- CS80 Discs - HP Models 7908, 7911, 7912 and 7935 with drivers DVM33/DVN33.

- HP Model 7900 Disc with driver DVR31.
- HP Model 9895 Flexible Disc with driver DVR33.

The utilities can be called both under Session Monitor and in a non-session environment. In a non-session environment, you operate under control of the Multi-Terminal Monitor (MTM). Refer to the RTE-6/VM Terminal User's Reference Manual (92084-90004), for a discussion of the Session Monitor and the MTM.

Command Syntax

Within this manual, the following command syntax conventions are used. When entering parameters in the runstring, do not include blank spaces between parameters and between the utility call and the first parameter. While blanks are stripped out of the runstring under Session Monitor control, they are significant in a non-session environment. You should, as a standard at all times, avoid entering blanks in the runstring.

[]	parameters surrounded with brackets are optional and may be omitted. If the parameter is included in the runstring without brackets, it is required and must be supplied.
,	the comma is used to separate command parameters. Unless noted, command parameters must be given in the order shown in the command runstring. If an optional parameter is omitted, the comma should be used as a place-holder.
oplop2	where multiple options are allowed in a runstring, they must be given with no preceding, intervening, or trailing characters (including blanks).
lower case letters	represent variables for which you must supply a value in the runstring.

In the examples demonstrating the use of the utilities, underlined entries represent responses you must make to the utility prompts.

Conventions for forming the file descriptor, and other standard RTE-6/VM conventions, are found in the Index to Operating System Manuals (92084-90001).

Chapter 2
General System Utilities

Introduction

The general system utilities described in this chapter allow you to determine the current status of your system, to call for help when error conditions occur, to compile and load programs, to manipulate files, and to program the function keys (if present on your keyboard).

The utilities WHZAT and LGTAT display the current status of all memory partitions, scheduled and suspended files, and the status of the system and auxiliary subchannels. Using the help utility group, HELP, CMD and GENIX, you can display messages defining any system or FMGR error code. The GENIX utility allows you to construct any desired text message for display at your terminal.

With the utility COMPL you can compile a program for subsequent loading, or you can compile and load a program in one operation using CLOAD. Files can be merged using the MERGE utility, and compared for similarities and differences using SCOM.

The general system utilities KEYS and KYDMP are used in conjunction with display stations that include the function key cluster on the keyboard. You can program the function key cluster, at the upper-right section of the keyboard, to enter a line of ASCII characters or to perform a specific command or text formatting function.

Status Utility (WHZAT)

The WHZAT utility provides current system environment information. By calling a specific option, you can output the status of all memory partitions, all scheduled and suspended files, or only those programs associated with your session.

Invoke WHZAT, either in session mode or system mode, with the runstring:

```
[RU,]WH[ZAT][,lu[,option]]
```

where

lu is the logical unit number of the device on which the status is to be displayed. The default is the log device.

option defines the type of status to be output. The default is to output those programs associated with your session. The options are specified as:

AL all scheduled and suspended programs.

SM only scheduled and suspended state 3 programs having a father-son relationship.

PA all partitions in use.

PL all active programs.

Program States

A program can exist in one of seven states relative to the RTE-6/VM operating system environment:

0 Dormant - the program is not scheduled to run.

1 Scheduled - the program is in the schedule list.

2 I/O suspended - the system is currently performing an I/O operation requested by the program.

3 General wait - the program has requested services or resources that are temporarily unavailable, or has scheduled a second program, or a device is down.

4 Memory suspended - the program has requested an operation for which sufficient System Available Memory (SAM) is not currently available.

5 Disc suspended - the program has requested the use of more disc space than is currently available.

6 Operator/program suspended - the program is awaiting further operator action or a program EXEC call before it can continue.

For state 2 programs (I/O suspend), the status output contains the message:

```
2,dev:nnn,AV:n,ST:nnn
```

where

dev:nnn is the EQT or LU number, in decimal

AV:n is the driver-independent availability code:
- 0 = available
- 1 = EQT disabled
- 2 = busy
- 3 = waiting for DMA

ST:nnn is the status (in octal) of the EQT. Refer to the associated driver manual for a description of the status codes.

If the EQT or LU is unavailable (down), the code DN is inserted in the message and the status is noted in the summary message at the end of the WHZAT output.

For state 3 programs, a message defining the reason for the general wait suspension is given. Table 2-1 lists all messages together with their meaning.

For state 4 programs (memory suspend), the status output summary at the end of the report contains the message:

```
MAX CONT. SAM AVAIL:          nnn
TOTAL SAM AVAIL:              nnn
MAX CONT. SAM EVER AVAIL:     nnn
```

where nnn is an octal number. The available-memory message is output only when a program is in state 4. When sufficient memory is available for program needs, the message is suppressed and the program is rescheduled.

Table 2-1. General Wait State Messages

MESSAGE	REASON FOR WAIT
LULK lu,LKPRG=progx	The listed program attempted to put a lock on logical unit lu. Program progx already has a lock on lu. The listed program will be rescheduled when progx removes its lock.
RN xx,LKPRG=progx	The listed program attempted to set resource number xx. Program progx already has a lock on the resource number. The listed program will be rescheduled when progx removes the lock.
RESOURCE	The listed program attempted to allocate a resource number. The system has no more resource numbers available. The operating system will reschedule the listed program when a resource number is available.
CLASS #	The listed program requested a class number but the system has no more available. The operating system will reschedule the listed program when a class number becomes available.
CL xx	The listed program is waiting on completion of a class GET to class number xx.
progx	The listed program scheduled progx with wait. The listed program will be rescheduled when progx completes.
progx'S QUEUE	The listed program scheduled progx on the queue with wait. progx is not dormant so the listed program must wait. The listed program will be rescheduled after the scheduling of progx completes.
BL,EQT xx	Upper buffer limit exceeded on EQT xx entry. Will be rescheduled when I/O on EQT xx drops below lower buffer limit.
EQLK xxx,LKPRG=PRGA	Program suspended for a locked EQT. Will be rescheduled when EQT unlocked.
EQLK TABLE FULL	Program attempts to lock EQT and EQT lock table is full. Will be rescheduled when an entry in EQT lock table is released.

WHZAT Output, AL/SM Options

The output for the AL and SM option states is shown in Figures 2-1 and 2-2. Following the system time line (hour,minute,second, millisecond), the output data column codes are given. The codes are defined below; where special codes are used in the column entry, these also are defined.

PRGRM	Program name; ** = father program (precedes name)
T	Program type; E = EMA program
PRIOR	Program priority (0-32767); B = batch
PT	Partition number (1-64); 0 = memory-resident A = assigned to partition
SZ	Page size of program; ** = memory-resident
DO	Dormant (state 0)
SC	Scheduled (state 1)
IO	I/O suspended (state 2)
WT	General wait state (state 3)
ME	Memory suspended (state 4)
DS	Disc suspended (state 5)
OP	Operator suspended (state 6)
PRG CNTR	Program counter, in octal. Value is listed for all programs, regardless of state; SWP = program is swapped out
NEXT TIME	Time at which program will next execute; 000000 = program not yet initiated

The AL/SM output is presented in three distinct physical blocks: user session programs, state 3 programs having a father-son relationship and, for AL output only, all other scheduled and suspended programs.

The sample output of Figure 2-1 contains examples of deadlock situations. In the first example, SON2 has called on SON3 and suspended itself until SON3 terminates. SON3 calls on the suspended SON2, creating a deadlock. In the second example, LOCKB attempted to set a resource previously locked by LOCKA, creating another deadlock situation. Refer to the RTE-6/VM Programmer's Reference Manual for procedures for recovering from a deadlock.

```
WH,AL

10:36:14:600

 PRGRM T PRIOR PT SZ DO.SC.IO.WT.ME.DS.OP.    .PRG CNTR.  .NEXT TIME.

**FATHR 3 00099 19  5 * * * * *  3,SON1   * * * * * P:40107SWP
  SON1  3 00099 21  5 . . . . .  3,SON2   . . . . . P:40111
  SON2  3 00099 20  5 . . . . .  3,SON3   . . . . . P:40100
  SON3  3 00099 17  5 . . . . .  3,FATHR'S QUEUE  . P:40106
** BLOCK **
********** DEAD LOCK **
*** SEE ABOVE FOR REPORT ON FATHR
**FMG31 3 00090 22 11 * * * * *  3,LOCKA  * * * * * P:46722
  LOCKA 3 00099 16  5 . . . . .  3,LOCKB  . . . . . P:40145SWP
  LOCKB 3 00099  7  5 . . . . .  3,RN  059,LKPRG=LOCKA
                                                     P:40067SWP
** BLOCK **
********** DEAD LOCK **
*** SEE ABOVE FOR REPORT ON FMG31
.
  WHZAT 2 00041  7  5 .  1, . . . . . . . . . . . . P:43610
.
**FMG01 3 00090 15 11 * * * * *  3,EDI01  * * * * * P:46722
  EDI01 4 00051 19 16 . . . 2,EQ:  7,AV:2,ST:000  P:26102
.
  R$PN$ 1 00010  0  . . . . .  3,CL  061  . . . . P:45100
  GRPM  1 00004  0  . . . . .  3,CL  060  . . . . P:54017
  LOGON 3 00045 13 12 . . . .  3,CL  062  . . . . P:42107
    .   .   .   .      .        .     .       .      .
    .   .   .   .      .        .     .       .      .
    .   .   .   .      .        .     .       .      .
  UPLIN 1 00003  0  . 0,  . . . . . . . . . . . . P:00000 8: 0:45: 70
  RTRY  1 00020  0  . . . . .  3,CL  059  . . . . P:64632
  DISPL 3 32767 16  4 .  1, .  .  .   . . . . . . P:40036

ALL LU'S OK
ALL EQT'S OK

10:36:18:390
```

Figure 2-1. Program Status Mode (AL) Output.

Since WHZAT executes dynamically, while the state of the system is changing, a program whose status has been reported could be called by another program. When the status of the second program is reported, the first program status is reported as the son of the calling program, as in the following:

```
PRGA          *          3,PRGX
PRGX          .  1
  .
  .
  .
PRGB          **         3,PRGX
**       *  SEE ABOVE FOR REPORT ON PRGX **
```

The output summary gives the status of the LUs and EQTs (OK, up, down, or locked). The final entry is the time at which WHZAT exited.

```
:wh,sm

  8:16:37:850

  PRGRM T PRIOR PT  SZ DO.SC.IO.WT.ME.DS.OP.     .PRG CNTR. .NEXT TIME.
**FMG72 3 00051 13  12 * * * *  3,WHZ72 * * * * * P:42131
  WHZ72 2 00041  8   5 .  1.  . . . . . . . . . . P:40004
.
**FMG65 3 00051 18  12 * * * *  3,EDI65 * * * * * P:42131
  EDI65 4 00051 22  16 . . . 2,EQ: 13,AV:2,ST:002 P:25553
**FMG68 3 00051 12  12 * * * *  3,LOA68 * * * * * P:42131
  LOA68 4 00090 25  27 . . . 2,EQ: 16,AV:2,ST:002 P:30023
.
  FMG73 3 00051 14  12 . . . 2,EQ: 21,AV:2,ST:002 P:54701

ALL LU'S OK
ALL EQT'S OK

8:16:47:700
```

Figure 2-2. Program Status Mode (SM) Output.

WHZAT Output, PA Option

The partition status output provides a dynamic map of the activity in each partition. The output format is shown in Figure 2-3. Following the date line (hour,minute, second, millisecond), the output data column codes are given. These are defined below; where special codes are used in the column entry, these also are defined.

PTN#	Partition number (1-64); M = Mother C = Subpartition, Chain mode S = Subpartition available R = Reserved
SIZE	Program page size
PAGES	Physical pages where program resides
BG/RT	Program run type; BG = Background RT = Real time
SHR/LBL	Shareable EMA partition and Label; SH = Shareable ** = Mother with shareable subpartition * = Subpartition of mother
ACT	Number of current users of shareable EMA
L	Partition lock status; L = Locked
PRGRM	Program name

In the output, if a mother partition is currently being used for sharing EMA, the PRGRM entry will show EMA as the name of the program occupying the mother partition and subpartitions.

```
:WH,PA

10:58:12:440

PTN#  SIZE    PAGES   BG/RT SHR/LBL #ACT L PRGRM
 1      4     77-  80    RT                PRMPT
 2      4     81-  84    RT                R$PN$
 3      9     85-  93    RT                SMP
 4     10     94- 103    RT                <NONE>
 5      9    104- 112 BG                    REDTI
 6     10    113- 122 BG                    BLINK
 7     14    123- 136 BG                    <NONE>
 8     15    137- 151 BG                    <NONE>
 9     17    152- 168 BG                    <NONE>
10     19    169- 187 BG                    <NONE>
11     25    188- 212 BG    SH SHAR1    0   PTOPM
12     27    213- 239 BG                    <NONE>
13     32    240- 271 BG    SH PAR13    0   <NONE>
14     32    272- 303 BG                    RFAM
15M   180    304- 483 BG    SH PAR15    0   $EMA$
16C    20    304- 323 BG     *              $EMA$
17C    26    324- 349 BG     *              $EMA$
18C    32    350  381 BG     *              $EMA$
19C    32    382- 413 BG     *              $EMA$
20C    70    414- 483 BG     *              $EMA$
21M   180    663- 306 BG                    WHZ31
22C    13    484- 499 BG                    <NONE>
23C    32    500- 531 BG                    <NONE>
24C   360    532- 663 BG    **              <NONE>
25M   180    664-1023 BG     *              <NONE>
26S    30    844- 933 BG    SH PAR29        <NONE>
27-33  <UNDEFINED>

 MAXIMUM PARTITION SIZE AVAILABLE
        10 PAGES, BG  180 PAGES, MOTHER   360 PAGES
 MAXIMUM PARTITION SIZE AVAILABLE - DUE TO SHAREABLE EMA
 RT     10 PAGES, BG  100 PAGES, MOTHER   180 PAGES

10:58:14:120
```

Figure 2-3. Partition Status Mode (PA) Output.

WHZAT Output, PL Option

The active program status output, shown in Figures 2-4 through 2-6, provides a list of all active programs in the system. Following the date line (hour, minute, second, millisecond), the output data column codes are given. These are defined below; where special codes appear in the column entry, these also are defined.

NAME	Program name
TY	Program type (see following discussion)
PRIOR	Program priority, 1-32767
LADDR	Low memory address
HADDR	High memory address
LOBP	Low base page
HIBP	High base page
SZ	Program size, in pages
EMA	Extended Memory Area size
MSEG	Memory Segment size, in pages
LBL	Shareable EMA partition label, if used
PTN	Partition number, if assigned to program
TM	Load type; TE = Temporary PE = Permanent = memory-resident
COM	System common type; SC = System common RC = Reverse common NC = No common
LU	Disc LU on which program is stored
S-ID	Session identifier if program loaded under session monitor.

The status output summary identifies the number of free 33-word program ID segments (long), the number of free 9-word program ID segments (short) available in the system, and the number of 3-word ID extentions available.

The PL option can be invoked to specify a listing of specific program types, as:

```
[RU,]WH[ZAT],PL,tt
```

where tt specifies one of the following program types. Where both a mnemonic and a number are given, the type may be specified using either entry. (Refer to the RTE-6/VM Programmer's Reference Manual for a detailed description of program types.)

RT (or 2)	Real-time programs
BG (or 3)	Background programs
LB (or 4)	Large background programs
EB (or 6)	Extended background programs
1	Memory-resident programs (this program type can only be referenced by number).
5	Program segment (this program type can only be referenced by number).

The PL option also can be specified to list only those programs with like characteristics by including a wildcard character (-) in the "don't care" positions of the namr, as:

WH,PL,$-----	list only those programs whose namr begins with $.
RU,WH,PL,F-----	list only those programs whose namr begins with F.
wh,pl,--FTN-	list only those files with FTN in the third through fifth positions in the namr (list all FTN file modules).

```
:wh,pl,5

 8:16: 6:200

NAME TY PRIOR LADDR HADDR LOBP HIBP SZ  EMA MSEG LBL  PTN TM COM LU S-ID
GASP1 5       44166 53453  107  212                        PE      2
GASP2 5       44166 56355  107  253                        PE      2
FMGR0 5       41561 61165   36  222                        PE      2
FMGR1 5       41561 54721   36  202                        PE      2
FMGR2 5       41561 56506   36  355                        PE      2
FMGR3 5       41561 53407   36  135                        PE      2
FMGR4 5       41561 52606   36  302                        PE      2
FMGR5 5       41561 45536   36  102                        PE      2
FMGR6 5       41561 54524   36  224                        PE      2
FMGR7 5       41561 53553   36  133                        PE      2
FMGR8 5       41561 54120   36  113                        PE      2
FMGR9 5       41561 53262   36  115                        PE      2
FMGRA 5       41561 54053   36  130                        PE      2
FMGRB 5       41561 56066   36  201                        PE      2
XXDR1 5       34754 55726  560  650                        PE      2
XXDR2 5       34754 35765  560  572                        PE      2
XXDR3 5       34754 46255  560 1076                        PE      2
XXDR4 5       34754 40673  560  635                        PE      2
ASMB0 5       24347 34463  363  733                        TE      2
ASMB1 5       24347 26605  363  512                        TE      2
ASMB2 5       24347 26477  363  454                        TE      2
ASMB3 5       24347 25274  363  400                        TE      2
ASMB4 5       24347 25677  363  377                        TE      2
LODR1 5       35050 56063  510  565                        PE      3
LODR2 5       35050 36066  510  515                        PE      3
LODR3 5       35050 46365  510 1017                        PE      3
LODR4 5       35050 40765  510  562                        PE      3
EDIT0 5       26356 43033  516 1100                        TE      2
EDIT4 5       26356 43237  516  770                        TE      2
  <29 SHORT BLANK ID>

   55 FREE LONG IDS,     29 FREE SHORT IDS,    10 FREE ID EXTS

 8:16:30:970
```

Figure 2-4. Active Program Mode (PL) Output, Type 5 Programs Only

```
:wh,pl,bg

 8:13:55: 50
```

NAME	TY	PRIOR	LADDR	HADDR	LOBP	HIBP	SZ	EMA	MSEG	LBL	PTN	TM	COM	LU	S-ID
,,,,,	3	99	34000	41406	2	220	4					PE	NC	2	0
GASP	3	80	34000	56355	2	107	11					PE	NC	2	0
LOGON	3	49	34000	56722	2	477	11					PE	NC	2	0
LGOFF	3	52	34000	53335	2	305	9					PE	NC	2	0
FMGR	3	51	34000	61165	2	36	12					PE	NC	2	0
D.RTR	3	1	34000	63504	2	403	13				10	PE	NC	2	0
CMM6	3	90	34000	60025	2	416	12					PE	NC	2	0
T5IDM	3	40	34000	44735	2	201	6					PE	NC	2	0
PTOPM	3	30	34000	50063	2	155	8					PE	SS	2	0
MATIC	3	30	34000	34716	2	17	2					PE	SS	2	0
DINIT	3	26	34000	53272	2	442	9					PE	SS	2	0
DSMOD	3	26	34000	50404	2	265	8					PE	SS	2	0
METER	3	80	34000	52076	2	360	9					PE	NC	2	0
CLEAR	3	99	34000	34611	2	13	2					PE	NC	2	0
HELP	3	90	34000	45475	2	265	6					TE	NC	3	0
FMG65	3	51	34000	61165	2	36	12					TE	NC	2	65
FMG73	3	51	34000	61165	2	36	12					TE	NC	2	73
DRRPL	3	90	34000	66721	2	654	18					TE	NC	21	0
FMG72	3	51	34000	61165	2	36	12					TE	NC	2	72
FMG68	3	51	34000	61165	2	36	12					TE	NC	2	68
WHO	3	45	34000	52476	2	337	9					PE	NC	2	0

```
  55 FREE LONG IDS,     29 FREE SHORT IDS,    10 FREE ID EXTS

 8:14:14:770
```

Figure 2-5. Active Program Mode (PL) Output, Background Programs Only

```
:RU,WHZAT,PL,#-----

16:35:38:430
```

NAME	T	PRIOR	LADDR	HADDR	LBP	HBP	SZ	EMA	MSEG	LBL	PTN	TM	COM	LU	S-ID
#SEND	1	3	65606	67066	432	465							SS	2	
#DIAL	2	20	42000	42046	2	2	2					PE	NC	2	0

```
   19 FREE LONG IDS,     21 FREE SHORT IDS,    9 FREE ID EXTS

16:35:39:360
```

Figure 2-6. Active Program Mode (PL) Output, #----- Programs Only

Log Track-Assignment Table Utility (LGTAT)

The utility LGTAT is designed to display information about the system and auxiliary disc subchannel tracks. With LGTAT, you can specify the LU to which the output will be directed, and request either an abbreviated form or a full display of the output.

The abbreviated LGTAT output shows the total number of available tracks and the number of tracks in the largest contiguous track block. In addition to the abbreviated output information, the complete LGTAT output shows the Track Assignment Table for the system and auxiliary disc subchannels and the location of the start of the logical source tracks.

Invoke LGTAT using the runstring

```
[RU,]LGTAT[,lu[,form]]
```

where

lu is the logical unit to which LGTAT will direct the output. Default is the log LU.

form is the output format. Default is to the abbreviated form. The format is specified as

0 = abbreviated form
1 = complete form

Abbreviated LGTAT Output

If the abbreviated form is specified, LGTAT will output only the following information:

```
TOTAL AVAILABLE TRACKS= xxx
LARGEST CONTIGUOUS TRACK BLOCK= yyy
```

where xxx and yyy are the available tracks and the largest block of tracks, in decimal.

Complete LGTAT Output

If the complete form is specified, the LGTAT output is a complete listing of the track assignment summary for the system disc subchannel and (if present) the auxiliary disc subchannel, the total available tracks, and the largest contiguous block of tracks.

The Track Assignment Table, shown in Figure 2-7, lists all tracks on the associated disc subchannel. Each entry in the table corresponds to one track; Table 2-2 defines the entries that can appear in the table.

LGTAT dynamically reports swapped tracks. Large EMA programs that are swapped may not always appear to be contiguous, due to the slow I/O processing relative to the speed of track allocation and deallocation. That is, while LGTAT is reporting the contents of one track, the contents of other tracks may have changed.

A track is identified by the first program contained in that track; LGTAT labels the track as belonging to that program even if other programs also reside on it. LGTAT reports a system track following the system entry point tracks. (This track is currently reserved for system use.)

Table 2-2. Track Assignment Table Entries

progx	track belongs to program progx.
progx&	track holds memory-image form of program progx.
progx^	track holds progx, swapped from memory to disc.
--	free track.
ENTS	track holds point list for system and system library.
LS	track holds logical source tracks.
SYSTEM	track is a system track.
GLOBAL	track is a globally allocated track.
FMP	track is part of the file management package (FMP).
LIBRY	track is a system library track.

```
LGTAT,1,1
TRACK ASSIGNMENT TABLE        & =PROG ^ =SWAP

TRACK     0       1       2       3       4       5       6       7       8       9
   0 SYSTEM  SYSTEM  SYSTEM  SYSTEM  SYSTEM  SYSTEM  SYSTEM  SYSTEM  SYSTEM  SYSTEM
  10 SYSTEM  LUQUE&  IOMAP&  FMGR &  FMGR0&  FMGR1&  FMGR1&  FMGR2&  FMGR4&  FMGR5&
  20 FMGR6&  FMGR8&  FMGR9&  FMGRA&  FMGRA&  D.RTR&  D.RTR&  LOGON&  LOGON&  LGOFF&
  30 LGOFF&  GASP1&  SMP  &  JOB  &  MATIC&  PTOPM&  QCLM &  RPCNV&  EXECW&  VCPMN&
  40 RFAM &  DLIST&  OPERM&  CNSLM&  DSMOD&  DINIT&  DINIT&  M6   &  M6   &  READT&
  50 READT&  READT&  WRITT&  WRITT&  WRITT&  LOADR&  LODR1&  LODR1&  LIBRY   LIBRY
  60 LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY
  70 LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY
  80 LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY
  90 LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY   LIBRY    ENTS   D.RTR   D.RTR
 100 CMM6 &  CMM6 &  RT601   RT601   RT601   RT601   RT601   RT601   RT601   RT601
 110 RT601   RT601   RT601     --      --      --      --      --      --      --
 120   --      --      --      --      --      --      --      --      --      --
 130  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 140  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
  .    .       .       .       .       .       .       .       .       .       .
  .    .       .       .       .       .       .       .       .       .       .
  .    .       .       .       .       .       .       .       .       .       .
 280  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 290  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP   D.RTR

AUXILIARY DISC

   0   --      --      --      --      --      --      --      --      --      --
  10   --      --      --      --      --      --      --      --      --      --
  20   --      --      --      --      --      --      --      --      --      --
  30   --      --      --      --      --      --      --      --      --      --
  .    .       .       .       .       .       .       .       .       .       .
  .    .       .       .       .       .       .       .       .       .       .
  .    .       .       .       .       .       .       .       .       .       .
 140   --      --      --      --      --      --      --      --      --      --
 150  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 160  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 170  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 180  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 190  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
  .    .       .       .       .       .       .       .       .       .       .
  .    .       .       .       .       .       .       .       .       .       .
  .    .       .       .       .       .       .       .       .       .       .
 480  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP
 490  FMP     FMP     FMP     FMP     FMP     FMP     FMP     FMP   D.RTR   D.RTR

THE LS TRACK(S) ARE UNDEFINED
TOTAL AVAILABLE TRACKS =   187
LARGEST CONTIGUOUS TRACK BLOCK =   150
```

Figure 2-7. Complete LGTAT Output

System Configuration Utility LUPRN

The utility LUPRN supplies the current system device configuration and identifies each device by its true driver name. The configuration tables can be directed to your terminal or to a list device and can be ordered by system LU number, session LU number, (session mode only) or by device select code. LUPRN will execute in both session and non-session environments.

When loading LUPRN, specify a search (SE) of the library $RSLIB or %DECAR for the modules used by LUPRN, rather than relocating the selected library. Either library can be specified; however, $RSLIB will produce the fastest results.

Running LUPRN

Invoke LUPRN with the runstring

```
[RU,]LUPRN[,lst lu[,AL[,LU[,SC[,TY[,DV[,??]]]]]]]
```

where

lst lu — is the LU of the list output device. The default is to your terminal. While the options are not order- dependent, this parameter must be the first named option if you want to specify a list device other than your terminal.

AL — list all device drivers in the system, sorted by system LU number. If the device is included in your Session Switch Table (or available to your MTM station), the device SLU number is listed. If the device is not available for your use, the SLU column is blank for that device.

LU[:n:n] — list all device drivers for the system LUs, sorted by system LU number. When the optional range is specified, the list includes only those device drivers with LU numbers within the decimal range given by :n:n. Without the range parameters, this form of the command is equivalent to the AL option. LU references are system LUs; session LUs are listed for those LUs contained in your SST.

If only one :n parameter is given, or if the second :n is smaller that the first, only the first specified LU is listed. If no LUs exist within the given range, LUPRN issues the message

..no LUs found in specified range..

as the body of the table.

SC[:n:n] list all LUs assigned to specific select codes, sorted by select code number. When the optional range is specified, the list only includes those device drivers within the decimal range given by :n:n. If no drivers exist within the specified range, LUPRN returns the message

..no LUs found in specified range..

as the body of the table.

TY[:nB:nB] list all device drivers for the System LUs, sorted by driver type. When the optional range is specified, the list only includes those drivers that fall within the range given (use the DV option, if necessary, to get the full list of drivers in the system). Note that this range should be given in octal, using the B suffix. If the suffix is omitted, the table will list the driver that is equivalent to the decimal number; that is, if you enter the decimal values :12:15, the list will show the equivalent octal drivers, 14B through 17B. As with the LU option, the selected driver list will be taken from the entire system.

DV list the system driver table, followed by the LU table. This option is useful when you want to see the total number of drivers and their descriptions.

?? list the descriptive summary of LUPRN and the available optional runstring parameters.

If LUPRN is called with no options, the list will contain your session LUs only, sorted by session LU number. In a non-session environment, calling LUPRN with no options is equivalent to the call LUPRN,AL. Note that if you specify the SC option and either the LU:n:n or TY:nB:nB options, the table will be sorted by select code rather than LU or driver type.

If you specify both TY and LU with search ranges, an output may not result unless the specified combination matches an existing condition of the system. For example, the runstring

```
RU,LUPRN,LU:6,TY:12B
```

will result in an output only if LU 6 is assigned driver select code 12B.

If you specify an illegal list LU, the message

```
..Illegal LU (nn) for output (down or not defined)
```

is issued, and LUPRN exits. The erroneous LU number is given in the message.

If you invoke LUPRN with an illegal option code, the message

```
..Unknown command: xx ...use ??  for help.
```

is issued, where xx is the erroneous command code you entered.

Output Table Format

In the output table headings, the RTE operating system under which LUPRN is running is listed, together withthe system's date code. The "sorted by" message will indicate Session LU, System LU, or Select Code, depending on the option selected. The Time Base Generator select code is included, or the entry <none> is shown. If the Privileged Fence card is included in the system, the select code is given, or the entry <none> is shown. The number of memory partitions is given, or the entry <none> is shown. The configuration table contains the following column headings, as applicable for the format:

SLU - User session LU numbers (this column is included at both the left and right sides of the table). An LU number with a D suffix indicates that the LU is down. This column is omitted in non-session environments.

LU - System LU numbers (this column is included at both the left and right sides of the table). An LU number with a D suffix indicates that the LU is down.

EQT - Equipment table entry number.

sc - Subchannel number.

SCD - Select code.

Flags - D = DCPC, Dual Channel Port Controller
B = Buffered
P = Driver handles power fail
S = Driver handles timeout
T = Device has timed out.

Av - Device availability: 1 = Device down
2 = Device busy
3 = Device waiting DCPC.

If there is no entry in this column, the device is available.

Stats - Device status octal code. Note that the status is for the last driver activity, and may not reflect current status. (Refer to the related Device Driver Reference Manual for a description of the status codes.)

Driver- Driver name.

DP - First physical page number of the driver partition. If a dollars sign ($) follows the DP entry, this identifies the partition page as the System Driver Area (SDA).

Device Name - True name of device. Where possible, subchannel information is used to further identify the device.

Examples

In the following example, LUPRN is invoked with the DV option. The listing is sorted by driver number, and is followed by the full configuration listing, sorted by system LU number.

```
:RU,LUPRN,DV
 LUPRN's Driver List

 NUM  Name   = Description
 ---  -----    -----------
   1  DVR00    CRT/TTY terminal
   2  DVC00    CRT/TTY terminal
   3  DVD00    Logical Driver
   4  DVT00    306C Cent.printr
   5  DVM00    12792A 8-CH MUX
   6  DVX00    CIS BMUX Port
   7  DVS00    12790 MUX-DVR00
   8  DVV00    DS-Remotemapping
   9  DVR01    Papertape Reader
  10  DVC01    Papertape Reader
  11  DVR02    Papertape Punch
  12  DVC02    Papertape Punch
  13  DVR05    264x/2x Terminal
  14  DVS05    12920 MUX-DVR05
  15  DVT05    Tektronix CRT
  16  DVA05    Terminal w/modem
  17  DVQ05    BACI card GP DVR
  18  DVM05    8-CH MUX (DDV05)
   .    .          .
   .    .          .
   .    .          .
  73  DVS56    Preston A/D
  74  DVR57    12564 10 bit A/D
  75  DVR60    6129 Volt.source
  76  DVG61    12967 Sync Card
  77  DVR61    Obsolete Driver
  78  DVA62    2313 (RTE3)
  79  DVR62    2440 A/D (2313)
  80  DVR63    3000 UI Card
  81  DVS63    Parallel FDPLX
:
```

In the following example, LUPRN is invoked to list all system LUs within the range of LU 12 through LU 23. If no LUs are found within the range, the ..no LUs found in specified range.. message is issued as the body of the table. Note that the table contains entries for unidentifiable drivers, flagged with an asterisk. The configuration list also includes a note, Note 1., related to the unidentified driver entry. LUPRN provides seven notes related to specific conditions of the system. These notes are defined following the examples.

```
:RU,LUPRN,LU:12:23
                    RTE-6 System Device Configuration
                 RTE-6 System rev = 2226  LUPRN's rev = 2240
             1:52 PM TUE., 9 NOV., 1982...Sorted by System LU
Time Base (14B)  Priv. Fence SC (none)  Partitions (24) Memory size (576K)

SLU LU EQT,sc SCD Flags AV T.out Stats  Driver DP Device Name      LU  SLU
--- --------- --- ----- -- ----- -----  ------ -- ---------------- --- ---
     12 38,3  31B  B              130B  DVB12  45 2608A read-back   12
     13 39    71B  B                    DVZ12   4 2608A Graphics    13
     15 40    04B                       DVP43   9$Power Fail        15
 16  16  1,1  15B D                120B  DV?32*  4 .Unknown Driver.  16  16
     17  1,2  15B D                120B  DVR32   4 7905/6/20/25 DSK  17
 18  18  1,3  15B D                120B  DVR32   4 7905/6/20/25 DSK  18  18
     19  3,4  16B D                100B  DV?32* 41 .Unknown Driver.  19
     20  3,1  16B D                100B  DVR32  41 7905/6/20/25 DSK  20
     21  3,10 16B D                100B  DVR32  41 7905/6/20/25 DSK  21
 22  22  3    16B D                100B  DVR32  41 7905/6/20/25 DSK  22  22
     23  3,11 16B D                100B  DVR32  41 7905/6/20/25 DSK  23
--- --- ----- --- ----- -- ----- -----  ------ -- ---------------- --- ---

   Note1: DV?XX* indicates that the true driver name is not determinable
          since there are other drivers with the same INIT/CONT addresses.

     DP=Driver Partition page ($=SDA), SLU=Session LU
     (T.out is in seconds)                EQT Flags:
     LU # with a      EQT availability:    D=DCPC, B=Buffered, T=Timed-out
     D means the       1=down, 2=busy,     P=Driver handles Powerfail
     LU is down.       3=waiting DCPC      S=Driver handles Timeout

:
```

In the following example, LUPRN is invoked to list all driver types within the range of 5B through 12B, sorted by system LU number. If no drivers are found within the range specified, the ..no LUs found.. message is issued as the body of the table. In the listing, those devices designated as spool printers are identified as such in the Device Name column. Note the letter "D" following LU entry 86. This signifies that the LU is down.

```
:RU,LUPRN,TY:5B:12B
                    RTE-6 System Device Configuration
                RTE-6 System rev = 2226  LUPRN's rev = 2240
             1:53 PM TUE., 9 NOV., 1982...Sorted by System LU
Time Base (14B)  Priv. Fence SC (none)  Partitions (24)  Memory size (576K)

SLU LU EQT,sc SCD  Flags AV T.out Stats Driver DP Device Name      LU  SLU
--- -- ------ --- ----- -- ----- ----- ------ -- ---------------- --- ---
  6   6  6    26B  B       200.00       DVR12  45 2767 80col Prntr   6   6
 11  11 38    31B  B                130B DVB12  45 2608A Printer     11  11
     12 38,3  31B  B                130B DVB12  45 2608A read-back   12
     13 39    71B  B                     DVZ12   4 2608A Graphics    13
 56  56 13    32B  B S                 2B DVA05  47 Terminal w/modem  56  56
     57 14    33B  B S   2             2B DVA05  47 Terminal w/modem  57
     58 15    34B  B S   2             2B DVA05  47 Terminal w/modem  58
     59 16    35B    S     200.00      2B DVA05  47 Terminal w/modem  59
      .  .     .    .       .         .   .     .        .
      .  .     .    .       .         .   .     .        .
      .  .     .    .       .         .   .     .        .
     84 20,1  41B  B S     200.00      2B DVA05  47 Left CTU@ LU 63   84
     85 20,2  41B  B S     200.00      2B DVA05  47 Right CTU@ LU 63  85
     86D21,1  42B    S     200.00      2B DVA05  47 Left CTU@ LU 64   86
     87 21,2  42B  B S   2             2B DVA05  47 Right CTU@ LU 64  87
     88 22,1  43B  B S     200.00      2B DVA05  47 Left CTU@ LU 65   88
     89 22,2  43B  B S     200.00      2B DVA05  47 Right CTU@ LU 65  89
     94 37    44B  B S     100.00      2B DVA05  47 Terminal w/modem  94
    101 34    75B                          DVS12   9$Spool....type=12 101
    102 35    76B                          DVS12   9$Spool....type=12 102
    103 36    77B                          DVS12   9$Spool....type=12 103
    117 37    44B  B S     100.00      2B DVA05  47 Terminal w/modem 117
    118 37,1  44B  B S     100.00      2B DVA05  47 Left CTU@ LU117  118
    119 37,2  44B  B S     100.00      2B DVA05  47 Right CTU@ LU117 119
--- --- ----- --- ----- -- ------ ---  ----- -- ---------------- --- ---

  DP=Driver Partition page ($=SDA), SLU=Session LU
  LU # with a      EQT availability:     D=DCPC, B=Buffered, T=Timed-out
  D means the       1=down, 2=busy,      P=Driver handles Powerfail
  LU is down.       3=waiting DCPC       S=Driver handles Timeout

:
```

The following example invokes LUPRN with the SC option to list all devices within the range of 10B through 23B, sorted by select code. The listing contains the entry "<Empty Select Code>" to indicate that no Equipment Table entry was found for this device select code. The configuration table also contains the entries ".Unknown Driver." for those drivers whose identity could not be determined, and includes an explanatory note. (Refer to the section LUPRN Notes for the text of all notes that can be listed.)

```
:RU,LUPRN,SC:10B:23B
                    RTE-6 System Device Configuration
              RTE-6 System rev = 2226   LUPRN's rev = 2240
            1:54 PM TUE., 9 NOV., 1982...Sorted by Select Code
Time Base (14B)  Priv. Fence SC (none)  Partitions (24)  Memory size (576K)

SLU LU EQT,sc SCD Flags AV T.out  Stats Driver DP Device Name      LU SLU
--- -- ------ --- ----- -- ------ ----- ------ -- ---------------- -- ---
               10B  <Empty Select Code>
    104 41    11B   PS              26B DVA66  53 HDLC/BiSync card 104
    106 43    12B   PS               4B DVA66  53 HDLC/BiSync card 106
  7   7 10    13B   PS        .03    4B DVA65  51 DS1000 to 1000     7  7
              14B                     <RTE>     Timebase Generator
  2   2  1    15B D                 120B DV?32*  4 .Unknown Driver.   2  2
  3   3  3    16B D                 100B DV?32* 41 .Unknown Driver.   3  3
      1  2    17B    S   2 327.67    2B DVR00  45 CRT/TTY terminal   1
 55   5  5    20B                    40B DVR01  45 Papertape Reader  55  5
 44   4  4    21B  B       200.00       DVR02  45 Papertape Punch   44  4
 54  54 12    22B  B S     200.00       DVA37  49 59310 HPIB Card   54 54
  9   9  9    23B   PS        .03       DVA65  51 DS1000 to 1000     9  9
--- --- ----- --- ----- -- ------ ----- ------ -- ---------------- --- --

  Notel: DV?XX* indicates that the true driver name is not determinable
         since there are other drivers with the same INIT/CONT addresses.

  DP=Driver Partition page ($=SDA), SLU=Session LU
  (T.out is in seconds)                EQT Flags:
  LU # with a      EQT availability:     D=DCPC, B=Buffered, T=Timed-out
  D means the       1=down, 2=busy,      P=Driver handles Powerfail
  LU is down.       3=waiting DCPC       S=Driver handles Timeout

:
```

The following example invokes LUPRN with no options, to list the configuration of all devices available to the session, sorted by session LU number. Note the unknown driver entries, and the explanatory note following the table.

```
:RU,LUPRN
                    RTE-6 System Device Configuration
                RTE-6 System rev = 2226  LUPRN's rev = 2240
             1:55 PM TUE., 9 NOV., 1982...Sorted by Session LU
Time Base (14B)  Priv. Fence SC (none)  Partitions (24)  Memory size (576K)

SLU LU EQT,sc SCD Flags AV T.out Stats  Driver DP Device Name      LU SLU
--- -- ------ --- ----- -- ----- -----  ------ -- ---------------- -- ---
  1  67 48     70B    ST             2B  DVV05  51 Logical DVR =  5  67  1
  2   2 1      15B D                120B  DV?32*  4 .Unknown Driver.  2  2
  3   3 3,16   16B D                100B  DV?32* 41 .Unknown Driver.  3  3
  4  92 25     45B  B        200.00  40B  DVM00  55 12792A 8-CH MUX   92  4
  5  93 26     45B  B          5.00  40B  DVM00  55 12792A 8-CH MUX   93  5
  6   6  6     26B  B        200.00       DVR12  45 2767 80col Prntr   6  6
  7   7 10,1   13B   PS          .03       DVA65  51 DS1000 to 1000    7  7
  8   8  8     24B  B S        5.00   4B  DVR23  41 9TK Mag Tape #0    8  8
  9   9  9,1   23B   PS          .03       DVA65  51 DS1000 to 1000    9  9
 10  10  7     27B    S        5.00   1B  DVR23  41 9TK Mag Tape #0   10 10
 11  11 38     31B  B              130B  DVB12  45 2608A Printer      11 11
 16  16  1,1   15B D                120B  DVR32   4 7905/6/20/25 DSK  16 16
 18  18  1,3   15B D                120B  DVR32   4 7905/6/20/25 DSK  18 18
 22  22  3     16B D                100B  DVR32  41 7905/6/20/25 DSK  22 22
 37  67 48     70B    ST             2B  DVV05* 51 .Unknown Driver.  67 37
 38  38  3,5   16B D                100B  DVR32  41 7905/6/20/25 DSK  38 38
 39  69 50     70B                  40B  DVV00  51 DS-Remotemapping  69 39
 41  41  3,8   16B D                100B  DVR32  41 7905/6/20/25 DSK  41 41
 44   4  4     21B  B        200.00       DVR02  45 Papertape Punch    4 44
 54  54 12,2   22B  B S      200.00       DVA37  49 HPIB address # 2  54 54
 55   5  5     20B                  40B  DVR01  45 Papertape Reader    5 55
 56  56 13     32B  B S             2B  DVA05  47 Terminal w/modem   56 56
--- --- ----- --- ----- -- ------ ----  ------ -- ---------------- --- --

 Notel: DV?XX* indicates that the true drivername is not determinable
        since there are other drivers with the same INIT/CONT addresses.

 DP=Driver Partition page ($=SDA), SLU=Session LU
 (T.out is in seconds)                EQT Flags:
 LU # with a      EQT availability:     D=DCPC, B=Buffered, T=Timed-out
 D means the       1=down, 2=busy,      P=Driver handles Powerfail
 LU is down.       3=waiting DCPC       S=Driver handles Timeout

:
```

The last example invokes LUPRN with the AL option to list the configuration of all LUs in the system, sorted by system LU. Where a spooling printer is idle, the entry <idle> is shown in the Device Name column. This listing also identifies all LUs that are unassigned in the system.

```
:RU,LUPRN,AL
                    RTE-6 System Device Configuration
                RTE-6 System rev = 2226  LUPRN's rev = 2240
             1:58 PM TUE., 9 NOV., 1982...Sorted by System LU
Time Base (14B)  Priv. Fence SC (none)  Partitions (24)  Memory size (576K)

SLU LU EQT,sc SCD Flags AV T.out Stats  Driver DP Device Name      LU SLU
--- -- ------ --- ----- -- ------ ----  ------ -- ---------------- -- ---
      1  2    17B    S     327.67   2B  DVR00  45 CRT/TTY terminal   1
  3   3  3,16 16B D      2         101B DV?32* 41 .Unknown Driver.   3  3
 44   4  4    21B  B       200.00       DVR02  45 Papertape Punch   44  4
 55   5  5    20B                 40B   DVR01  45 Papertape Reader  55  5
  6   6  6    26B  B       200.00       DVR12  45 2767 80col Prntr   6  6
  7   7 10,1  13B   PS         .03       DVA65  51 DS1000 to 1000    7  7
  8   8  8    24B  B S       5.00   1B  DVR23  41 9TK Mag Tape #0    8  8
  9   9  9,1  23B   PS         .03       DVA65  51 DS1000 to 1000    9  9
 10  10  7    27B    S       5.00   1B  DVR23  41 9TK Mag Tape #0   10 10
 11  11 38    31B  B               130B DVB12  45 2608A Printer     11 11
     12 38,3  31B  B               130B DVB12  45 2608A read-back   12
     13 39    71B  B                    DVZ12   4 2608A Graphics    13
     14 11    61B  B       200.00  40B  DVR00  45 CRT/TTY terminal  14
     15 40    04B                       DVP43   9$Power Fail        15
 16  16  1,1  15B D               120B DVR32   4 7905/6/20/25 DSK   16 16
     17  1,2  15B D               120B DVR32   4 7905/6/20/25 DSK   17
 18  18  1,3  15B D               120B DVR32   4 7905/6/20/25 DSK   18 18
     19  3,4  16B D      2        100B DVR32  41 7905/6/20/25 DSK   19
      .   .    .       .          .    .    .        .             .
      .   .    .       .          .    .    .        .             .
      .   .    .       .          .    .    .        .             .
```

```
     .    .     .         .          .     .     .         .          .
     .    .     .         .          .     .     .         .          .
     .    .     .         .          .     .     .         .          .
 5   93 26     45B  B         5.00  40B  DVM00  55 12792A 8-CH MUX    5 93
     94 37     44B  B S     100.00   2B  DVA05  47 Terminal w/modem  94
     95 28     45B  B       200.00  40B  DVM00  55 12792A 8-CH MUX   95
     96 29     45B  B       200.00  40B  DVM00  55 12792A 8-CH MUX   96
     97 30     45B  B       200.00  40B  DVM00  55 12792A 8-CH MUX   97
     98 31     72B                       DVS43   9$Spooling (idle)   98
     99 32     73B                       DVS43   9$Spooling (idle)   99
    100 33     74B                       DVS43   9$Spooling (idle)  100
    101 34     75B                       DVS12   9$Spool....type=12 101
    102 35     76B                       DVS12   9$Spool....type=12 102
    103 36     77B                       DVS12   9$Spool....type=12 103
    104 41     11B   PS             26B  DVA66  53 HDLC/BiSync card 104
    105 42     11B   PS                  DVA66  53 HDLC/BiSync card 105
    106 43     12B   PS              4B  DVA66  53 HDLC/BiSync card 106
    107 44     12B   PS                  DVA66  53 HDLC/BiSync card 107
    108 45     62B                       DVA66  53 HDLC/BiSync card 108
    109 46     62B                       DVA66  53 HDLC/BiSync card 109
    110 12     22B  B S     200.00       DVA37  49 59310 HPIB Card  110
    111 12,1   22B  B S     200.00       DVA37  49 HPIB address # 1 111
    112 12,2   22B  B S     200.00       DVA37  49 HPIB address # 2 112
    113 12,3   22B  B S     200.00       DVA37  49 HPIB address # 3 113
    114 12,4   22B  B S     200.00       DVA37  49 HPIB address # 4 114
    115 51     67B                       DV?77+ 57 .Unknown Driver. 115
    116 52     66B                       DV?77+ 59 .Unknown Driver. 116
    117 37     44B  B S     100.00    2B DVA05  47 Terminal w/modem 117
    118 37,1   44B  B S     100.00    2B DVA05  47 Left CTU@ LU117  118
    119 37,2   44B  B S     100.00    2B DVA05  47 Right CTU@ LU117 119

        ************  System LU's 120 thru 122 Unassigned  *************
--- --- ------ --- ----- -- ------ -----  ------ -- ---------------- -- ---

Note1: DV?XX* indicates that the true drivername is not determinable
       since there are other drivers with the same INIT/CONT addresses.

Note2: DV?XX+ indicates that no entry point in the system matches either
       INIT/CONT addresses in this EQT.  Possibly a sysgen error or
       incomplete patch has been made, or this is a dummy driver.

DP=Driver Partition page ($=SDA), SLU=Session LU
(T.out is in seconds)                  EQT Flags:
LU # with a       EQT availability:      D=DCPC, B=Buffered, T=Timed-out
D means the        1=down, 2=busy,       P=Driver handles Powerfail
LU is down.        3=waiting DCPC        S=Driver handles Timeout
```

LUPRN Notes

The following explanatory notes can be displayed by LUPRN as specific related conditions are found in the system configuration:

Note1: DV?XX* indicates that the true driver name is not determinable since there are other drivers with the same INIT/CONT addresses.

Note2: DV?XX+ indicates that no entry point in the system matches either INIT/CONT addresses in this EQT. Possibly a system error or incomplete patch has been made, or this is a dummy driver.

The above two notes are related to the symbols * and + as they appear next to the driver number in the Driver column.

Note3: 'No EQT' indicates that no EQT claims use of this select code. Possibly a sysgen error was made in the EQT for an interrupt address.

The above note is related to the entry or entries identified by the message <Unknown Interrupt Table Entry - no EQT> following the SCD (select code) column.

Note4: The program name listed will be scheduled upon an interrupt from this Select Code. There is no EQT or driver associated with this S.C

The above note is related to the entry or entries identified by the message <Interrupt schedules program: (prog name) ...see Note 4> following the SCD column. The program name is given in the message.

Note5: The value listed is not an ID segment address and no EQT points to this Select Code.

The above note is related to the entry or entries identified by the message <Interrupt table entry unknown (value)...See Note 5> following the SCD column. The value is given in the message. This condition may be due to an incorrect patch or a corrupt system.

Note6: A subchannel for CS-80 disk is not part of track map table. This LU cannot be accessed.

The above note is related to the entry or entries identified by the message <CS80 subchan bad, (nn)> following the SCD columnn. The subchannel is given in the message. This may be due to a system error, or as a result of assigning an LU to a non-existent disc subchannel.

Note7: Unable to find track map table for CS-80 disk driver(s). The TMT is not in the system entry point list.

The above note is related to the fact that LUPRN only searches the system entry point list for the track map tables. This allows LUPRN to continue without asking the driver for the TMT entry. (The CS80 driver requires an I/O card to return the TMT entry.

If the device is a CS80 cartridge tape drive (CTD) and an assigned cache is not found in memory, LUPRN issues the following message:

> Warning: The CS-80 Cartridge Tape Unit is not setup with a disc cache. This will cause excessive wear on the drive.

LUPRN Errors

On an error condition, LUPRN issues the message

```
...LUPRN: entry error code = nn
```

where nn identifies the I/O call error code. Refer to the RTE-6VM Quick Reference Guide for a description of the I/O error codes.

Help Utilities (HELP, CMD, GENIX)

The help utility group supplies descriptive information and corrective action for error conditions, and provides a means for you to generate any desired text messages for display at your terminal or any selected device. The group includes three utilities: HELP, CMD, and GENIX.

HELP passes an error code or a key to CMD and schedules that utility to retrieve the associated message. CMD searches the disc-resident !HELP and/or !CMD files for the key and displays the associated message on your terminal or the device specified in the runstring. CMD also can be called directly to display a message retrieved from the disc-resident help files or from a specific message file named in the runstring. GENIX allows you to create files of keyed messages for display through calls to CMD. HELP and CMD each require 8Kw of memory; GENIX executes in a 16Kw partition.

HELP

The HELP utility provides a detailed explanation of an error code or message key. If no code/key is included in the command runstring, HELP retrieves the last error code generated. HELP then schedules the CMD utility and passes the code/key to CMD, together with the name of the help file (!HELP) to be searched. CMD searches the files to find an exact match for the error code/key and lists the associated text. HELP terminates when CMD completes the list function.

Calling HELP

Invoke HELP with the runstring

```
[RU,]HE[LP][,key[,lu]]
```

where

- key — is the word or phrase to be searched for in the indexed message file. If no key is given, the default is to the last error posted in the Session Control Block (SCB). If an error has not been posted, HELP outputs the message NO ERROR IN SCB and exits.
- lu — is the logical unit to receive the output. Default is to the log terminal.

Any valid session LU (0 to 255) is allowed. If the output is to a line printer, spooling should be enabled since the device is not locked during output. (For information on spooling, refer to the FMGR SL command in the RTE-6/VM Terminal User's Reference Manual.)

The key word or phrase can be from 1 to 24 characters in length; leading blanks are ignored. If, however, the key is a single character, it is considered unmatchable and HELP lists ten valid keys alphabetically surrounding the runstring character. If a key cannot be uniquely matched, a KEY NOT UNIQUE message is listed, together with the valid keys (to a maximum of ten) that match, in part, the runstring key. If the key cannot be matched at all, a KEY NOT FOUND message is listed, together with ten valid keys alphabetically surrounding the runstring key. In all cases, HELP exits after the unsuccessful match attempt.

Examples

```
:RU,HE,R                    Single-character entry
 LIST OF KEYS

 OF
 PACK
 PR
 QU
 READ02
 READ03
 READ04
 REV
 RN
 RP
:                           HELP exits

:he,axx                     Invalid key entry

 KEY NOT FOUND 'AXX'
 LIST OF KEYS

 AC
 ACI
  .
  .
  .
 BL
 BR
:                             HELP exits
```

```
:he,READ                        Nondefinitive entry

 KEY NOT UNIQUE 'READ'
 LIST OF KEYS

 READ 001
 READ 002
  .
  .
  .
 READ 008
:                                HELP exits
```

When the search is successful, HELP echoes the key word/phrase, then prints the associated text and exits. Examples:

```
:run,help,re

 RE

 A RE-ENTRANT SUBROUTINE ATTEMPTED
 TO CALL ITSELF.
:


:HE,FMGR-006

 FMGR-006

 FILE NOT FOUND
 AN ATTEMPT WAS MADE TO ACCESS A FILE THAT CANNOT BE FOUND.
 CHECK THE FILE NAME OR THE CARTRIDGE REFERENCE.
:
```

Error Messages

HELP generates the following error messages. (Note that error messages also may be generated by CMD running under HELP. Refer to the CMD utility for a description of CMD error messages.)

NO ERROR IN SCB

No key was given and the SCB has not had an error posted.

HELP PROGRAM ERROR HELP0001
NOT ENOUGH PARAMETERS
NOT OPERATING UNDER SESSION CONTROL AND A KEYWORD WAS NOT SPECIFIED. RE-ENTER THE COMMAND AND THE APPROPRIATE KEYWORD AS THE FIRST PARAMETER.

HELP PROGRAM ERROR HELP0005
HELP WAS UNABLE TO SCHEDULE THE PROGRAM CMD TO PRINT OUT ANY MESSAGE FOR YOUR KEYWORD.

CMD

The CMD utility provides the means for retrieving and listing messages for any interactive program on the RTE-6/VM operating system. Messages are generated by GENIX as indexed key files; CMD searches the file index for the matching key and lists the related message on your terminal or any LU designated in the runstring. Any message - expanded error description, command syntax, text - can be generated for retrieval. CMD can be called directly, or scheduled by HELP to run under that utility.

CMD supports an interactive mode that allows you to examine all the keys in a file and to call for additional information about a key if the message has been configured in multiple blocks.

Calling CMD

Invoke CMD with the runstring

```
[RU,]CMD[,key[,lu[,namr[,NI]]]]
```

where

key is the word or phrase to be searched for in the index file. If no key is given, CMD responds with a message describing how to use the utility.

lu is the logical unit (0 to 63) to receive the output. Default is to your terminal. If output is to a line printer, spooling should be enabled since the device is not locked during output. (For information on spooling, refer to the FMGR SL command in the RTE-6/VM Terminal Users Reference Manual.)

namr is the namr of the message file to be searched. The file must be resident on disc and must be created by GENIX. If a file is not specified, default is to the file !CMD.

NI directs CMD to run in noninteractive mode. Default is to interactive mode.

A key can be from 1 to 24 characters in length and can contain any ASCII character except comma (, used as a command delimiter) or slash (/ used as an interactive prompt). If a key is exactly one character in length, it is considered unmatchable. Leading blanks are ignored; however, they are considered in calculating the 24-character line length. If trailing blanks are significant they should be entered in the runstring using minus signs as placeholders. When calling CMD interactively, trailing blanks can be entered directly using the keyboard space bar.

Only the characters entered are used in the match. For example, the key LOA would be considered to be matched by the file key LOADR (but not the file key LO).

In noninteractive mode, all user prompts are suppressed and CMD exits after completing the list function associated with the runstring key word or phrase. If the key is matched, the entire message text is output; if the key is not matched or is not unique, the ten keys alphabetically surrounding it are output.

In interactive mode, if a key is not matched or is not unique, the prompt

```
[/]=MORE, OR KEY, OR [CR]?
```

is issued following the listing of the first ten alphabetically surrounding keys. Entering / allows you to continue the listing of valid keys in groups of ten; entering a new key word or phrase initiates a search of the file index for the new match; entering a carriage return exits CMD.

When the end of the valid-key list is reached, the message

```
\ BOTTOM OF LIST
 KEY, OR [CR]?
```

is issued to allow you the option of entering another key or exiting CMD.

When a key is matched, the MORE prompt is issued following the message if the information is formatted in multiple text blocks. (For example, a command syntax message could be in multiple blocks: the first block could contain only the command runstring parameters; a second message block could contain definitions of the parameters and would only be output in response to the MORE prompt.)

In creating the message file, GENIX allows you to specify a transfer to another file to retrieve all or a portion of a message, using the :TR function. When CMD encounters the :TR transfer control characters in the message, the transfer line is displayed on your terminal with the OK TO TRANSFER prompt, as

```
:TR, MSGS1, LONG
OK TO TRANSFER? [/] = YES, OR KEY, OR [CR]
```

This gives you the option of accepting or rejecting the transfer. When a :TR is encountered and you enter the / response, the current file is closed and CMD transfers to the new file.

Transfer commands can be nested; that is, the message in the transfer file can itself contain a :TR command to another file and key that could also contain a :TR command. In each case, CMD closes the current file and transfers to the new file. When the last block of the message has been displayed, CMD remains in the last open file and issues the KEY, OR [CR] prompt. The proper response here is [CR] to exit, since CMD is no longer in the original indexed file.

If a key matches two or more keys that point to different text blocks, a KEY NOT UNIQUE message is output, together with ten valid keys alphabetically surrounding the runstring key. If, however, the runstring key matches two or more keys that point to the same text block, it is considered a match and the text block is output. (Refer to the utility GENIX for a description of the multiple-key referencing scheme.)

If a key cannot be matched at all, the KEY NOT FOUND message is output, together with the ten valid keys alphabetically surrounding the runstring key.

Examples

```
:RU,CMD,LIST
 KEY NOT FOUND 'LIST'                    Invalid key entry
 LIST OF KEYS

 HE    HELP FCN      BR,FMG
 IF    CONDITIONAL     FMG
 IN    INIT CARTR      FMG
 IT    INTERVAL TIME    BR
 LGTT SYSTEM TRACKS
 LI    LIST FILE       FMG
 LL    SET LIST DEV    FMG
 LO    SET LOG DEV     FMG
 LOADR LOADER
 LU    SET SYSTEM LU    BR

 [/]=MORE, OR KEY, OR [CR]?LI        New key entered

 LI    LIST FILE       FMG                                  [10]

 LI,namr [,S [,nl [n2 ]]             Text message
         ,B
         ,D

 List file namr to list device (which can be set with LL command)

 [/]=MORE, OR KEY, OR [CR]? <cr>     <cr> exits
:
```

```
:cmd,lo                                  Nondefinitive key entry
 KEY NOT UNIQUE 'LO'
 LIST OF KEYS

  HE     HELP FCN       BR,FMG
  IF     CONDITIONAL      FMG
  IN     INIT CARTR       FMG
  IT     INTERVAL TIME     BR
  LGTT SYSTEM TRACKS
  LI     LIST FILE        FMG
  LL     SET LIST DEV     FMG
  LO     SET LOG DEV      FMG
  LOADR LOADER
  LU     SET SYSTEM LU     BR

  [/]=MORE, OR KEY, OR [CR]?/            List continuation command

  MC     MOUNT CART       FMG
  ME     DISP MESSG       FMG
  OF     TERM PRGM     BR,FMG
  ON     SCHED PRGM        BR
  OP     OPER CMD          BR
  PA     PAUSE            FMG
  PK     PACK CART        FMG
  PR     CHG PRIOR         BR
  PU     PURGE FILE       FMG
  QU     MOD TIMESL        BR

  [/]=MORE, OR KEY, OR [CR]? <cr>        Exit
:

:CMD,FMGR 000,,!HELP                     Search file specified

 FMGR 000

 BREAK
 THIS IS AN INFORMATIVE MESSAGE ONLY. NO ERROR HAS OCCURRED.

 KEY, OR [CR]? <cr>                      No continuation block
:

:RU,CMD                                  No key entry
 NO KEY

 RU,CMD [,key [,lu [,namr [,NI]]]] How-to-use-CMD information

 CMD     A general-purpose help facility program (this program!).

 [/]=MORE, OR KEY, OR [CR]? <cr>         Exit
:
```

Error Messages

CMD can generate the following error messages:

```
FMGR ERROR-xxx ON FILE yyyyyy
```

If the error is FMP -005, -006, -007, -008, -009, -012 or -032, the associated error message also is listed.

```
CMD CANNOT FIND YOUR HELP FILE.
NOTIFY SYSTEM MANAGER
```

```
CORRUPT HELP FILE
       1) MAKE SURE HELP FILE CREATED BY GENIX
       2) MAKE SURE HELP FILE IS TYPE 1 BINARY
       3) NOTIFY SYSTEM MANAGER
```

Input file may not have been indexed by GENIX, or may have been corrupted by backup from magnetic tape not in binary mode.

```
HELP FILE CURRENTLY IN USE BY ANOTHER
PROGRAM OR PROGRAMS
```

A lock error has occurred. File is exclusively locked by another program, or more than seven programs are simultaneously accessing the file.

```
SECURITY CODE FOR HELP FILE NOT VALID
NOTIFY SYSTEM MANAGER
```

Security code violation for access to file has occurred. (Note that CMD does not write to a file, and only needs read access.)

```
PARAMETER ERROR
       LIST LU GIVES AN ERROR ON WRITE
       GIVE VALID LU FOR SECOND PARAMETER
       OR DEFAULT TO YOUR TERMINAL
```

This may occur if the list LU is a disc or is non-existent.

GENIX

The GENIX utility allows you to create keyed text files for display through calls to the CMD utility. There are no restrictions on the size or the content of your text; however, each text block must be formatted as described below. For each input text file, GENIX constructs an indexed type 1 file that can be accessed by CMD. GENIX executes in a 16K partition, and can accommodate 44 key words or phrases per page of memory.

Calling GENIX

Invoke GENIX with the runstring

```
[RU,]GENIX,inamr[,lnamr],onamr
```

where

inamr is the input file of text and keys to be indexed.

lnamr is the list file or LU to receive the GENIX messages generated during processing of the file. The default is to the log device.

onamr is the output file name. The output file is disc-resident and may not be an LU. Any size or type parameters in the onamr descriptor are ignored.

The type 1 output file is created in the correct size after GENIX scans the input file to determine the space required. During processing, the input file text is saved in a temporary type 4 scratch area until the output file is generated.

If you invoke GENIX without specifying all required parameters, the following instruction message is issued:

```
TO RUN GENIX PLEASE USE THIS RUNSTRING

RU,GENIX,INPUT TEXT FILE,LIST FILE OR LU,NEW OUTPUT FILE

        THE LIST FILE MAY BE OVERLAID ONLY IF THE FIRST
        CHARACTER IS '.
        THE OUTPUT FILE MUST NOT ALREADY EXIST ON THE
        DESTINATION CRN
```

Input File Format

GENIX recognizes the following control characters entered in column 1 of the input; any other characters on the control line are ignored, except in the case of the transfer function (TR) where the record is searched for the transfer command.

""
A double quote signifies that the following record is a key word or phrase.
&
An ampersand signifies that the next block of text is a continuation of the preceding keyed message, and that the MORE prompt is to be issued at this point, before the record is displayed. The MORE prompt is of the form

```
[/]=MORE, OR KEY, OR [CR]?
```

If the response is / for MORE, CMD displays the information following the & control character. There is no limit on the number of information blocks that can be created for a key word (however, more text than necessary is generally counterproductive).
:TR,namr,key
A colon TR command initiates a transfer to the named file and key. The command is given as

```
:TR [,name [,key]]
```

where

name — is the name of the file to which CMD is to transfer. If the name is omitted, the current file is used for the search.

key — is the unique character string to be used in the search. If the key is omitted, the CMD utility NO KEY option is used. (Refer to the CMD Utility for examples of the NO KEY response.)

Transfer commands can be nested; that is, the message in the transfer file can itself contain a :TR command to another file and key that could also contain a :TR command. In each case, CMD closes the current file and transfers to the new file. When the last block of the message has been displayed, CMD remains in the last open file.

A key can be from 1 to 24 characters long. Any character may be used except the comma (,used as a command delimiter) and the slash (/ used as an interactive prompt). Key words may be entered in either upper or lower case, but may not be mixed within a word. All keys are converted by GENIX to upper case. Leading blanks are ignored; however, they are considered in the 24-character-length restriction. As an example, if the key "descriptors" is physically entered into the input file beginning at column 19, it is truncated to "descri" since the first 19 blank columns plus the key word exceed the 24-character limit.

Note that more than one key can reference the same message. In this case, the form of the entry is

```
""
KEY1
""
key2
Related text...
```

GENIX text format is entered on the line immediately following the key, in the exact form it is to appear; that is, upper and lower case, paragraph separator lines, etc. Text also may include any terminal escape sequences except NULL or EOT, as these characters are used by CMD. Any line that does not begin with one of the GENIX control characters is considered to be text. Note, however, that unless the text block is preceded by a valid control character entry, the entire block is ignored.

Examples

```
""
KEYA
This is the text for the first key. Note that the key can be given
in either upper or lower case, but not a combination of both.
""
keyb
""
KEYC
This is the text  for the second  and third keys.  Either key will
cause this text to be displayed.
&
The  ampersand  will cause  CMD to  issue the MORE  prompt at this
point when  either keyb  or KEYC is entered  in the CMD runstring.
The text is printed only when the "/" response, requesting further
information, is given.
&
Any number of message blocks can be generated.
""
LOAD
In this example the text message block is listed and a transfer is
directed below.
:TR,!LOAD,LOADER
When the :TR command is encountered CMD echoes the command on your
terminal, issues the OK TO TRANSFER prompt and, if the  / response
is given, closes  the current file  and opens  transfer file !LOAD
to search for the key LOADER.  Note that the :TR commands can  be
nested; each time a :TR is encountered, the current file is closed
and CMD opens the named transfer file.
""
ØØØØØØØØØØØØØØØØØØØØDESCRIPTORS
This key  will be indexed as DESCRI since the physical key word
line length exceeds 24 characters.  Leading  blanks are ignored in
indexing a key, but are considered in calculating the 24-character
line length.
```

Error Messages

GENIX can generate the following error messages. Fatal errors that will cause GENIX to abort are indicated with a (F) following the error description.

HEAP/STACK COLLISION IN LINE xxxx

Too many keys were found to save in the Pascal heap area. Resize the program for a larger memory partition and rerun.

NO KEYWORDS FOUND

The input file contains no key words at all. At least one key word is required for a valid index. (F)

WARNING: BLANK KEYWORD FOUND

A block of text preceded with a blank key word was found. File processing continues and the warning message is inserted in the file listing.

DUPLICATE KEYWORD 'xxx...x'

A duplicate key word was found. Any text associated with the duplicated key word is ignored. File processing continues and the warning message is inserted in the listing.

REGULAR PASCAL I/O ERRORS

FMP and I/O errors are handled using the standard Pascal methods. Note that the listed file names are the physical names supplied in the runstring rather than the Pascal logical file name. The name SCRATCH is used if an error occurs with the type 4 scratch file created by GENIX while processing the file.

Compile Utility (COMPL)

The COMPL utility allows you to automatically invoke the appropriate HP supported compiler or assembler for a specified source file.

Calling COMPL

Invoke COMPL using the runstring

```
[RU,]COMPL,snamr[,lnamr[,rnamr[,cs[,onamr]]]]
```

where

snamr is the source file to be compiled or assembled; snamr is the only required parameter. Default values for the other parameters depend on the format or the file contents of snamr.

lnamr is the disc file or LU to which the listing is to be sent. If the specified LU was set up at GASP initialization and is not an interactive device, the output is automatically spooled to that LU. In this case the message

```
SPOOL FILE = COXXYY
```

is issued, where

CO identifies the operation as a compile.

XX is the session identification number (or the terminal LU number if operating under the multiterminal monitor).

YY is the number of the COMPL or CLOAD spool file. A session is limited to a maximum of 80 COMPL or CLOAD spool files. (Refer to the Compile and Load Utility for a description of CLOAD.)

COMPL assumes that SPLCON and JOBFIL reside on the spool disc. If you do not want automatic spooling, specify the LU as "lu:NS" for Non-Spooling (for example, 8:NS). If lnamr is omitted and a minus sign is specified, and if snamr begins with an ampersand (&), the ampersand is replaced with an apostrophe (') and the list file is created under the snamr and stored on the same cartridge. If this parameter is omitted totally, or if snamr does not begin with an ampersand, the listing goes to your terminal.

rnamr is the file descriptor in which the relocatable code is to be placed. If this parameter is specified with a minus sign, or is omitted totally, and snamr begins with an ampersand (&), the ampersand is replaced with a percent sign (%) and the file is created under the snamr and stored on the same cartridge. If snamr does not begin with an ampersand, rnamr must must be specified or the relocatable code will not be saved.

cs is an optional control statement that allows you to override a control statement in the source file. Refer to the appropriate Compiler or Assembler manual for the format of the control statement.

onamr is the namr of the file containing a control string option list. This parameter can be used only with the Pascal compiler.

Examples

```
:RU,COMPL
```

By invoking COMPL with no parameters, you enter interactive mode and COMPL prompts with

```
NAMR(S),NAMR(L),NAMR(R),<C.S.>
```

You can respond with the namr of the source program and any desired optional parameters.

```
:RU,COMPL,&PROG          file name starts with ampersand (&)
```

Since the source file name starts with an ampersand, the listing goes to your terminal and the relocatable code is placed in a file named %PROG, created by COMPL.

```
:RU,COMPL,&PROG,-,-      file name starts with ampersand (&)
```

Since the source file name starts with an ampersand and the minus sign is specified for the lnamr and rnamr parameters, the list file goes to a disc file named 'PROG and the relocatable code is placed in a disc file named %PROG, both created by COMPL.

```
:COMPL,PROG              file name does not start with
                         ampersand (&).
```

The source file, PROG, is compiled and the listing goes to your terminal; no relocatable code file is created.

```
:COMPL,PROG,-,%PROG,FTN4,L,A
```

The source file, PROG, is compiled, the listing goes to your terminal and the relocatable code is placed in disc file %PROG, created by COMPL. The control statement in the source file, PROG, is overridden by the cs option statement FTN4,L,A.

```
:RU,COMPL,&PROG,6,-
```

The source file &PROG is compiled. The listing is spooled to LU 6 (the line printer) and the relocatable code is placed in the disc file %PROG, created by COMPL.

```
:COMPL,&PROG,6:NS,-
```

The source file &PROG is compiled. The listing goes to LU 6 (the line printer), but is not spooled. The relocatable code is placed in disc file %PROG, created by COMPL.

Compile and Load Utility (CLOAD)

The CLOAD utility allows you to automatically invoke the appropriate HP-supported compiler or assembler for a specified source file. In addition, CLOAD schedules LOADR to relocate the compiled code. Refer to the RTE-6/VM Loader Reference Manual for a detailed description of LOADR. Because LOADR is not specified directly, the following LOADR default options are used:

OPTION	DEFAULT
opcode	BG = Background program type NC = No Common (local common) TE = Temporary program load type
partition	Any available partition
size	As required by program; no additional logical address space
format	Base page links only

Calling CLOAD

Invoke CLOAD using the runstring

```
[RU,]CLOAD,snamr[,lnamr[,rnamr[,cs[,onamr]]]]
```

where

snamr is the source file to be compiled or assembled. This is the only required parameter.

lnamr is the LU to which the listing is to be sent. You may not specify a file as the lnamr; it must be an LU. If the specified LU was set up at GASP initialization and is not an interactive device, the output is spooled automatically to that LU. In this case the message

```
SPOOL FILE = COXXYY
```

is issued, where

CO identifies the operation as a compile.

XX is the session identification number (or the terminal LU number if operating under the multiterminal monitor).

YY is the number of the CLOAD spool file. A session is limited to a maximum of 80 CLOAD spool files.

CLOAD assumes that SPLCON and JOBFIL reside on the spool disc. If you do not want spooling, specify the LU as "lu:NS" (for example, 6:NS).

If lnamr is omitted and a minus sign is specified, and if snamr begins with an ampersand (&), the ampersand is replaced with an apostrophe (') and the list file is created under the snamr and stored on the same cartridge. If lnamr is omitted totally, or if snamr does not begin with an ampersand, the listing goes to your terminal.

rnamr is the file in which the relocatable code is to be placed. If this parameter is specified with a minus sign or is omitted totally, and snamr begins with an ampersand (&), the ampersand is replaced with a percent sign (%) and the file is created under snamr and stored on the same cartridge as the source file. If snamr does not begin with an ampersand, a file descriptor must be specified or the relocatable code will not be saved.

cs is an optional control statement that allows you to override a control statement in the source file. Refer to the appropriate Compiler or Assembler manual for the format of the control statement.

onamr is the namr of the file containing a control string option list. This parameter is usable only with the Pascal compiler.

Examples

```
:RU,CLOAD
```

If the utility is specified without any parameters, CLOAD prompts for the parameters with

```
NAMR(S),NAMR(L),NAMR(R),<C.S.>
```

```
:RU,CLOAD,&PROG          file name starts with ampersand (&)
```

Since the source file name starts with an ampersand, the listing goes to your terminal, the relocatable code is placed in a file named %PROG, and LOADR relocates %PROG creating a temporary program.

```
:RU,CLOAD,&PROG,-,-      file name starts with ampersand (&)
```

Since the source file name starts with an ampersand, the listing goes to your terminal, relocatable code is placed into a disc file named %PROG, and the LOADR relocates %PROG creating a temporary program.

```
:RU,CLOAD,PROG           file name does not start with
                         ampersand (&).
```

The source file, PROG is compiled. The listing goes to your terminal, but a relocatable code file is not created and LOADR is not scheduled.

```
:RU,CLOAD,PROG,-,%PRG,FTN,L,A
```

The listings go to your terminal, relocatable code is placed in a disc file named %PRG, and LOADR relocates %PRG creating a temporary program. The control statement in PROG is overridden by the cs FTN4,L,A.

```
:RU,CLOAD,&PROG,6,-
```

The listings are spooled to LU 6 (the line printer), the binary relocatable code is placed in the disc file %PROG, and LOADR relocates %PROG creating a temporary program.

```
:RU,CLOAD,&PROG,6:NS,-
```

The listings go to LU 6 (the line printer), but are not spooled. The binary relocatable code is placed in the disc file %PROG, and LOADR relocates %PROG creating a temporary program.

File Manipulation Utilities (MERGE, SCOM)

The file manipulation utilities allow you to merge files, to concatenate and store files, and to compare files to determine line-by-line differences. Two utilities, MERGE and SCOM, comprise the file manipulation group.

MERGE

The MERGE utility allows you to transfer files to an LU, or to concatenate files by either overlaying an existing file or storing them in a file created by MERGE. The names of the files to be concatenated or stored can be provided interactively from your terminal, from a command file on disc, or from an LU such as a minicartridge or magnetic tape.

Invoke MERGE with the runstring

```
[RU,]MERGE[,snamr[,dnamr]]
```

where

snamr is the command file or LU from which the names of the files to be concatenated or stored will be supplied. The default is to the log terminal.

dnamr is the destination file or LU that is to receive the files to be concatenated or stored. If an existing file is specified, it is overlaid by the concatenated files; if the file specified does not exist, MERGE creates it and stores the concatenated files there. If an LU is specified, MERGE transfers the files to that LU.

If neither snamr and dnamr are specified in the runstring, MERGE prompts with:

ENTER DESTINATION NAMR	enter the destination file or logical unit to which the files to be concatenated or stored are to be sent.
ENTER COMMAND NAMR	enter the name of the command file or LU that contains the names of the files to be concatenated or stored.

If the log terminal is specified as the command namr, MERGE prompts for the names of the files to be concatenated or transferred:

ENTER NAMR	enter the name of the file to be concatenated or transferred.

MERGE repeats the ENTER NAMR prompt until the code /E is entered to signify the end of the input file list.

```
ENTER NAMR /E

MERGE STOP
:
```

If the command namr is specified as a command file, MERGE searches for and executes the file. A command file could be specified as

```
FILEA:AA:1000,MAIN DATA BANK
```

A comment may be added to the command file by entering a comma following the file-name entry. The comment line remains with the command file and can serve to readily identify the file contents. It is not necessary to specify the end code /E in the command file; MERGE recognizes the end-of-file mark (EOF). If a file listed in the command file cannot be found, MERGE reports the error (FMGR-006) and continues executing.

After each file is written to the destination namr, a zero length record is written as a file delimiter. Subfile marks within a file are saved on concatenation, and an EOF is written on completion.

When concatenating files using a command file, the process can be halted before completion by entering break mode (i.e., striking any key on the terminal while there is no read pending) and entering the BR command. MERGE halts after processing the current file.

To append files to an existing file without destroying the file, enter the existing file descriptor as the first destination file. By doing this, the existing file is first overlaid with itself. The runstring sequence would be

```
:RU,MERGE
 ENTER DESTINATION NAMR  FILEA:AA:1000

 ENTER COMMAND NAMR 1

 ENTER NAMR  FILEA:AA:1000

 ENTER NAMR  FILEB::1000

 ENTER NAMR  FILEC::1000

 ENTER NAMR  /E

 MERGE STOP
:
```

SCOM

The utility SCOM compares two input files and identifies their differences by matching sequences of lines and flagging those lines that are unique to a single file. By specifying options, you can configure a listing of only those lines unique to either one or both files, or only those lines that are common to both, with or without line numbers. SCOM can be called programatically, or from your terminal.

Invoke SCOM with the runstring

```
[RU,]SCOM,namr1,namr2[,onamr[,option[,nmatch[,nchar]]]]
```

where

namr1
namr2 — are the input files to be compared. Both files must exist.

onamr — is the output file/device to receive the results of the comparison. The onamr must specify a new file to be created. Default is to your terminal.

option is any of the following:

F1 - list only the lines unique to namr1
F2 - list only the lines unique to namr2
BO - list the lines common to both files
NN - suppress line numbering
TB - include trailing blanks as a factor in determining a line match
Dx - use "x" as a "don't care" character
Cx - ignore blank lines with "x" in column 1.

Options must be specified without intervening commas, as ,NNF1F2C*,.

The default is F1F2. The default options remain in effect if other options are specified. However, if a default option is specified in the runstring, only that option is enabled with the selected options; the other default options are suppressed.

nmatch specifies the number of lines that must match in the files before a mismatch is ended. Default is three lines.

nchars is the maximum record size for the input files. Default is 156 characters.

If you call SCOM without specifying any parameters, the above runstring information is displayed to aid you in selecting the needed parameters.

SCOM compares the two input files and, when a mismatch is found, begins searching for a rematch. The files are considered rematched only when the number of lines (including blank lines) specified in nmatch are identical.

The Dx option allows you to specify a "don't care" character. It will match any character in the comparison file that is in the same line position as the "don't care" character. The "don't care" character can be any ASCII character (including a blank) except a comma, which is used as a runstring parameter delimiter. The D option is useful in creating a mask file for comparison, filling the "don't care" fields with the D option character.

The Cx option allows you to force SCOM to ignore specified lines in searching for a rematch of the files. If an otherwise blank line has the C option character in column one, the line is not considered in the search for a rematch. This option is useful for ignoring blank comment lines in a file. Typically, the C option character is * for assembler files and C for FORTRAN files. It also may be a blank, forcing SCOM to ignore totally blank lines in a file.

Invalid options in the runstring are ignored, and SCOM executes using only the valid options. Any entry in the nmatch or nchar parameters other than a positive integer greater than zero is ignored, and SCOM executes using the default values. SCOM does not issue a status message if any options are invalid and thus ignored; the run state is defined in the header of the comparison display.

Scheduling SCOM Programatically

SCOM can be scheduled programmatically to return completion status parameters and, if desired, output messages defining the results of the comparison. As an example, the following routine will schedule SCOM and output status messages keyed to the results of the comparison. The returned parameters can be recovered by calling the RMPAR routine described in the Relocatable Library Reference Manual.

```
DIMENSION IBACK(5),ISTNG(15),INAME(3)
DATA ISTNG /30HRU,SCOM,FILE1,FILE2,OFILE,F1F2/, ISTGL /15/

C Schedule SCOM, pass runstring (IDUM is a dummy parameter)

CALL EXEC(23,INAME,IDUM,IDUM,IDUM,IDUM,IDUM,ISTNG,ISTGL)

C Pick up returned parameters

CALL PRTN(IBACK)
IF (IBACK(1)) 700,500

C Successful completion

500 WRITE(6,*)"Comparison Complete"

C Test for truncated records

IF (IBACK(2) .EQ. 1) WRITE(6,*)"Input records truncated"

C test for identical files/mismatches

IF (IBACK(1) .EQ. 0) WRITE(6,*)"Files identical"

IF (IBACK(1) .GT. 0) WRITE(6,*)"Files differ", I6, " places." IBACK(1)

STOP

C Unsuccessful completion

700 WRITE(6,710)
710 FORMAT ("Comparison aborted")

IF (IBACK(1) .EQ. 100000B) WRITE(6,*)"RTE or operator error"

IF (IBACK(2) .EQ. -1) WRITE(6,*)"FO error", I6, ",ID", I6 IBACK(2), IBACK(3)

IF (IBACK(1) .EQ. -2) WRITE(6,*)"SC error", I6, ",ID", I6 IBACK(2), IBACK(3)
STOP
END
```

The three parameters, IBACK(1), (2), and (3) are coded to define:

```
Parameter 1:      0 = comparison complete, files identical
                n>0 = comparison complete, n mismatches found
                 -1 = comparison aborted, file-handling error
                 -2 = comparison aborted, SCOM error

Parameter 2:      n = FMP error code, if parameter 1 = -1
                  n = SCOM error code, if parameter 1 = -2
                  1 = record truncation flag, else 0

Parameter 3:      n = file number at which FMP error occurred, if
                      parameter 1 = -1
                  n = SCOM failure message number, if parameter
                      1 = -2.  Refer to Error Messages section for
                      assigned message numbers.
```

Status Interrogation

While SCOM is executing a file compare, you can request the current status of the compare by entering break mode (striking any key on the keyboard) and using the command BR. SCOM responds with one of the following messages, followed by the abort/continue prompt:

```
    /SCOM: Files presently match at: FILE1 line YYYYYY
                                     FILE2 line XXXXX
or

    /SCOM: Files presently mismatch at: FILE1 line XXXXXX
                                        FILE2 line YYYYYY
           ZZZZZZ subsequent lines compared in each file without
           rematch
and

    /SCOM: <cr> to abort
```

Examples

The following examples show the summary line that gives the result of the comparison. The summary information is included in the output only when the destination is your terminal; it is omitted if the output destination is a disc file.

Example 1:

```
:SCOM,TEST1,TEST2,,BO
 /SCOM: 11:29 AM  WED.,  8  JULY, 1981
 /SCOM: FILE1 = TEST1
 /SCOM: FILE2 = TEST2
 /SCOM: OPTIONS:        F1F2BO,  NMATCH =      3,  NCHARS =    156
-------------------------------------------------------------------

         1             FTN,L
                     1 FTN^^
-------------------------------------------------------------------

         2           2         PROGRAM FTEST
         3           3         IMPLICIT INTEGER (A-Z)
         4           4         LOGICAL W1,W2,WBOTH
         5           5         COMMON /MISC/ LUT,W1,W2,WBOTH
-------------------------------------------------------------------

         6                     LUT = 1
-------------------------------------------------------------------

         7           6         WRITE (1,10)
         8           7   10    FORMAT("N = ")
         9           8         READ(1,*) N
        10           9         CALL ERROR(N)
        11          10         END
        12          11         BLOCK DATA MISC
        13          12   C
        14          13         LOGICAL W1,W2,WBOTH
        15          14         COMMON /MISC/ LUT,W1,W2,WBOTH
        16          15   C
-------------------------------------------------------------------

        17                     END
        18                C
                    16         END$
-------------------------------------------------------------------

 /SCOM: COMPARISION OF FILES COMPLETE
 /SCOM: ---      3  DIFFERENCES WERE FOUND ---
:
```

In the above example, all defaults were taken. Column one represents TEST1 line numbers, column two represents TEST2. Where the line contents differ, the line number appears in the appropriate file column. The areas of differing lines are bracketed with dashed lines.

Example 2:

```
:SCOM,TEST1,TEST2,1,NNBO
 /SCOM: 11:34 AM  WED.,  8  JULY, 1981
 /SCOM: FILE1 = TEST1
 /SCOM: FILE2 = TEST2
 /SCOM: OPTIONS:      NNF1F2BO,  NMATCH =      3,  NCHARS =    156
------------------------------FILE 1 ONLY--------------------------

FTN,L
------------------------------FILE 2 ONLY--------------------------

FTN^^
-----------------------------------------------------------------

       PROGRAM FTEST
       IMPLICIT INTEGER (A-Z)
       LOGICAL W1,W2,WBOTH
       COMMON /MISC/ LUT,W1,W2,WBOTH
------------------------------FILE 1 ONLY--------------------------

       LUT = 1
-----------------------------------------------------------------

       WRITE (1,10)
10     FORMAT("N = ")
       READ(1,*) N
       CALL ERROR(N)
       END
       BLOCK DATA MISC
C
       LOGICAL W1,W2,WBOTH
       COMMON /MISC/ LUT,W1,W2,WBOTH
C
------------------------------FILE 1 ONLY--------------------------

       END
$
------------------------------FILE 2 ONLY--------------------------

       END$
-----------------------------------------------------------------

 /SCOM: COMPARISION OF FILES COMPLETE
 /SCOM: ---      3  DIFFERENCES WERE FOUND ---
:
```

In this example, the NN option suppresses line numbering. In this case the sections of differing lines are identified by the dashed line headers FILE 1 ONLY or FILE 2 ONLY.

Example 3:

```
:RU,SCOM,TEST1,TEST2
 /SCOM: 11:39 AM  WED.,  8  JULY, 1981
 /SCOM: FILE1 = TEST1
 /SCOM: FILE2 = TEST2
 /SCOM: OPTIONS:        F1F2   ,  NMATCH =      3,  NCHARS =    156

-------------------------------------------------------------------
-->> BEGINNING OF FILE <<--
         1              FTN,L
                     1  FTN^^
         2           2          PROGRAM FTEST
-------------------------------------------------------------------


-------------------------------------------------------------------
         5           5          COMMON /MISC/ LUT,W1,W2,WBOTH
         6                      LUT = 1
         7           6          WRITE (1,10)
-------------------------------------------------------------------


-------------------------------------------------------------------
        16          15  C
        17                     END
        18              $
                    16         END$
-->> END OF FILE <<--
-------------------------------------------------------------------


 /SCOM: COMPARISION OF FILES COMPLETE
 /SCOM: ---       3  DIFFERENCES WERE FOUND ---
:
```

In this example, the F1F2 options are specified, and only the differing lines are displayed. By specifying default options F1 and F2, you automatically suppress the B0 option that calls for displaying lines common to both files. In this case, SCOM adds the immediately preceding and following lines common to both files.

Example 4:

```
:RU,SCOM,TEST1,TEST2,,NN
 /SCOM: 11:40 AM  WED.,  8  JULY, 1981
 /SCOM: FILE1 = TEST1
 /SCOM: FILE2 = TEST2
 /SCOM: OPTIONS:      NNF1F2   ,  NMATCH =      3,  NCHARS =    156

-->> BEGINNING OF FILE <<--
------------------------------FILE 1 ONLY--------------------------
FTN,L
------------------------------FILE 2 ONLY--------------------------
FTN^^
---------------------------------------------------------------------
       PROGRAM FTEST


       COMMON /MISC/ LUT,W1,W2,WBOTH
------------------------------FILE 1 ONLY--------------------------

       LUT = 1
---------------------------------------------------------------------
       WRITE (1,10)

C
------------------------------FILE 1 ONLY--------------------------
       END

$
------------------------------FILE 2 ONLY--------------------------
       END$
---------------------------------------------------------------------
-->>  END OF FILE  <<--

 /SCOM:  COMPARISON OF FILES COMPLETE
 /SCOM:  ---     3 DIFFERENCES WERE FOUND
:
```

In this example, specifying the NN option suppresses line numbering.In this case SCOM adds the immediately preceding and following lines common to both files.

Example 5

```
:scom,test1,test2,,d^BO
 /SCOM:  10:42 AM WED.,    8 JULY, 1981
 /SCOM:  FILE1 = TEST1
 /SCOM:  FILE2 = TEST2
 /SCOM:  OPTIONS:  D^  F1F2BO,     NMATCH =      3,    NCHAR =    156
          1         1  FTN,L
          2         2         PROGRAM FTEST
          3         3         IMPLICIT INTEGER (A-Z)
          4         4         LOGICAL W1,W2,WBOTH
          5         5         COMMON /MISC/ LUT,W1,W2,WBOTH
-------------------------------------------------------------------------
          6                   LUT = 1
-------------------------------------------------------------------------
          7         6         WRITE (1,10)
          8         7  10     FORMAT ("N - ")
          9         8         READ(1,*) N
         10         9         CALL ERROR(N)
         11        10         END
         12        11         BLOCK DATA MISC
         13        12  C
         14        13         LOGICAL W1,W2,WBOTH
         15        14         COMMON /MISC/ LUT,W1,W2,WBOTH
         16        15  C
-------------------------------------------------------------------------
         17                   END
         18             $
                   16         END$
-------------------------------------------------------------------------

 /SCOM:  COMPARISON OF FILES COMPLETE
 /SCOM:  ---      2  DIFFERENCES WERE FOUND ---
:
```

In this example, the ^ is specified as a "don't care" character, thus the first lines of the two programs now match. Note that the text of matching lines is always read from FILE1.

Example 6:

```
:SCOM,TEST3,TEST4,,BO,1
 /SCOM: 11:42 AM  WED.,  8  JULY, 1981
 /SCOM: FILE1 = TEST3
 /SCOM: FILE2 = TEST4
 /SCOM: OPTIONS:          F1F2BO   NMATCH =     1,   NCHAR =     156
         1         1   ASMB,R,L,C
         2         2           NAM TEST3
         3         3   *
-------------------------------------------------------------------
         4             B       EQU 1
                   4   A       EQU 0
-------------------------------------------------------------------
         5         5   *
-------------------------------------------------------------------
                   6   B       EQU 1
                   7   *
-------------------------------------------------------------------
         6         8   START   HLT
         7         9           END START

 /SCOM:  COMPARISON OF FILES COMPLETE
 /SCOM:  ---     2  DIFFERENCES WERE FOUND ---
:
```

In this example, two similar files are compared with the NMATCH set to one line for rematch of identical lines. The difference between the files is the addition of two instructions (lines 4 and 5) to file TEST4:

```
4   *
5   A    EQU 0
```

In this case, SCOM did not detect the difference since it found that line 5 in each file matched. Because of this, SCOM could not detect the more preferable rematch of line 4 in TEST3 and line 6 in TEST4. This problem of extraneous identical lines causing erroneous rematches occurs most often with blank comment lines. Example 7 demonstrates a method of avoiding this problem.

Example 7:

```
:RU,SCOM,TEST3,TEST4,,C*BO,1
 /SCOM: 11:45 AM  WED.,  8  JULY, 1981
 /SCOM: FILE1 = TEST3
 /SCOM: FILE2 = TEST4
 /SCOM: OPTIONS:  C*    F1F2BO,  NMATCH =      1,  NCHARS =    156
         1          1  ASMB,R,L,C
         2          2          NAM TEST3
         3          3  *
-------------------------------------------------------------------

                    4  A       EQU 0
                    5  *
-------------------------------------------------------------------

         4          6  B       EQU 1
         5          7  *
         6          8  START   HLT
         7          9          END START

 /SCOM: COMPARISION OF FILES COMPLETE
 /SCOM: ---      1  DIFFERENCES WERE FOUND ---
:
```

In this example, the C* option specifies that all otherwise blank lines with * in column 1 are ignored when searching for a rematch in the files. The result is a more desirable rematch after the mismatch at line four in each file.

Error Messages

Error messages are only displayed at the log terminal; they are not included in the output file. Fatal errors that cause SCOM to abort are identified with (F) following the error description.

WARNING: RECORD TOO LONG

A record has been read whose length is greater than that specified by the NCHRS parameter. The record is truncated and the comparison may be invalid. This is only a warning and does not abort SCOM.

FMP ERROR = -xxx ON FILE xxxxxx:xx:xx

The indicated file access error was encountered on the named file. (F)

#1. USER SUPPLIED MAXIMUM RECORD SIZE IS TOO LARGE

There is not sufficient internal buffer space to accommodate the specified NCHARS. Specify a smaller NCHARS value and rerun SCOM. (F)

#2. ILLEGAL INTERNAL SUBROUTINE PARAMETER

The subroutine was called with an invalid parameter; a utility self check has failed. Reload SCOM and rerun. (F)

#3. CACHE DATA STRUCTURE CORRUPT

The internal check of the buffer data structure shows corruption; a utility self check has failed. Reload SCOM and rerun. (F)

Function-Key Manipulation Utilities (KEYS, KYDMP)

Two utilities, KEYS and KYDMP, are provided for use in conjunction with display stations that include the function-key cluster on the keyboard. With the utility KEYS, you can define a specific set of functions to be performed by the keys. For example, a function could be defined to output an often-used text combination (up to 80 characters), system logon/logoff messages, or text formatting commands. The utility KYDMP programs the keys, termed soft keys, to perform the functions defined by KEYS.

Any soft keys programmed using KEYS and KYDMP can be reset to their default values by pressing the RESET TERMINAL key twice. The current status of the soft keys can be displayed by simultaneously pressing the CNTL and NEXT PAGE keys on the display station keyboard.

KEYS

KEYS allows you to define a specific set of functions to be performed by the eight soft keys (f1 through f8) on the display station keyboard. With KEYS you can create a set of commands, list an existing command-set file, modify the command set, and store the new file or modified file on disc or on a mini-cartridge.

KEYS is a totally interactive utility that issues prompts for each step in the creation/modification of a command set. Responses must be entered as upper-case characters; if lower-case characters are entered, KEYS repeats the prompt and waits for the proper response.

Calling KEYS

Invoke KEYS with the runstring

```
[RU,]KEYS[,console[,list]]
```

where

console is the LU of the display terminal to receive the KEYS prompts. Default is to LU 1 (your terminal).

list is the device on which the command-set file is to be listed. Default is to your terminal.

KEYS responds with the prompt

```
ENTER ONE OF THESE FUNCTIONS: [CREATE,MODIFY,OUTPUT,LIST]
OR PRESS [RETURN] TO TERMINATE THIS PROGRAM:
```

If your response is C for create, the following sequence of prompts is issued:

```
ENTER [SOFT KEY NUMBER (1-8)] TO BE PROGRAMMED OR
PRESS [RETURN] IF LAST ASSIGNMENT HAS BEEN MADE:
1
```

KEYS echoes the response with

```
SOFT KEY ASSIGNMENT FOR FUNCTION KEY 1
```

and then prompts for a label assignment. The label you assign to identify each key's function is displayed at the top of the screen when the keys are programmed by KYDMP.

```
ENTER UP TO [16 CHARACTERS] FOR SOFT KEY LABEL OR
PRESS [RETURN] IF NO LABEL IS TO BE ASSIGNED:
KEY1 LABEL
```

The next prompt asks you to define the command string type:

0 = command string is transmitted to the computer and is echoed at your terminal

2 = command string is transmitted to the computer but not echoed, and a carriage return/linefeed is generated.

The default is to a type 2 command string.

```
ENTER [0] FOR NORMAL OR [2] FOR TRANSMIT ONLY
COMMAND STRING TYPE:
0

ENTER [UP TO 80 CHARACTERS] FOR SOFT KEY COMMAND
STRING TO BE ASSIGNED TO THIS KEY OR PRESS [RETURN]
TO DEFAULT TO STANDARD COMMAND STRING:
KEY 1 COMMAND STRING
```

The ENTER SOFT KEY prompt sequence is repeated until you enter a carriage return to indicate to KEYS that the last assignment has been made.

```
ENTER [SOFT KEY NUMBER (1-8)] TO BE PROGRAMMED OR
PRESS [RETURN] IF LAST ASSIGNMENT HAS BEEN MADE:
2

SOFT KEY ASSIGNMENT FOR FUNCTION KEY 2

ENTER UP TO [16 CHARACTERS] FOR SOFT KEY LABEL
OR PRESS [RETURN] IF NO LABEL IS TO BE ASSIGNED:
<cr>

ENTER: [0] FOR NORMAL OR [2] FOR TRANSMIT ONLY
COMMAND STRING TYPE
<cr>
```

The <cr> return causes the command string type to default to type 2.

```
ENTER [UP TO 80 CHARACTERS] FOR SOFT KEY COMMAND
STRING TO BE ASSIGNED TO THIS KEY OR PRESS [RETURN]
TO DEFAULT TO STANDARD COMMAND STRING
KEY 2 COMMAND STRING

ENTER [SOFT KEY NUMBER (1-8)] TO BE PROGRAMMED OR
PRESS [RETURN] IF LAST ASSIGNMENT HAS BEEN MADE:
<cr>
```

The create operation is now complete. KEYS now prompts again with the ENTER FUNCTION message. This allows you to list and/or output the created command set to a file on disc or mini-cartridge. If the LIST function is selected, the following sequence of prompts is issued.

```
ENTER ONE OF THESE FUNCTIONS: [CREATE,MODIFY,OUTPUT,LIST]
OR PRESS [RETURN] TO TERMINATE THIS PROGRAM:
LIST

ENTER [FILE NAME,SECURITY CODE,CARTRIDGE] OR [2645A LU]
WHERE SOFT KEY COMMAND SET TO BE LISTED IS STORED OR
PRESS [RETURN] TO LIST DIRECTLY FROM THIS PROGRAM:
```

Since the sequence of operations in this example began with a CREATE, the correct response here is a carriage return to direct the listing to your terminal from KEY itself. The listing shows the labels assigned to the keys, the command string type, and the string. Only the assigned key labels, strings and the string types are listed; the default functions are not shown. The listing for the command set created above is

```
KEY1 LABEL
0
KEY1 COMMAND STRING

2
KEY2 COMMAND STRING

2
.
.
.
2
```

Following the listing, KEYS again issues the ENTER FUNCTION prompt. This allows you to output the command set file to disc or to a minicartridge. The output prompt sequence is

```
ENTER ONE OF THESE FUNCTIONS: [CREATE,MODIFY,OUTPUT,LIST]
OR PRESS [RETURN] TO TERMINATE THIS PROGRAM:
O

ENTER [FILE NAME,SECURITY CODE,CARTRIDGE] OR [2645A LU]
WHERE SOFT KEY COMMAND SET TO BE OUTPUT IS STORED OR
PRESS [RETURN] TO OUTPUT DIRECTLY FROM THIS PROGRAM:
<cr>
```

In this example, the proper response to this prompt is a carriage return since the file is being created.

```
ENTER [FILE NAME,SECURITY CODE,CARTRIDGE] OR [2645A LU]
TO WHICH COMMAND SET IS TO BE OUTPUT OR [RETURN] TO
REPLACE ORIGINAL FILE OR LU:
KEYFIL
```

By entering a file name, you direct KEYS to store the created command-set file on disc. To store the file on a minicartridge, use LU 4. When storing on a minicartridge, take care to position the tape correctly to prevent writing over an existing file.

Modifying an Existing Command-Set File

To modify a command-set file, first LIST the file and then enter modify mode. The prompt sequence is

```
ENTER ONE OF THESE FUNCTIONS: [CREATE,MODIFY,OUTPUT,LIST]
OR PRESS [RETURN] TO TERMINATE THIS PROGRAM:
M

ENTER [FILE NAME,SECURITY CODE,CARTRIDGE] OR [2645A LU]
WHERE SOFT KEY COMMAND SET TO BE MODIFIED IS STORED OR
PRESS [RETURN] TO CONTINUE MODIFYING A COMMAND SET IN THIS PROGRAM:
KEYFIL
```

Following this prompt, the sequence of KEY prompts is the same as that for creating a command set.

Error Messages

KEYS can generate the following messages. All errors are fatal, causing the utility to abort.

```
KEYS HAS BEEN ABORTED!
```

The abort character (A) has been entered and KEYS has exited.

```
FILE MANAGER ERROR -xx HAS OCCURRED
```

File Manager error xx has occurred in trying to create, open, read, write or close a file.

ERROR IN READING COMMAND SET FROM LU!

The number of words read from a command set does not conform to standard format, or a device error has occurred.

NO ORIGINAL FILE OR LU EXISTS

An attempt has been made to output to an original file or LU that was not read in by KEYS.

KYDMP

The utility KYDMP programs the display station soft keys to perform the functions defined in the command set generated with KEYS.

To load the terminal soft keys from a file created by KEYS, use FMGR command DU:

```
DU,KEYFIL,1
```

where KEYFIL is the name of the file created by KEYS and 1 is the session LU number of your terminal. From either FMGR or the operating system, or to load the terminal soft keys programmatically, you can use program KYDMP:

```
RU,KYDMP,,KE,YF,IL
```

where the double comma defaults to LU 1. A file security code can be included in the run string, as described in the following subsection.

Calling KYDMP

To program the soft keys from a KEYS file stored on disc, invoke KYDMP with the runstring

```
[RU,]KYDMP[,console],name[,sc]
```

where

console is the display terminal whose soft keys are to be programmed. The default is your terminal.

name is the KEYS file containing the command set. The filename must be specified as three pairs of

characters separated by commas - fi,le,na. Note that whenever the file name is separated into character pairs in the runstring, each pair must have at least one non-numeric character, or a non-numeric character must be appended to the pair. For example, to dump the following KEY files to LU1:

```
RU,KYDMP,,AB,CD,E6        file name ABCDE6
RU,KYDMP,,AB,CD,56A       file name ABCD56
RU,KYDMP,,AB,C4,5F        file name ABC45F
RU,KYDMP,,AB,34A,EF       file name AB34EF
RU,KYDMP,,A2,34A,56A      file name A23456
```

If the file name is shorter than five characters, and a security code follows in the runstring, include a comma for each missing pair of characters.

sc is the file security code.

To program the soft keys from a KEYS file stored on a mini-cartridge, invoke KYDMP with the runstring

```
[RU,]KYDMP[,console],ctu
```

where

ctu is the LU number of the display station cartridge tape unit.

KYDMP reads the soft key command set from the specified source and outputs it to the specified display station to program the soft keys. The function key labels assigned with KEYS are displayed in inverse video at the top of the screen, as shown in Figure 2-8. If a function key has not been assigned a label, that label field is blank. The display is protected from being written over and remains at the top of the screen. Each time the function keys are reprogrammed, the label field is updated and protected.

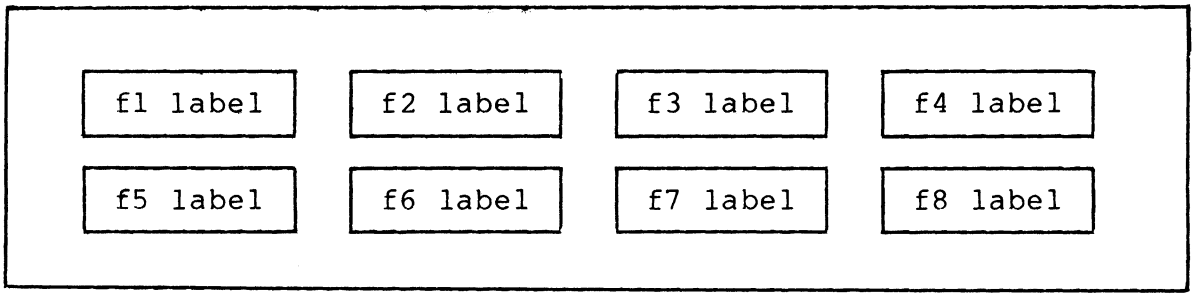


Figure 2-8. Soft Key Label Display Format

Error Messages

The following error messages can occur when running the KYDMP utility. All error conditions are fatal, causing KYDMP to abort.

NO SECOND PARAMETER SPECIFIED OR NEGATIVE

The second parameter was not specified or was specified as a negative. This message also is issued if the file name was specified other than as three pairs of ASCII characters separated with commas. The commas must be included even if the file name is less than six characters.

FMGR ERROR -xxx WHEN READING FROM FILE

File Manager error xx has occurred in trying to open the file, or an incorrect security code was specified. For error - 6 (file not found), note handling of numeric character pairs described above for the runstring parameter fi,le,na. Error - 7 indicates incorrect security code.

ERROR WHEN READING FROM CTU

The mini-cartridge was not mounted or was positioned at the wrong file.

Chapter 3
Record Reconfiguration Utility

Introduction

OLDRE is a stand-alone utility that provides backward compatiblity with earlier operating systems and generators. Using OLDRE, you can translate the new RTE-6/VM relocatable record formats to the formats used by earlier language processors. Translations are provided for source programs written in MACRO/1000, Pascal and FORTRAN.

In record word two, new records are identified with a seven in the identifier field (bits 13 - 15), and with one of the following sub-identifiers in bits 6 - 12:

Sub-Ident	New Record	Type
1	XNAM	Extended name relocatable program. Superset of NAM.
2	XENT	Extended entry point. Superset of ENT.
3	XDBL	Extended define left byte address. Superset of DBL.
4	XEXT	Extended external reference. Superset of EXT.
5	XEND	Extended terminate program. Superset of END.
6	RPL	Contains code-replacement information.
8	MSEG	Declares EMA memory segment size.
9	ALLOC	Combined ENT/EXT for FORTRAN block data.

OLDRE creates a scratch file into which the translated relocatable file is built. Each record is scanned and, if a non-translatable feature is encountered, the record is skipped and an appropriate message is issued. OLDRE does not halt on an error, but continues translation of the entire file. If non-translatable code has been encountered, the scratch file is purged following the translation attempt and the original relocatable file is left unchanged. If the translation is completed successfully, the original relocatable file is purged and the scratch file is truncated to the exact size required to hold the translated code and renamed with the original file namr. The resulting translated file then can be loaded to execute under the appropriate operating system.

OLDRE processes MACRO/1000 code at approximately 800 lines per second, Pascal code at approximately 100 lines per second, and FORTRAN code at approximately 133 lines per second.

Running OLDRE

Invoke OLDRE using the runstring

```
[RU,]OLDRE,namr
```

where namr must begin with a percent sign (%), and identifies the relocatable file that combines old-format records and new-format records to be translated. OLDRE cannot be run interactively; the source namr must be specified.

The source file must be a type 5 (object program) and must reside on a mounted cartridge so that OLDRE can construct the scratch file on the same cartridge. If either condition is not met, OLDRE issues one of the messages

```
OLDRE: Input must be a relocatable file - type 5
OLDRE: Relocatable must be a disc file
```

and exits. OLDRE also aborts if a bad record checksum is encountered, the break flag is set, or a FMGR error occurs. The message

```
OLDRE: Converting new relocatable records to old format
```

is displayed when OLDRE is running. Any of the following messages also can be displayed to describe conditions encountered during the translation process. Each of these error conditions will result in nullifying the translation.

```
OLDRE: Symbol name truncated to 5 characters
OLDRE: Local EMA, SAVE and code area are illegal
OLDRE: New feature not available in old format
OLDRE: New feature in RPL illegal in old record
OLDRE: Cannot allocate EMA or SAVE common
OLDRE: More than 255 externals found
```

The OLDRE exit message defines the success or failure of the translation process:

```
OLDRE: End OLDRE - No errors
OLDRE: End OLDRE - Relocatable file unchanged.
```

Translation Results

The input instruction to OLDRE and the translated output are listed below.

Input	Output	Comment
NAM	NAM	
XNAM	NAM	Non-zero EMA size, SAVE size, or pure code size will cause an error. Program names longer than five characters will cause an error.
ENT	ENT	
XENT	ENT	An ENT to EMA or externals will cause an error. An entry point name longer than five characters will cause an error.
EXT	EXT	
XEXT	EXT	Weak external will be translated to a strong reference. Names longer than five characters are errors.
DBL	DBL	
XDBL	DBL	Byte externals and DDEFs to EMA will cause an error. If the memory area to be initialized is not absolute, common, or program relocatable an error will be issued.
ALLOC	ENT	Translation if ALLOC is to labeled common in FORTRAN BLOCK DATA. Any DBLS to this ALLOC have external ID number deleted.
ALLOC	EXT	Translation for all other cases except ALLOC to SAVE (illegal). Any DBLs to this external are are illegal. Only one symbol per record is legal and it must be a COMMON ALLOC. Note that only program type and name will appear in the NAM record for this module.
MSEG		Error. Will not be translated.

EMA	EMA	
DATA		Error. Will not be translated.
RPL	ENT	
END	END	
XEND	END	
GEN	GEN	
LOD	LOD	
DEBUG		Ignored. No error or record output.

Program Restrictions

The following restrictions must be observed in preparing the source programs to ensure that the relocatable code can be successfully translated.

Macro/1000

1. External, entry point, and variable names may not exceed five characters. Names exceeding five characters are truncated to characters 1 - 4 and the last character.
2. The ALLOC, DDEF, and MSEG commands may not be used.
3. The RELOC command may not be used with EMA or SAVE, or with pure code.
4. Two-word RPLs may not be used.
5. DBLs and DBRs may not refer to externals.
6. No more than 255 externals can be referenced in one module.

Pascal

1. Level 1 procedure names and program names may not exceed five characters, unless ALIASed to an external name of less than six characters. Names exceeding five characters in length are truncated to characters 1 - 4 and the last character.

2. No more than 255 level 1 procedures may be contained in one program.

FORTRAN

1. The SAVE command may not be used.

2. Subroutine, function, program or variable names may not exceed five characters. Names exceeding five characters are truncated to characters 1 - 4 and the last character.

3. Local EMA may not be specified.

4. Labeled EMA common may not be used.

5. The ALIAS feature may not be used.

6. Character data may not be in labeled common.

7. No more than 255 common block names, external subroutines, or function calls may be contained in one module.

Chapter 4
File Copy Utility

Introduction

The file backup utility FC provides a means for copying files between disc cartridges and tape media, either disc-to-disc, disc-to-tape, or tape-to-disc (tape-to-tape copying is not supported). Files can be copied to and from 800/1600-bpi magnetic tapes (mag tapes) or CS80 Cartridge Tape Drive (CTD) tapes.

Files being copied may be given a different name, security code, or cartridge by specifying those fields of the destination namr. When copying from disc, file extents are automatically gathered and copied in ascending order following the main extent. As an option, you can eliminate extents and copy all sections of a file to the main extent. Options also can be selected to purge source disc files after copying, to list or suppress the listing of files copied, to replace duplicate files with the last duplicate copied, or to verify the copy. This verification is in addition to tape checksums, which are always used.

Running FC

FC can be run interactively from your terminal, by specifying a command in the runstring, or by using the TRansfer command in the runstring to specify a control file.

When run interactively, FC continues to prompt for commands until you enter the exit or abort command, as

```
:RU,FC
 FC.nn: <command>
 FC.nn: <command>
   .
   .
   .
 FC.nn: EX
```

The .nn in the utility prompt identifies your FMGR session. Only the first two characters are required to identify a command, and FC recognizes commands entered as either upper case or lower case characters. Note that a comment can be entered at the end of a command-string, or by entering an asterisk (*) as the first character in the response. If a comment is entered after the command-string and all available optional parameters for the command are not used, commas must be inserted at the end of the string as place holders for the omitted parameters.

When a command is given in the runstring, as

```
:RU,FC,<command><,parameters>
```

FC executes the command and terminates.

When a TR command is given in in the runstring, as

```
:RU,FC,TR,<control file>
```

where <control file> can be a device LU or namr, FC executes each of the commands contained in the control file and then terminates.

FC Commands

FC incorporates command sets that allow you to configure the copy operation and specify names for the formatted tape files and comment files, to selectively copy files and group copy commands, to transfer to and return from command files, and to direct listings to a log or list device. The command sets are summarized in Table 4-1 and described by functional groups in the following subsections.

Table 4-1. Summary of FC Commands

Command	Function
Information Commands	
?	Provide summary of FC commands and command syntaxes.
HELP	Define FMGR errors encountered during FC execution.
Configuration Commands	
LL	Set list device to be used by listing commands.
TITLE	Establish title to be used in header file.
CF	Establish namr for tape comment file.
ECHO	Turn ON or OFF echoing of commands to list device.
SCRATCH	Specify disc cartridge for internal scratch files.
Copy and Related Commands	
COPY	Initiate copy operation as specified.
DEFAULT	Set default source, destination, and options for subsequent COPY commands.
GROUP EG AG	Used for grouping more than one copy command into a single COPY operation.
Listing Commands	
DL	Compile directory list of FC tape.
CL	List FMP cartridge list, or cartridges on FC tape.
CLAL	List global FMP cartridge list.
LC	List comment file from FC tape.
LH	List header file from FC tape.
Transfer and Exit Commands	
TR	Transfer to/from FC command file.
EX	Exit FC. If a group copy is active, it is processed before FC aborts.
AB	Abort FC, including any active group copy.
Comment Command	
*	Identifies following string as comment line.

Information Commands

FC incorporates a help function to provide you with a brief summary of available commands and to provide information on FMGR errors encountered during execution.

Command Summary Function

Entering ? as the runstring command causes the command summary to be written to the log device, as

```
--------- FC commands ------- commands may be abbreviated to 2 chars ---------
COPY,srce,dest,opts,[file1],[file2],[msc]    copy files
DEFAULT , srce , dest , opts                 set defaults for COPY command
GROUP / EG / AG            begin / end / abort GROUP of COPY commands
LL , namr                  set list file/device (dash means log device)
DL , srce , [msc] , opts   list tape directory (srce = -tlu or -tlu{namr})
CL , [-tlu] , options      list local cartridge list or tape cartridge list
CLAL                       list global cartridge list (RTE-IVB/6 only)
LH , -tlu , opts           list tape header file
LC , -tlu , opts           list tape comment file
ECHO [, ON/OFF]            turn ON/OFF cmd echo to list device (default ON)
TITLE , title              set tape title (for subsequent COPYs to tape)
CF , comment-file-namr     set comment file (for subsequent COPYs to tape)
TR , namr                  transfer to cmd file/device (dash means log device)
TR                         return from command file/device
EXIT                       exit FC
ABORT                      abort FC (same as EX except if copy group active)
SCRATCH , cartridge        set cartridge that FC will use for scratch files
* comment                  command line starting with * treated as comment
HELP [, key [, lu]]        get help, RTE-IVB/6 only (useful for FMGR errors)
? , <command>              list info about particular command (incl. options)
? , <option>               list info about particular option (all commands)
```

If you enter the ? command together with a command or option (as ?,SCRATCH), FC will respond with further information about the selected command/option.

FMGR Error Help Function

You also can call HELP, without exiting FC,to define any FMGR errors encountered in execution of the utility, as

```
HE[,key[,LU]]
```

As an example,

```
.
.
.
FMGR-032
FC.nn: HE
FMGR-032
CARTRIDGE NOT FOUND
AN ATTEMPT WAS MADE TO ACCESS A CARTRIDGE THAT CANNOT BE
FOUND IN THE CARTRIDGE LIST.  CHECK THE CARTRIDGE NUMBER
FOR CORRECTNESS
FC.nn:
```

Configuration Commands

The configuration commands set attributes that affect the execution of other FC commands. They allow you to select the list device, to enable/disable command echoing to the list device, to set the title and comment file for tapes subsequently written, and to set the scratch-file disc cartridge.

List Device: LL

Initially, the list device for FC is your terminal. With the LL command, you can specify another device or file to receive the list information. The command string for LL takes the form

```
LL,namr
 or
LL,-
```

where

namr is the namr of the device or file to receive the list information. If a file namr is specified, it will be created if it does not exist. Any data in an existing file will be overwritten with the new information.

- resets the list device to your terminal - used when a preceding LL command has named another list device.

The cartridge and tape directories, header files, comment files, "?" command responses and command echoes are listed under control of the LL command. Copy status messages and error messages are always displayed on your terminal.

Title Command: TI

The TI command establishes the title to be included in the header file of tapes created by subsequent copy commands. Each TI command title replaces the preceding title. The TI command takes the form

```
TI,title
```

Enter the title exactly as it is to appear in the header; lower-case characters are not converted to upper case. The title can be any ASCII characters up to a maximum of 72 characters. Titles longer than 72 characters are truncated to 72 characters.

Name Comment File: CF

The CF command establishes the namr of the optional comment file to be copied from disc to tapes created by subsequent copy commands. When the copy is to mag tape, the file is copied uncoded. Because CTD tapes are formatted in fixed-length blocks, files must be coded for copy to CTD devices. Comment-file records longer than 128 words are truncated. Note that a comment file should not contain zero-length records, or a checksum error could result when the file is listed using the LC command. The CF command takes the form

```
CF,namr
```

Because the comment file is not encoded on mag tapes, the file can be read using the FMGR ST, LI or DU commands.

Scratch Area Definition: SC

Normally, FC builds each of its internal scratch files on the first cardridge with sufficient space for the initial size of the file. With the SC command, you can specify the cartridge on which the files are to be created. (This command is useful as a means of avoiding Cartridge Full errors in copy operations that require large amounts of scratch-file space.) If used, this command must be given before any CO operation. The command string takes the form

```
SC,cart
```

where

cart is the CRN or -LU to be used for the scratch file.

Echo Command: EC

Initially, command echoing to the list device is OFF. The EC command allows you to specify that each command be echoed to the list device as the command is processed and, subsequently, to suppress echoing if desired. The default is ON if neither ON or OFF is specified. The command takes the form

```
EC,ON
```

or

```
EC,OFF
```

Copy and Related Commands

The COpy command initiates the copy operation from disc-to-tape, tape-to-disc, or disc-to-disc. Cartridges and tapes can be written and restored by naming the devices as the source and destination parameters. Files being copied can be selected by name, security code, cartridge, etc., as desired. Selection by name can make use of wildcard characters.

The COpy command takes the form

```
CO[,srce[,dest[,optns[,file1[,file2[,msc]]]]]]
```

where

srce selects the file or files to be copied. The srce can be a single namr, a list of namrs enclosed in braces, or a negative tape LU with an optional namr or list of namrs enclosed in braces. Each namr may take the abbreviated form of a CRN or negative disc LU. Wildcard characters are acceptable in srce namrs.

Where used, the braces {} are required as defined. If the srce is a single namr, the braces can be omitted; if, however, the srce is a list of namrs, the list must be enclosed with braces, as

```
namr                          *single-namr srce
{namr,namr...,namr}           *list of srce namrs
```

If the source is a tape LU, the braces can be omitted. If, however, it contains one or more optional namrs, the single namr or list of namrs must be enclosed with braces, as

```
-tlu
-tlu{namr}
-tlu{namr,namr...,namr}
```

dest determines whether the files selected by the source parameter being copied to disc or to tape, and allows specification of the name, security code, and cartridge for the destination files. The dest can be either a namr (for disc destinations) or a negative tape LU with an optional namr enclosed in braces. The namr may take the abbreviated form of a CRN or negative disc LU. The dest takes one of the forms

```
namr
-tlu
-tlu[{namr}]
```

Only one destination can be specified, and wildcard characters cannot be used in dest namrs. (Refer to the section Grouping CO Commands for a method to specify multiple destinations.) If the name, security code or cartridge is not specified in the destination namr, then that attribute of the destination will be the same as on the source file.

optns may include any of the following option letters:

B Do not list namrs for files being copied
F List namrs for files being copied
C Clear destination cartridge, issue prompt
! Clear destination cartridge, suppress prompt
D Replace duplicate files
E Eliminate file extents
I Ignore file data errors on tape-do-disc copy
K Keep tape on-line when finished
L Lock cartridges copy mode
O Open files copy mode
P Purge source files after successful copy
S Read single volume of multi-volume set
T Display tape length required for copy
U recover Unused space
V Verify data transfers

The options are not order-dependent; that is, they can be specified in any sequence. Refer to the subsection FC Options for a detailed description of each available option.

filel
file2 specifies optional limits on the range of files to be copied from the source. Within the range specified, files conforming to the source parameter are selected for copying to the destination. Note that disc-resident files are referenced by namr, tape-resident files are referenced by disc file reference number. (Refer to the DL command.) If only filel is specified, the range is interpreted as falling between filel and the last file on the source medium. If only file2 is specified, the range is interpreted to be all files between the first file on the source medium and file2.

If filel is specified but not found, no files are copied. If file2 is specified but not found, all conforming files between the range of filel and the end of the source medium are selected for copying to the destination.

If multiple disc cartridges are specified in the source parameter, the filel, file2 parameters are applied separately to each cartridge; for tape sources, the filel and file2 parameters apply to the entire tape or set of tapes.

msc is the system Master Security Code. The msc allows a security code to be specified in the destination even if a security code is not specified in the source parameters. This parameter also allows files to be purged using the P and D options without requiring that file security codes match. The msc is also required if the C or ! options are specified. (Refer to the section COpy Command Options for a detailed description of the available options.)

Source, Destination Parameter Considerations

When forming the source and destination parameters, the following characteristics of FC should be considered:

- A null security code field is considered to be matched by any security code. A zero in the security code field is matched only by a zero security code.
- For disc sources, a null cartridge field in the source namr is interpreted to mean the first cartridge containing a file matching the source file namr (including wildcard characters). A given namr cannot be used to select files from more than one cartridge.
- For tape sources, a null cartridge field in the source namr is interpreted to mean all cartridges on the tape.
- All files selected by the source and the file1 and file2 parameters are copied, except for type 0 files and files with "illegal names". The illegal-name files include scratch files with numeric names and special files created by system programs that contain a "+" in the file name.
- In the destination parameter, a file security code field can be specified only if the security code is specified in the source namr, or if the msc is specified.

- When the specified destination cartridge is different from that in the source, the copy operation is:

  Disc-to-disc: the file is copied to the cartridge named in the destination parameter.

  Disc-to-tape: the file is labeled on the tape with the cartridge named in the destination parameter.

  Tape-to-disc: the file is copied to the cartridge named in the destination parameter.

- When copying to tape, if the destination cartridge that will be listed in the tape directory is specified by the destination parameter, it must be specified as CRN, not -LU.

- Namrs specified with only one- or two-character file names (all other fields empty) cannot be distinguished from ASCII CRNs written in the abbreviated form (CRN only, rather than ::CRN). When applicable, the FMP cartridge list is searched first and, if a match is found, the namr is interpreted to be a CRN. If no match is found, the namr is interpreted as a file name. In some situations, it is impossible to recognize a CRN as such if it is given in the abbreviated form:

  1. If the destination parameter specifies a disc CRN that is to be automatically allocated (it does not yet exist on the cartridge list).

  2. If a CRN on a tape is specified in the source or destination parameter.

  In both of these cases, the CRN must be explicitly specified as ::CRN.

COpy Command Options

Any of the following options may be specified with the CO command. Note that the options are order-independent and may be specified in any sequence. Any intervening spaces in the option string are ignored.

Option	Function

B,F Brief, Full status display format. As each file is copied, the source namr (and sometimes the destination namr) can be displayed, if desired. If the Brief mode is selected, the namrs are displayed only when an error occurs, to identify the affected file. If the Full mode is selected, the namrs are displayed for successfully copied files as well as those with errors. (Refer to the Error Messages section for the format of the displayed namrs.) Brief mode is the default if the files are selected only by tape LU and/or disc cartridge; Full mode is the default in all other cases.

C,! Clear destination disc. When this option is specified, the destination disc is cleared before the copy operation. This option is permitted only if the destination is a disc cartridge and if the system Master Security Code (MSC) is correctly specified in the FC command-string. Under session, any value in the msc parameter will be accepted for clearing private and group cartridges; the true system Master Security Code is required only for system and non-session cartridges.

Before clearing the disc, FC issues the message

```
DO YOU REALLY WANT TO PURGE DISC LU nn, CRN nnn   ?
```

and waits for a YES or NO response. The ! works the same as the C option except that the above message is suppressed.

D Replace Duplicate files. Normally, if a duplicate file is encountered in a copy, a duplicate name error (FMGR-002) is issued and the duplicate file is not copied. When the Duplicate option is selected and a duplicate file is encountered, the original file is purged and replaced with the duplicate. Normally, the replacement will not take place unless the security code of the duplicate destination file matches that of the destination file to be replaced (according to the standard FMP definition of matching security codes). However, if the system Master Security Code (MSC) is correctly specified in a CO command that includes the D option, file security code rules are superseded and all duplicates are replaced regardless of whether the file security codes match. (The security code of the new destination file comes from the source file except when this is overridden by a security code specified in the destination parameter.)

Duplicate names can occur when more than one source cartridge is copied to a single destination cartridge. In such cases the file from the first source cartgridge normally is copied successfully and the rest are not copied because of duplicate name errors. With the D option selected, each successive duplicate replaces the previous one so that only the last duplicate file is copied.

E Eliminate extents. Selecting this option results in combining all file extents into the main extent. Extents are not eliminated in files that have missing extents (sparse files), or if the resulting main extent would exceed 16383 blocks. In these cases, a warning is issued. Be aware that if subsequent changes to a file result in the creation of a new extent, the extent created will be the same block size as the main extent.

I Ignore data errors. Normally, if a checksum or verify error occurs, the file is not copied. The I option overrides this feature on tape-to-disc copies, and the file (including data for which checksum errors were detected) is copied to the destination. Files that are copied with errors are not identified as such in the directory entry for the file; however, a message is issued to define the range of bad blocks copied. The I option is applicable only for tape-to-disc copy operations.

K Keep tape on-line. This option keeps the tape unit on-line under all circumstances. While this option is a convenience, it introduces the possibility that the tape could be overwritten by another program since the tape unit remains unlocked and on-line. (Refer to the Tape Handling section for more details.)

L,O Cartridge Lock, Open. Files can be copied to/from cartridges either by locking the cartridge (Lock mode) or by opening individual files for the copy (Open mode). Lock mode is generally faster, but excludes other programs from accessing the cartridge. (Refer to the Performance Considerations section for details.) Lock mode cannot be used if other programs have files open on the cartridge. Lock mode also should not be used if other programs might need access to files on the cartridge during the copy. Once a Lock mode copy is started, other programs are prevented from opening files on the cartridge for the duration of the copy. In Open mode, source files are opened non-exclusively and destination files are opened exclusively. Therefore even the Open mode restricts access to files, although the restrictions are less severe than those for Lock mode. If neither L or O mode is specified, the defaults are:

1. If the source name (or names) contains wildcards or is null, or if more than one COpy operation is GROUPed, a cartridge lock is attempted. If it is successful, Lock mode is used.

2. If the lock is rejected, or if there are no wildcard or null source names and the GRoup command has not been given, Open mode is selected.

P Purge source file. If the P option is specified, FC will purge the source file after it is successfully copied to the destination. To purge a file with a non-zero security code, either the security code field of the source parameter or the msc parameter must be specified correctly. If the necessary security code or msc is not specified, the file will not be purged. (This does not, however, prevent the file from being copied.) If the destination is a tape, the source files are purged after all files are copied. Thus, in the event of an error or if you abort the disc-to-tape copy in process (using the BReak command), the source files are not purged.

S Copy Single volume of multi-volume tape set. Normally, when copying from a multiple volume tape set, all volumes must be read in sequence. With the Single option you can suppress this feature and mount a single tape volume of a set. Note that files that cross volumes cannot be copied when using the S option.

T Display required Tape length. When this option is selected, FC reads the disc directory, calculates the amount of tape required to copy the specified files and issues the appropriate message.

U Recover Unused space. In order to copy data faster, FC may use more space on the destination medium than is required for the files being copied. This can only happen on disc-to-tape copies and lock mode disc-to-disc copies and usually does not result in a significant waste of space. The U option, however, can be specified to prevent any extra space being used.

V Verify transferred data integrity. The V option specifies the following data verification operations:

Disc Read - Data is read twice and the checksums are compared to verify the read.

Disc Write - Data is written to the disc and then immediately read to a second buffer. The two data buffers are compared to verify the write.

Tape Read - The checksum is calculated for each buffer read from the tape and compared to the checksum read with the data. (This tape-read verification is performed even if the V option is not selected.)

Tape Write - The tape is rewound after each volume is written, and a tape-read verification performed as above.

Default Command

The DEfault command is used with the COpy command to set default values for the source, destination, or options parameters. The default command takes the form

```
DE[,srce[,dest[,optns]]]
```

where the srce, dest and optns are as defined for the COpy command, except that a list of namrs may not be used in the srce parameter.

Fields that have been omitted in the source and destination parameters of the COpy command are replaced with the contents of the corresponding fields specified with the DEfault command. If the source and/or destination fields are specified in the COpy command, these fields override any DEfault fields specified.

When options are included in the DEfault command, these are appended to any options specified in succeeding COpy commands. As an example, a DEfault/COpy command set of the form

```
DE,,,EPV
CO,<srce>,<dest>,CDE
CO,<srce>,<dest>,D
```

would result in specifying CDEPV as the command options for the first CO command, and DEPV as the options for the second CO command. If a DEfault option conflicts with an option specified in the CO command, the default option is overridden.

Each DEfault command supersedes the preceding DEfault; therefore, defaults can be cancelled by specifying a DEfault command with all null fields.

CO Command Examples

The following are examples of the CO command formats for copying files between discs and tapes.

```
FC.nn: CO,X,-8
```

Copy cartridge X to tape LU 8.

```
FC.nn: CO,&-----::X,-8
```

Copy all files beginning with & on cartridge X to tape LU 8. The ----- characters in the name field define the "don't care" (or wildcard) characters. The null security code will match any file security code.

```
FC.nn: CO,{filea,fileb,filec},Y
```

Copy three files to cartridge Y. Note that the destination cartridge is specified as "Y" rather than "::Y". This abbreviated form can be used to specify ASCII CRNs provided it appears in a disc source or destination parameter and is a currently mounted cartridge. The abbreviated form can always be used to specify numeric CRNs, (as CRN 1000) or if the cartridge is being specified as a negative disc LU (as LU -25).

```
FC.nn: CO,{FILE1::A,FILE2::B},-8,v
```

Copy FILE1 from cartridge A and FILE2 from cartridge B to tape LU 8 and verify the data transfer. Any files existing on LU 8 will be overwritten as the specified files are copied.

```
FC.nn: CO,{::X:4,::X:5},-8
```

Copy all type 4 and type 5 files from cartridge X to tape LU 8.

FC.nn: CO,{%-----::X,&-----::X},-8

Copy files beginning with % and & from cartridge X to tape LU 8.

FC.nn: CO,-8

Restore all files from tape LU 8 to their original cartridges. The destination parameter is omitted, therefore the source namrs (from the tape) are used for the destination files. The tape directory file identifies the cartridge from which each file was copied to the tape.

FC.nn: CO,-8,X

Restore all files from tape LU 8 to cartridge X. The tape directory information identifying the cartridge of origin will be ignored and all files will be copied to cartridge X.

FC.nn: CO,-8{::X}

Restore from tape LU 8 only those files that were copied to the tape from CRN X. Because no destination cartridge is specified in the command, the files will be restored to the same CRN.

FC.nn: CO,-8{::X,::Y}

Restore from tape those files that were copied to tape from CRNs X and Y. Because no destination cartridge is specified in the command, the files will be restored to their original CRNs.

FC.nn: CO, FILE1,FILE2

Copy FILE1 to FILE2 on the same cartridge.

FC.nn: CO,A::X,B::Y

Copy file A on cartridge X to cartridge Y and name the file B.

FC.nn: CO,A:2,B:3

Copy file A to the same cartridge, name it file B, and change the security code from 2 to 3.

FC.nn: CO,::X,::Y,,FILE2,FILE6

Copy all files within the range of FILE2 and FILE6 from cartridge X to cartridge Y. If FILE2 is not found, no files will be copied. If FILE6 is not found, all files in the range of FILE2 and the end of the source medium are copied. In specifying the file1,file2 parameters, the range must be ascending on the medium; that is, a command of the form

FC.nn: CO,-8{FILEA},::Y,,FILE6,FILE2

would result in no files being copied if FILE2 precedes FILE6 on the cartridge.

FC.nn: CO,%-----::X,Y,,FILEA,%FILE6

Copy all files beginning with % within the range of FILEA and %FILE6 from cartridge X to cartridge Y.

FC.nn: CO,-8{%-----},Y,,2,6

The same as the preceding example, except that the source is a tape -LU and therefore the range of files must be specified as file reference numbers obtained from the DL command.

FC.nn: CO,-8,::X,,,5

Copy all files from file 1 through file 5 from tape LU 8 to cartridge X.

Grouping CO Commands

In copying to or from tapes, there may be applications that require specifying multiple source/destination pairs in a single copy operation. This would be true, for example, if files are to be copied to a tape and renamed, or if tape files are to be copied to discs other than the discs specified in the tape directory. The GRoup command allows you to group multiple copy commands in a single copy operation. The command takes the form

```
GROUP
CO,<parameters>
CO,<parameters>
 .
 .
 .
CO,<parameters>
EG
```

When the GRoup command is encountered, all following CO commands are executed as a single operation, when the EG (End Group) command is encountered. The GR operation can be aborted using the AG (Abort Group) command.

Where the source or destination is common to all of the grouped CO commands, the DEfault command also can be specified, as

```
DE,-8                Common source tape -LU
GR
CO,::AA,XX           Restore CRN AA from tape to CRN XX
CO,::BB,YY           Restore CRN BB from tape to CRN YY
CO,::CC,CC           Restore CRN CC from tape to itself
EG                   End group
DE                   Cancel default

DE,,-8               Common destination tape -8
GR
CO,ABCD,WXYZ         Copy file ABCD to -8 as WXYZ
CO,EFGHIJ,QRSTU      Copy file EFGHIJ to -8 as QRSTU
EG                   End group
```

Listing Commands

These commands direct listings of the FMP cartridge list, global FMP cartridge list, tape directory list, and the tape comment and header files to your terminal, a selected list device, or a file.

List Directory: DL

The DL command provides a directory list of files on an FC tape specified by the command. A sample tape directory listing is shown in Figure 4-1. (For directory listings of disc cartridges, use the FMGR DL command.) The DL command takes the form

```
DL,srce[,msc[,opts]]
```

where

srce is the source FC tape device, with optional namr, that specifies the files to be included in the list. If the namr is omitted, all files are listed. The source can only be a tape; use the FMGR DL command for listing cartridge files. If a namr is specified with the negative tape LU, the optional namr must be enclosed within braces, as

```
-tlu{namr}
```

msc is the system master security code. When the msc is specified, the file security code is contained in the directory list.

optn is one or more of the following options:

F Selects Full option, that results in including the extent size, record size, and # extents columns in the directory list.

S Selects Single option, that compiles a directory of only one volume of a multi-volume tape set.

K Keeps the tape unit on-line under all circumstances. Note that while the tape is on-line, it could be overwritten by another user program since the on-line tape is unlocked.

The LL command can be used to direct the DL listing to another list device.

```
title: SAMPLE OF TAPE DIRECTORY FORMAT
volume: 2
date and time of creation: 7:09 PM TUE.,  14 JULY, 1981
created under account name:  KURT.SYSTEM

                          extent  record  #extents  security  discfile
name     crn  LU  type     size    size    log  phy    code      ref #

JLMHI    CS   25   1        1       128     1    1      HP         1
*        CS   25   4        1               1    1                 2
'DVA32   CS   25   4       11               1    1                 3
*****************************************************************
 mounted volume starts here
*****************************************************************
'DVM33   CS   25   4       92               3    3      HP         4 CONT
*****************************************************************
 mounted volume ends here
*****************************************************************
'DVM33   CS   25   4       92               3    3      HP         4 CONT
'CM80L   CS   25   4       50               1    1                 5
**       CS   25   4        1               1    1                 6
'VERIF   CS   25   4       24               1    1                 7
'PBERS   CS   25   4      234               1    1                 8
'BKUP    CS   25   4       55               1    1                 9
&BOOTC   CS   25   4       18               1    1                10
/BIGL    CS   25   4        2               1    1                11
'DISC    CS   25   4        7               1    1                12
'R       CS   25   4        3               1    1                13
'NUMS    CS   25   4       41               1    1                14
```

NOTES:

1. title - specified using TI command
2. crn, LU - appear only if multiple cartridges are on a tape
3. extent size, record size - appear only if F option specified
4. # extents - appears only if F option specified:
   log = logical # of extents = last extent + 1
   phy = physical # of extents = total extents allocated to file
5. security code - appears only if msc specified correctly
6. "mounted volume starts/ends here" messages - appear only in multi-volume directory lists
7. CONT - appears if file is continued on next volume or is continued from previous volume

Figure 4-1. Tape Directory List Format

List Cartridges: CL and CLAL

The CL and CLAL commands function in exactly the same way as the FMGR CL and CLAL commands: CL command displays the list of all cartridges accessible to you, and CLAL command displays the list of all cartridges in the system.

Only the CL command can be used with a tape source; the CLAL command is not defined for use with tape.

When the CL command is given with a tape -LU source parameter, as

```
CL,-tlu[,K]
```

the resultant list identifies the cartridge files stored on the tape, as

```
title:  NNNNN
volume:  X
date and time of creation: hh:mm   ddd., mmmm, yyyy
created under account name: NNNNNN.NNNN

LU      CRN     LABEL     P/G/S    USER.GROUP
xx      xxx     xxxxx       x      xxxx.xxxx
```

If the cartridge was renamed as part of the copy operation, only the cartridge CRN is listed.

The K option can be specified to keep the tape drive on-line under all circumstances. Note that while the tape is on-line, it could be overwritten by another user program since the on-line tape is unlocked.

The LL command can be used to direct the CL and CLAL listings to another list device.

List Comment/Header Files: LC,LH

The LC and LH commands allow you to list the ASCII Header file, (LH command) and the Comment file (LC command) to the list device. The commands take the form

```
LC,-tlu[,K]
```

and

```
LH,-tlu[,K]
```

The LL command can be used to direct the file listing to another list device.

The K option can be specified to keep the tape drive on-line under all circumstances. Note that while the tape is on-line, it could be overwritten by another user program since the on-line tape is unlocked.

Transfer Command

FC can be run interactively from your terminal, from a command file, or as a combination of both. The TRansfer command allows you to transfer control to and return from a control file, as

TR,namr	Transfer control to the named control file
TR,-	Transfer control to the log device
TR	Transfer back to the next higher level control file when TR files are nested. When encountered in the first level command file, serves the same function as the FC EXit command.

Control files can be nested to four levels, with control passed from file level to file level using the TR,namr command. The current state of any of the configuration commands (Set List Device, Title Header File, Name Comment File, Echo Commands, Group Commands) and the default values remain in effect through all levels of the nested command files.

Tape Handling

Tapes are created by FC in a special format unique to this utility, and thus cannot be written or read using the FMGR ST or DU commands. (Refer to the RTE-6/VM Technical Specifications Reference Manual for a description of the FC tape format.)

One of the following messages

```
put volume on LU n

put volume number n on LU n
```

is issued to prompt you to mount the initial tape on the specified device, or to mount the next volume of a multi-volume set during the read or write operation. FC checks the tape and device status and, if a problem is encountered, issues the appropriate error message:

tape LU n not ready

tape LU n media not initialized

tape LU n not write enabled

Each of the messages above is followed by one of the messages

when ready to continue type GO, otherwise type BR

when ready to continue type GO, otherwise type BR or SK

In the case of the "media not initialized" error, you must initialize the CTD tape using the FORMC FO command (refer to Chapter 8).

Entering BR in response to the "when ready" prompt causes FC to terminate cleanly.

The SK option is viable only if the operation is a multi-volume read. The SK response indicates that the volume specified by "n" in the "put volume" message should be skipped. In this case, FC will repeat the "put volume" message for the next volume in the set, and will report "files lost" or "data lost" error messages for files on the SKipped volume. Refer to the section Messages Logged by FC for a description of the message text and corrective action.

FC automatically rewinds the tape at the beginning of a copy operation and writes over any existing data. For this reason, a tape created by FC cannot be modified; all files must be copied in a single operation. The LH (List Header) or DL (Directory List) commands may be used to check the current contents before copying over a previously used tape. You can write a transfer file to run the FC LH command and issue a prompt before executing the copy, similar to the following:

```
:RU,FC,LH,-8,K
:PA,,DO YOU WANT TO OVERWRITE THIS TAPE (type :,YE or :,NO)?
:IF,1G,EQ,YE,2
:IF,1G,NE,NO,-3
:TR
:RU,FC,CO,XX,-8
:TR
```

FC reads the first record of a tape volume and, if it does not conform to the FC tape format, either logs a tape format error or displays the first two records in ASCII and the message

tape not readable by FC, first two records are

FC locks the LU before a tape is accessed, to prevent other processes from accessing the tape. If the tape is write-protected, the LU is unlocked at the end of the command. If the tape is not write-protected, the LU remains locked until:

a tape command is encountered that specifies a different LU,
a timeout occurs while FC is waiting for a command, or
FC finishes executing.

To protect the tape, FC sets the device off-line before the lock is released, unless the Keep tape on-line option was specified in the last tape command, or the command was terminated due to an error or a BReak command. Under these conditions, the tape remains on-line. The CTD device is taken off-line programmatically by UNLOADing the cartridge (as if the UNLOAD button had been pressed at the front of the drive).

Since having the device taken off-line can be inconvenient, you should use the write-protect mechanism or run FC interactively to retain the lock between commands. If you use the K option to keep the tape on-line under all circumstances, be aware that the result can be an unlocked tape on-line and unprotected from access by other processes.

If a copy to tape is interrupted by a BReak command, the full tape generally will not be usable. (In multi-volume sets, of course, volumes copied prior to the BR will be intact.) If, however, the BR was issued during the verify phase, all data copied up to the point of the break will be usable.

Destination Disc Handling

On tape-to-disc copies in a session environment, automatic cartridge allocation is attempted when the destination cartridge is not already mounted and is either specified by CRN or defaulted. An attempt is made to allocate a cartridge at least as large as the data to be copied or, whenever possible, 20% larger to allow for expansion. When a cartridge is allocated, FC issues the message

allocated CRN xxxxx, LU nnn

and executes the commanded copy operation. Be aware that if a cartridge is mounted to your account but is not active in the current session, it will be allocated; thus there is the possibility of reactivating an existing cartridge rather than allocating a pool cartridge. If a cartridge cannot be allocated, FC issues the appropriate message and terminates the copy. Note that if the destination disc is specified by -LU, the disc is not automatically mounted; if the specified -LU is not mounted, the appropriate message is issued and FC exits.

If the destination cartridge was mounted, FC performs a size check on tape-to-disc copies before any files are copied. If the size check shows that there is insufficient data or directory space for the files to be stored, FC issues the appropriate error message and exits. (Note that in open mode, other processes can create files on the cartridge, thus it is possible for cartridge-full errors to occur even when the size check indicates sufficient space for the copy operation.)

For disc-to-disc copies, the size check is not performed and if a directory-full or cartridge-full error occurs, FC terminates and stores the status in the FMGR globals. (Refer to the section Error Handling for a definition of the FMGR global returns.)

Loading FC

FC must be loaded on-line since it runs as an extended background program. Load the utility using the LOADR command file #FC6 (92084-17151). After the utility is loaded, use the FMGR SP command to save FC and its segments (FC000, FC001,..., FC00n) on a cartridge available to all FC users. If necessary, use the FMGR OF command to remove any previous versions of FC and its segments. The load process also requires copying the FC help file, "FCHLP (92084-17150) to a cartridge available to all users.

Performance Considerations

The following guidelines are presented as an aid in further increasing the speed of the file copy operation.

1. Specify source cartridges explicitly if multiple cartridges are contained in the cartridge list. For example, if the cartridge list includes cartridges 10, 20 and 30 and you want to copy from cartridge 30, use a COpy command of the form

   ```
   CO,{A::30,B::30,C::30,D::30,E::30},-8
   ```

   In this way, FC will not have to search cartridges 10 and 20 for the files to be copied. You also could use the DEfault command to specify the cartridge, as

   ```
   DE,::30
   CO,{A,B,C,D,E},-8
   ```

   In the latter command form, be aware that sequences of commands cannot be put in the runstring.

2. Keep file extents collected. The speed of the FC COpy operation is increased if the file main and its extents are adjacent on the source cartridge. You can collect extents (make them adjacent) on all files on a cartridge using a COPY command of the form

   ```
   CO,100,,LDF
   ```

   which copies the contents of cartridge 100 to itself (the destination cartridge defaults to the same cartridge as the source). The L option is needed to ensure that extents created are adjacent. The D option is needed so that the new files (with extents collected) can replace the old files. The F option logs the name of each file copied.

   If a cartridge-full or directory-full error (FMGR-033 or FMGR-014) occurs on the above operation, pack the cartridge using the FMGR PK command and enter the CO command again. This may have to be repeated if the FMGR-033/-014 error recurs. If the error occurs on the same file, however, the PK will not help and a larger cartridge will be needed for the operation.

   If you have many small extents, you may want to make files sizes larger to reduce the number of extents. You can do this by copying each file using the ST command, or you can Eliminate all extents on a cartridge by using the CO command shown above with the E option added. (Note that it is not always desirable to eliminate extents because FMP will not be able to re-use the space as easily when these files are purged, and because larger file sizes tend to result in less efficient use of space.) Extent collection (and elimination, if desired) can be done as a periodic cleanup function much like a FMGR PK.

If the extent collection/elimination process affects a lot of files (as shown by the F option log), you should do a FMGR PK after the CO command.

3. Use Lock and Open modes efficiently. Determine whether Lock mode or Open mode works faster for your application and use that mode whenever possible. You can force a particular mode by specifying the L or O option in the COPY command. For most copy operations involving large numbers of files, Lock mode is more efficient since entire cartridges are locked in one operation rather than opening individual files. However, in certain cases Lock mode may introduce other overhead which could make it slower than Open mode. Open mode is usually more efficient when a small number of files is being copied. The faster mode for a given cartridge may depend on:

   The number of files being copied to or from the cartridge

   The total number of files on the source disc cartridge

   Whether the destination disc cartridge was initially empty

   The number of files purged as a result of the D option.

4. Use cartridges wisely. Copying to empty cartridges in Lock mode is faster than copying to partially filled cartridges.

Error Handling in Transfer Files

Certain error conditions can be detected in FMGR transfer files by checking the FMGR globals after running FC. If FC is scheduled by a program other than FMGR, the scheduling program can retrieve the same information as would appear in the FMGR globals by making a call to RMPAR. Since only one error can be returned in the globals, only the highest error category number (most severe error) is returned if multiple errors occur. The parameters returned by RMPAR correspond to the FMGR globals:

```
PARAM(1) = 1P or characters 1-2 of 10G (next-source-file information)
PARAM(2) = 2P or characters 3-4 of 10G
PARAM(3) = 3P or characters 5-6 of 10G
PARAM(4) = 4P (error category number)
PARAM(5) = 5P (next-source-file information status)
```

where

```
4P = 10000 - program aborted due to OF or system violation.
             Violations (such as Memory Protect) could result
             from a bad segment load, or could be an FC bug.
```

The following errors terminate FC:

= 1000 - FC internal error (possibly a bug in FC)
= 300 - transfer file stack overflow
= 200 - scratch file error (see SCRATCH command)
= 100 - segment loading error

The following errors terminate current command:

= 90 - general error causing command termination
= 80 - destination disc cartridge or directory full
= 70 - break detected (system BR command)

The following errors allow command to continue,
but purge phase is skipped on disc-to-tape copies:

= 50 - copying from current source volume aborted (tape-to-disc)
= 40 - unusual source file access error - file not selected
= 30 - general error on file being copied - file skipped
= 25 - tape format error (reading or verifying tape)

The following warnings do not affect FC operation:

= 20 - warning
= 10 - soft data error detected
= 0 - no error or warning

5P = 1 - next source file information not applicable
= 0 - no error
= -1 - next source file information valid (4P = 80)
= -2 - source is tape, number of next file greater than 32767

10G
1P = IF 5P = -1 (4P = 80):

if source is disc:
10G = name of next file to be copied from disc

if source is tape:
1P = ref # of next disc file to be copied from tape

IF 5P = 1 and 4P not 10000, or 5P = -2, or 5P = 0:

10G = ASCII nulls
1P = 0

IF 4P = 10000 (5P = 1):

10G and 1P undefined

In some circumstances the meaning of the FMGR global information may be ambiguous. Since only the most severe error is posted to the FMGR globals, it is not possible to determine whether other errors occurred, or to determine the command on which an error occurred if multiple commands were given. Further, if a copy command has more than one source cartridge, you cannot determine which cartridge contains the "next source file". If the copy command grouping feature is used, additional ambiguity can result.

Because of these possible ambiquities, the use of FMGR globals to determine the "next source file" is intended only for command files containing a single COpy command and involving a single source cartridge.

Using Globals In Transfer Files

The information in the FMGR globals can be used to advantage in writing transfer files to copy one source cartridge to a series of destination cartridges, moving to the next cartridge when the current cartridge is full. Such a file is particularly useful for backing up large fixed discs to a series of flexible discs.

The file takes advantage of the fact that the CO command filel parameter specifies the first file to be copied, and the fact that the 10G global contains the name of the next file to be copied when a copy terminates due to a cartridge-full error. A series of COpies in which the ASCII contents of 10G is used as the filel parameter will allow each COPY to continue where the previous one stopped.

The following is an example of a transfer file for this purpose. The source cartridge is specified in the 1G global, the negative flexible disc LU is specified in the 2G global, and the system master security code (MSC) is specified in the 3G global. The MSC is required for the ! option, which is specified to clear the destination disc without the need for user intervention.

```
:PA,,Put in floppy and type TR
:MC,2G
:RU,FC,CO,1G,2G,!VF,,,3G
:DC,2G
:IF,5P,NE,-1,5
:PA,,Put in next floppy and type TR
:MC,2G
:RU,FC,CO,1G,2G,!VF,10G,,3G
:DC,2G
:IF,5P,EQ,-1,-5
:TR
```

Assuming a transfer file name FLOPUP, source CRN 100, destination LU -10, and MSC XX, the following FMGR TR command will cause CRN 100 to be backed up to a series of flexible discs on LU -10.

```
TR,FLOPUP,100,-10,XX
```

If CRN 100 contains 25 files, and 10 of these fit on the first flexible disc, 10 on the second, and five remain for the third, the process would be

```
:TR,FLOPUP,100,-10,XX
 :PA,,Put in floppy and type TR
:TR
 :MC,-10
 :RU,FC,CO,100,-10,!VF,,,XX
FILE1
  .
  .                                        First ten files copied
  .
FILE11
FMGR-033                                   Cartridge-full error
copy terminated
 :DC,-10
 :IF,-1,NE,-1,5
 :PA,,Put in next floppy and type TR
:TR
 :MC,-10
 :RU,FC,CO,100,-10,!VF,FILE11,,XX          Next file namr in 10G
FILE11
  .
  .                                        Next ten files copied
  .
FILE21
FMGR-033                                   Cartridge-full error
copy terminated
 :DC,-10
 :IF,-1,EQ,-1,-5
 :PA,,Put in next floppy and type TR
:TR
 :MC,-10
 :RU,FC,CO,100,-10,!VF,FILE21,,XX          Next file namr in 10G
FILE21
  .
  .                                        Copy remaining files
  .
FILE25
 :DC,-10
 :IF,0,EQ,-1,-5                            Test for cartridge-full
 :TR                                       error
:                                          Copy complete; exit
```

Messages Logged by FC

This section contains a list of all the messages that can be generated by FC. In the listing, brackets [] surround those parts of a message that are included only under certain conditions. Angle brackets <> indicate variables that are replaced with the appropriate value when the message is issued. Some message reference portions of the FC tape format; refer to the RTE-6/VM Technical Specifications Reference Manual for a complete description of the tape format.

Errors Requiring Operator Action/Response

```
DO YOU REALLY WANT TO PURGE DISC LU <LU>, CRN <CRN> ?
```

A COPY command specified the 'C' option in the options parameter to Clear the destination cartridge of all current data. FC requires a response to this prompt to be sure that you really want to destroy the data on the named LU or CRN. Use the '!' option instead of the 'C' option to clear the destination cartridge without the warning prompt.

```
LU <LU> is down
correct problem and UP device, or use BR to break FC
```

Prior messages may help to indicate the nature of the problem. Once the situation has been corrected, bring the device up using the system UP command, at which point FC will continue. If the problem was not correctly solved before the UP command was issued, this message will occur again. If the problem cannot be corrected, issue the system BR command to signal FC to terminate the current command.

```
tape LU <LU> not ready
when ready to continue type GO, otherwise type BR [or SK]
```

This error usually means that the mag tape device is not on-line or that the CTD cartridge is not loaded. Be sure that the correct LU was specified in the command. (Refer to the section on Tape Handling for an explanation of the GO, BR, and SK responses.)

```
tape LU <LU> media not initialized
when ready to continue type GO, otherwise type BR
```

An attempt was made to write to an uninitialized CTD cartridge. Initialize the cartridge using the FORMC FOrmat command (Chapter 8). Refer to the section on Tape Handling for an explanation of the GO and BR responses.

```
tape LU <LU> not write enabled
when ready to continue type GO, otherwise type BR
```

For mag tape, the write-protect ring should be installed to allow writing to the tape. For CTD, the protect switch on the cartridge should be changed so that the arrow points away from the direction labelled 'SAFE'. (Refer to the section on Tape Handling for an explanation of the GO and BR responses.)

```
put volume [number <n>] on LU <tape LU>
when ready to continue type GO, otherwise type BR [or SK]
```

The required tape should be mounted and on-line, or loaded. (Refer to the section on Tape Handling for an explanation of the GO, BR, and SK responses.)

```
waiting to lock list device
```

FC attempted to lock the list device and got an indication that either another program has the list device locked, or a resource number for the lock is not available. In either case FC waits, retrying the lock every 2 seconds. While operator action is not required, you can issue the system BR command at this point to abort the command rather than waiting for the device or resource to become available.

Information Messages and Warnings

```
<source namr>  [<dest namr>]
```

When a namr or a pair of namrs is logged with no other message, it means that the indicated file was successfully copied from the source to the destination. When only a single namr is logged, it is the namr for the source file. When two namrs are logged, they are for the source and destination files, respectively. (The namrs may consist of the file name only, or may include the CRN.) The logging of namrs for successfully copied files is done in Full mode only. (Refer to the description of the COpy command B and F options for more information.)

tape needed: <n> feet if 800 BPI, <n> feet if 1600 BPI

This message is produced on a COPY from disc to 800/1600 BPI mag tape if the T option is specified.

tape needed: <n> CTD blocks

This message is produced on a COPY from disc to CTD if the T option is specified.

scanning directories

Indicates that FC is searching the cartridge directories to determine what files are being copied to tape.

copying files

Indicates that FC is about to begin copying files to tape (assuming that the tape unit is ready).

cleaning up

Indicates that FC has finished copying files to tape, and is now either closing source files or unlocking source cartridges. This process may take some time, and should be allowed to complete.

verifying volume

Indicates that FC has finished writing to the current volume and is now reading the volume in order to verify it.

break acknowledged

This indicates that FC has detected a system BR command, and is in the process of cleanly terminating the current command.

writing tape at: <system time string>

This indicates the date and time that will be written in the tape header for this disc-to-tape copy.

beginning group

A GROUP command has been entered, and subsequent COPY commands will be saved and executed together once the EG (END GROUP) command is entered.

group aborted

The AG (ABORT GROUP) or AB (ABORT FC) command was entered, causing all COPY commands since the GROUP command to be ignored.

allocated CRN <CRN>, LU <LU>

On a tape-to-disc copy, a required CRN was not found in the cartridge list so it was allocated from the disc pool.

device is up, FC continuing

The device is now UP, and FC is attempting to continue the operation.

FC xxxxx-xxxxx REV.xxxx <xxxxxx.xxxx>
Use ? for help.

This message appears before the first prompt is issued in interactive mode. The first line indicates the FC part number, the revision code, and the date/time code for the last revision.

(timeout)

FC detected a terminal timeout while waiting for input from the log device. If a tape LU was locked, the lock is released, which may require that the tape unit be taken off-line. Refer to the section on Tape Handling for details.

copy terminated

This message appears when a copy operation is terminated before completion for any reason, which could be an error or an operator break.

title: <title>

This message is logged to verify the title entered in the TITLE command.

warning: title truncated
title: <title>

The TITLE command specified a title that was too long, causing the title to be truncated. The truncated title is logged.

warning: no match for:[namr = <namr>][file1 = <?>][file2 = <?>]

No files were found to match the combined restrictions of the source namr and the file1 and file2 parameters. Any combination of the three items may be listed, depending on what was specified. Note that this does not mean that no match was found for the file1 or file2 parameter itself, but rather that no files within the specified file1/file2 range were found to match the namr.

warning: unable to eliminate extents in one or more files

The E option was specified, but extents could not be eliminated from some files. See the description of the E option for details.

no files selected

The COPY command had no effect because no source files were selected.

disc write required retries:
LU <LU> trk <n> sec <n> thru trk <n> sec <n>

The indicated disc write was unsuccessful on the first try, but succeeded after it was retried one or more times. The failure could be due to an error reported by the disc driver, or could be due to a verify error if the V option was specified.

disc read required retries:
LU <LU> trk <n> sec <n> thru trk <n> sec <n>

Same as above but for disc read.

disc directory write required retries:
LU <LU> dir tr <n> thru dir tr <n>

Same as above but for disc write in the directory area. The location of the read is isolated to a particular track or range of tracks. The track numbers indicated after 'dir tr' are the directory track numbers, not the absolute track numbers on the disc LU. The first directory track (the one with the greatest absolute track number) would be indicated as 'dir tr 1'.

disc directory read required retries:
LU <LU> dir tr <n> thru dir tr <n>

Same as above but for disc read in the directory area.

tape format error <positive error code>

This error indicates that the data read from tape deviated from the expected format in some way, but copying from the tape could continue. The deviation is usually the result of some non-fatal tape read error. This message does not necessarily imply any loss of data; other errors also will occur if any data is lost. This message is provided mainly for diagnostic use by HP. This error is to be contrasted with a tape format error that has a negative error code, covered under 'Errors that Result in Rejecting Current Source Tape Volume'.

bad tape cartridge entry is:
<octal dump>

This message may accompany a tape format error message and is for diagnostic use by HP.

bad tape discfile entry is:
<octal dump>

Same as above.

bad chunk header is:
<chunk size> <chunk type> <#entries> <lst file # or 0>

Same as above.

bad microdirectory entry is:
<file#> <file blk> <#blks> <buffer block>

Same as above.

Disc Data I/O Errors

disc write failed: LU <LU> trk <n> sec <n> thru trk <n> sec <n>

A disc write failed, even after retries were done. The failure could be due to an error reported by the disc driver, or could be due to a verify error if the V option was specified. FC will also report a disc error (FMGR-001) for each file affected.

disc read failed: LU <LU> trk <n> sec <n> thru trk <n> sec <n>

A disc read failed, even after retries. The failure could be due to an error reported by the disc driver, or could be due to a verify error if the V option was specified.

FC will also report a disc error (FMGR-001) for each file affected. On disc-to-disc copies, the affected files are not copied. On disc-to-tape copies, the affected files are copied, but the affected parts of the files are flagged to indicate the disc error so that an appropriate error message can be given when the tape is read.

Non-Fatal Tape Read Errors

tape read error

The driver indicated an unrecoverable error on the tape read. FC

does no retries in this case because the driver would have retried if appropriate.

checksum error (chunk header)

The driver indicated a successful read, but FC's checksum information indicated a problem. FC attempts retries to recover from this. If the retries fail, FC considers the information to be lost.

checksum error (chunk body)

The driver indicated a successful read, but FC's checksum information indicated a problem. FC attempts retries to recover from this. If the retries fail, FC considers the information to be lost, unless the I option was specified. In this case, FC may attempt to use the information in spite of the checksum error, depending on the context in which the error occurs. Appropriate warnings are given in this case. See the 'data lost' messages below for more information.

bad record length

The driver indicated a successful read (from mag tape), but the record length indicated by the transmission log returned from the driver disagreed with the length value specified in the header field of the record. FC attempts retries to recover from this. If the retries fail, FC considers the information to be lost.

attempting to use data in spite of checksum error

See information above for 'checksum error (chunk body)'.

retrying tape read

A 'checksum error ...' or 'bad record length' error has occurred, and FC is attempting to recover from the error by retrying the tape read.

Errors Affecting Single File

can't open <name>::<CRN> due to error:
FMGR <nnn>

The indicated source file could not be opened to copy. This is usually because the file is open exclusively to another program (FMGR-008), or because the cartridge is locked to another program (FMGR-013).

```
file <name>::<CRN>
purged or replaced during copy or directory entry corrupt
```

The indicated file was purged and possibly replaced by a different file during the time FC was scanning the directory, or else a directory entry for the indicated file is corrupt. The file is not copied.

```
<source namr>  [<dest namr>]
FMGR <nnn>
```

When a namr or a pair of namrs is logged, followed by an FMGR error code, it means that the indicated file was not copied because of the FMGR error, except for FMGR-001 errors on disc-to-tape copies. For the FMGR-001 error, the file is copied to tape but the affected parts of the file are flagged to indicate the disc error, so that an appropriate error message can be given when the tape is read.

When only a single namr is logged, it is the namr for the source file. When a pair of namrs are logged, they are for the source and destination files, respectively. The namrs may consist of the file name only, or may also include the CRN.

```
<source namr>  [<dest namr>]
FMGR <nnn>
warning:  above file copied but not purged from source
```

Similar to the above error, except that the FMGR error only prevented the source file from being purged as requested by the P option, and did not prevent the file from being copied.

```
<source namr>  [<dest namr>]
data lost
```

Similar to the above error, except that the 'data lost' error occurred rather than a FMGR error. Data lost errors can only occur on tape-to-disc copies, and indicate that data was lost for the file due to some problem reading the source tape. Problems reading the tape may have been identified in earlier messages (see NON-FATAL TAPE READ ERRORS) or a record could have been dropped, in which case no problem would have been detected other than the data lost error itself.

For data lost errors, it is possible that other messages may intervene between the line with the namrs and the data lost message. The data lost error applies to the most recently logged file namrs even if the namrs do not immediately precede the data lost message.

If the message 'checksum error (chunk body)' appears anywhere before a 'data lost' message, it may be possible to recover some of the lost data by repeating the tape-to-disc copy with the I option specified. When the I option is specified, destination files are created even though some or all of the data was lost, and 'data lost' messages are replaced by some of the more specific messages below, which indicate the exact part of the file affected and whether the data loss is certain or not.

```
<source namr>  [<dest namr>]
data lost due to disc error when tape was made
```

Same as above except the data was lost because of a disc error that occurred when the tape was written, not because of a problem reading the tape. A FMGR-001 (disc error) occurred when this file was being copied to tape.

```
<source namr>  [<dest namr>]
data lost: entire file
```

Same as 'data lost', except this message occurs when the I option was used to attempt to recover part of the file, but the attempt did not succeed. Because the I option was specified, the destination file is created even though all of its data was lost. The resulting destination file is of no value, except for any identifying information that might be obtained from the directory entry.

```
<source namr>  [<dest namr>]
data lost:
[xtnt <xl>] blk <bl> thru [xtnt <xn>] blk <bn>
```

Same as above, except that only part of the file was lost. The message indicates the range of blocks that were lost. The first and last block in the range are specified by giving the extent number (omitted if the block is in the main extent) and the relative block number within the main or extent. Blocks are numbered starting at zero within the main or extent.

More than one of these messages may be given for a single file. The file namrs are not repeated for subsequent data lost messages on the same file, even though it is possible that other messages may intervene, breaking up the sequence of data lost messages.

Because the I option was specified, the destination file is created even though some or all of its data was lost.

<source namr> [<dest namr>]
possible data loss: entire file

Same as 'data lost: entire file' except that it is uncertain whether any data was lost. The file should be examined to determine the extent of the loss, if any.

<source namr> [<dest namr>]
possible data loss:
[xtnt <x1>] blk <b1> thru [xtnt <xn>] blk <bn>

Same as above, except that only part of the file might have been lost. The message indicates the range of blocks that might have been lost. The first and last block in the range are specified by giving the extent number (omitted if the block is in the main extent) and the relative block number within the main or extent. Blocks are numbered starting at zero within the main or extent.

More than one of these messages may be given for a single file. The file namrs are not repeated for subsequent data lost messages on the same file, even though it is possible that other messages may intervene, breaking up the sequence of data lost messages.

<source namr> [<dest namr>]
data lost due to disc error when tape was made: entire file

Same as 'data lost due to disc error when tape was made' except this message occurs when the I option was used to attempt to recover part of the file, but the attempt did not succeed.

Because the I option was specified, the destination file is created even though all of its data was lost. The resulting destination file is of no value, except for any identifying information that might be obtained from the directory entry.

<source namr> [<dest namr>]
data lost due to disc error when tape was made:
[xtnt <x1>] blk <b1> thru [xtnt <xn>] blk <bn>

Same as above, except that only part of the file was lost. The message indicates the range of blocks that were lost. The first and last block in the range are specified by giving the extent number (omitted if the block is in the main extent) and the relative block number within the main or extent. Blocks are numbered starting at zero within the main or extent.

More than one of these messages may be given for a single file. The file namrs are not repeated for subsequent data lost messages on the same file, even though it is possible that other messages may intervene, breaking up the sequence of data lost messages. It is likely that this message may appear repeatedly for different parts of the same file, even if the different parts are adjacent, and even if the net meaning of all the messages is that the whole file was lost.

Because the I option was specified, the destination file is created even though some or all of its data may have been lost.

source file <name>::<CRN>
selected by commands with conflicting parameters

On a tape-to-disc copy specified by a GROUP with more than one COPY command, the indicated source file was selected by more than one COPY command but those particular COPY commands did not meet the following restrictions.

1. The destination parameters must be the same in all the commands.

2. The D option must be consistently selected or not selected in all the commands.

3. The E option must be consistently selected or not selected in all the commands.

4. The msc parameter must be consistent through all the commands.

5. It is not permissible for the L option to be specified in some of the commands and the O option in others.

Loss of Unidentified Files on Copy From Tape

files lost, reference numbers <n1> thru <n2>, names not available

Indicates that distributed directory information was lost for files with reference numbers n1 through n2. Because the directory information was lost, the names and other information about the files is not known. The main tape directory can be listed with the DL command to determine what files these numbers correspond to.

Disc-to-Tape Copy Verify Errors

files lost, reference numbers <n1> thru <n2>

Indicates that distributed directory information was lost for files with reference numbers n1 through n2. If necessary, the main tape directory can be listed with the DL command to determine what files these numbers correspond to.

data lost for file, reference number = <n>

Indicates that data was lost for file reference number n. If necessary the main tape directory can be listed with the DL command to determine what file this number corresponds to.

Errors That Cause Rejection of Current Source Tape

tape not readable by FC, first two records are:
<record 1>
<record 2>

The tape mounted is apparently not an FC tape. The first two records are logged to help in determining just what the tape is.

volume does not match others

On a multi-volume tape-to-disc copy, the tape volume just mounted does not come from the same set as the previous volumes.

wrong volume

The tape volume just mounted does not have the correct volume number. Volumes in a multi-volume tape set must be restored in the correct sequence. When FC prompts for a new volume, it indicates which volume number is to be mounted.

fatal tape read error

The tape driver reported an error while the header or comment file was being read. FC makes no attempt to recover from this, however you can retry the operation manually. On multi-volume tape-to-disc copies this does not involve starting over with the first volume, but merely requires responding 'GO' to the prompt to mount the volume. Refer to the section on Tape Handling for an explanation of the options available at this point.

header file checksum error

The information in the header file is questionable due to a checksum error. It would be unwise for FC to continue reading the volume in this situation. As with the fatal tape read error, you can retry the operation manually.

tape format error <negative error code>

This error indicates that the data read from tape deviated from the expected format so severely that FC is unable to continue reading the volume. For CTD this usually means that the tape is blank or is not an FC tape; for other tapes, it is usually the result of some non-fatal tape read error. The negative error code is for diagnostic use by HP. As with the fatal tape read error, you can retry the operation manually.

This error is to be contrasted with a tape format error that has a positive error code, covered under the heading INFORMATION AND WARNING MESSAGES.

Command Syntax, Parameter Errors (Command is Skipped)

error: no such command (use ? for help)

Self-explanatory.

error: bad or missing parameter

Self-explanatory.

error: braces used where not permitted

Self-explanatory.

error: bad namr

An incorrectly formed namr appeared in the command.

error: mismatched braces

Self-explanatory.

error: namr list not allowed in this context

A list of namrs delimited by braces was used in a parameter where a list of namrs is not permitted.

error: unrecognized option character

An unrecognized character appeared in the 'options' parameter.

error: incompatible options

Contradictory options, such as B and F, or L and O, were specified in the same command.

error: option not defined for this command

An option character was specifed in the 'options' parameter which is not meaningful for the this command.

error: option not applicable

An option character was specified in a COPY command which is not applicable for this type of copy (disc-to-disc, disc-to-tape, or tape-to-disc).

error: bad file1 or file2 param

The value for the file1 or file2 parameter was incorrectly specified. These parameters specify file names when copying from disc, and file numbers when copying from tape.

error: bad msc parameter

Self-explanatory.

error: no tape-to-tape copy

Both the source and destination parameter specified a tape device, implying a tape-to-tape copy, which is not supported. Make sure the DEFAULT command hasn't been used incorrectly.

error: source or destination incompatible with group

The source and destination parameters must be consistently either a disc or a tape throughout the GROUP, and only one tape LU can be specified in a given group.

error: options inconsistent with group

Within a GROUP, certain options may not be specified in one copy command but omitted from another. These options are the V, I, S, F, B, T, and K options. The DEFAULT command may be useful in ensuring that these options are used consistently in all COPY commands within a GROUP.

error: L and O options must be consistent for a given cartridge

The L and O options must be consistent for a given disc cartridge within a GROUP. To be consistent, both the L and O options may not be specified for the same cartridge, even in different COPY commands within a group. It is permissible for the L or O option to be specified in one command while another command specifies neither option. (The command which specified neither option will be treated as if the same option had been specified.)

error: renaming multiple files to same name, or cartridge not found

The source name was omitted, or has wildcards, or the source is a list of namrs but the destination specifies a file name. A destination file name is permitted only if it is clear that the source parameter can select only one file.

Another possibility is that the destination parameter was intended to specify an ASCII CRN in the abbreviated form (CRN rather than ::CRN) but the ASCII CRN was interpreted as a file name because no cartridge with that CRN was found on the cartridge list.

error: bad destination name

An illegal file name was specified in the destination parameter.

error: on disc-to-tape copy,
dest cartridge may not be specified as -LU

If a cartridge is specified in the destination parameter on a disc-to-tape copy, it must be specified as a positive CRN, not a negative LU.

error: P option requires explicit source cartridge

The P option may not be specified without also specifying the source disc cartridge explicitly in the source parameter.

error: L and O options require explicit source cartridge

The L or O option may not be specified without also specifying the source disc cartridge explicitly in the source parameter.

error: P option not allowed if source and dest are same cartridge

The P option may not be specified on a disc-to-disc copy in which the source and destination are on the same cartridge.

error: C and ! options require explicit destination cartridge

The C or ! option may not be specified without also specifying the destination disc cartridge explicitly in the destination parameter.

error: dest secu code not allowed w/o source secu code or good msc

A security code was specified in the destination parameter, but no security code was specified in the source parameter and the msc parameter was not specified correctly.

error: bad LU or device not supported

An LU specified in the command is not the LU of a disc or tape device supported by FC. Check the Session Switch Table.

error: bad tape LU

The tape LU specified in the command is not the LU of a tape device supported by FC. Check the Session Switch Table.

Command Out-of-Sequence Errors (Command is Skipped)

error: already in group

A GROUP command was entered before the previous GROUP was ended.

error: not in group

An EG (END GROUP) or AG (ABORT GROUP) command was entered, but there was no previous GROUP to end or abort.

Other Errors That Terminate Current Command

error on help file "FCHLP:
FMGR <nnn>

The indicated error occurred when attempting to access FC's help file, "FCHLP, needed for the ? command.

error allocating CRN <CRN>:
FMGR <nnn>

On a tape-to-disc copy a required CRN was not found in the cartridge list, so FC attempted to allocate the CRN from the disc pool, but the allocation failed due to the indicated FMGR error.

```
cartridge lock error on disc LU <LU>:
FMGR <nnn>
```

FC was unable to lock the indicated disc cartridge due to the FMGR error. If the error is FMGR-103 (CORRUPT DIRECTORY DETECTED), resolve the problem before trying to access any files on the cartridge, otherwise unnecessary loss of data could result.

```
cartridge unlock error on disc LU <LU>:
FMGR <nnn>
```

Same as above, but for cartridge unlock.

```
cartridge clearing error on disc LU <LU>:
FMGR <nnn>
```

Same as above, but for clearing the directory on the disc cartridge.

```
error in cartridge status on disc LU <LU>:
FMGR <nnn>
```

Same as above, but more general.

```
disc directory write failed:  LU <LU>  dir tr <n> [thru dir tr <n>]
error writing directory on disc LU <LU>:
FMGR <nnn>
```

A directory write to the specified track or tracks failed due to a disc error (FMGR-001) or verify error (FMGR-049), even after retries were attempted. The location of the write is isolated to a particular track or range of tracks. The track numbers indicated after 'dir tr' are the directory track numbers, not the absolute track numbers on the disc LU. The first directory track (the one with the greatest absolute track number) would be indicated as 'dir tr 1'.

```
disc directory read failed:  LU <LU>  dir tr <n> [thru dir tr <n>]
error reading directory on disc LU <LU>:
FMGR <nnn>
```

Same as above but for directory read instead of write.

```
comment file error:
FMGR <nnn>
```

The indicated FMGR error occurred when the comment file (determined by the most recent CF command) was opened or read.

error: number of cartridges to be copied to tape exceeds limit

Cartridges were renamed on a copy to tape with the result that the tape's cartridge list would contain more than 64 distinct cartridges.

error: volume not big enough for header/comment/directory files

The destination tape is too small to be used. There is not enough room for the minimum amount of information that must be written before going on to another volume.

error: fatal i/o error on LU <LU>, status = <octal A-Register status>

FC attempted I/O to the specified LU and the status returned in the A-Register indicates a problem that should never occur. The status value in the message is the value returned in the A-Register from the EXEC call.

error: comment file checksum error

A checksum error was detected when reading the comment file from the tape.

error: not enough memory

This message indicates that FC does not have enough free memory to allow the command to proceed. If FC was sized to less than 32 pages, increasing the size may solve the problem. If the GROUP command was used, fewer COPY commands in the GROUP may prevent this problem.

error: tape lock failed

FC is unable to lock the tape LU required by the command. Either the tape LU is locked to another program or no resource number is available for the lock.

error: tape DESCRIBE error

The CTD driver reported an error when FC issued a 'DESCRIBE' control request.

error: error on tape write

An unrecoverable error was reported by the driver on a tape write request. FC does not retry in this case because the driver would have retried if appropriate. FC can not continue copying to tape after this because the resulting tape would be corrupt.

HELP scheduling error: <4 char system error mnemonic>

The HE command was unable to schedule the HELP program due to the indicated system error.

CRN <CRN> not found

In a non-session user environment, a tape-to-disc COPY command with an unspecified destination cartridge was issued, and the required CRN--determined by reading the tape directory--was not found. This error is not possible in a session environment, because the required cartridge could be allocated from the pool.

CRN <CRN> LU <LU> not large enough:
data blocks/dir entries needed: <bl>/<ent>, available: <bl>/<ent>

The specified cartridge does not have enough file or directory space for the files to be copied from the tape.

Other Errors

list file error:
FMGR <nnn>

The indicated FMGR error occurred when FC attempted to access the list file (determined by the most recent LL command). This causes the list device to revert to the log device, and does not cause the command to terminate.

command file error:
FMGR <nnn>

The indicated FMGR error occurred when FC attempted to read a record from the command file (determined by the most recent TR command). This causes an automatic TR to the log device. (FC enters interactive mode.)

Errors That Terminate FC

fatal scratch file error:
FMGR <nnn>

An error occurred when FC tried to create or access one of its scratch files. Each scratch file created by FC is created on the first cartridge which has room for the initial size of the file.

If the error is FMGR-033 (disc cartridge full) or FMGR-014 (directory full), make more room on the cartridge or use the SCRATCH command to specify that FC create the scratch files on a cartridge with more room.

For any other error, if the SCRATCH command is being used, make sure it is used before any COPY commands are done.

can't load segment <segment name>

The indicated segment could not be loaded. Make sure that the type 6 files for the main (FC) and its segments (FC000,FC001,...,FC00n) all exist on the same cartridge and that no other files with the same names exist on other cartridges higher on the cartridge list. Then OF the main and all the segments and try running FC again.

TR stack overflow

The command file stack used for the TR command overflowed. This will occur if more than four nested TR commands are given (excluding any TR command in the runstring).

internal error at <address>, last segment loaded: <seg name>

FC detected an inconsistency that cannot be interpreted. This error should never occur under normal conditions. The address and segment name are for diagnostic use by HP.

internal error: pascal <error type#> <error#> <line#>

FC is being terminated due to a Pascal run-time error. This error should never occur under normal conditions. The numeric values are for diagnostic use by HP.

exec error <system error mnemonic>, exec params: <params in octal>

The operating system or I/O driver rejected an EXEC call made by FC. This error should never occur under normal conditions. The numeric values are for diagnostic use by HP.

Chapter 5
Cartridge Backup Utilities

Introduction

The cartridge save and restore utilities, WRITT and READT, allow you to back up disc cartridges to magnetic tape. You can back up a single cartridge or, by inhibiting the rewind feature and repeating the WRITT utility, back up multiple cartridges on a single tape. Cartridges can be selectively restored using the FMGR control command CN to position the tape to the desired file and then invoking READT.

Write Tape Utility (WRITT)

WRITT saves a private or group cartridge to tape and generates a tape-file header to identify the saved cartridge by LU or CRN number, cartridge label and cartridge type. The saved cartridges then can be restored from the tape to disc using the utility READT.

Calling WRITT

Invoke WRITT with the runstring

```
[RU,]WRITT[,sdisc[,MT:lu[,IH[,DC[,VE[,"..."]]]]]]
```

where:

sdisc is the disc cartridge to be saved. This parameter can be either a negative LU number or a positive CRN number. The defaults are defined below.

The following optional parameters are order-independent. They can be specified in any sequence.

MT:lu is the LU number of the magnetic tape unit on which the cartridge is to be saved. The default is to LU 8. This parameter also may be given using the LU number only (positive or negative), but then must be the second parameter in the run string.

IH inhibits tape rewind. The default is to rewind tape before and after each cartridge restoration.

DC disables the overlay check. Unless disabled by this command, WRITT checks to see if a previously stored cartridge file will be overlaid by this operation.

VE verify the mag tape; compare the data on tape with that on disc.

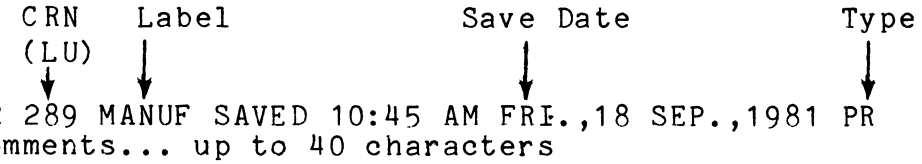
"..." comment to be appended to tape header and displayed by WRITT during the overlays check; 40 characters maximum.

If the source disc parameter is defaulted and you are under Session Monitor, WRITT saves the first private or group cartridge mounted to your session. If the source disc parameter is defaulted in a non-session environment, WRITT saves the first cartridge in the cartridge list. In either environment, if there are only system cartridges mounted, WRITT issues the message

```
ONLY THE SYS MNGR MAY SAVE SYSTEM DISCS
/WRITT: STOP
```

and exits.

When a cartridge is saved on tape, WRITT creates a file header to identify the cartridge, as



WRITT automatically rewinds the tape before and after a cartridge is saved, unless the IH parameter is specified to inhibit the rewind. If the end of tape is reached before the entire cartridge is saved, WRITT rewinds the tape, issues the message

```
Please mount another tape
After mounting enter "GO"
```

and waits for you to mount the tape. WRITT increments the tape number and writes the number to the continuation tape, then completes the save operation.

CAUTION

Do not attempt to replace a previously saved cartridge file on a tape as this can corrupt the remaining files on the tape.

Multiple saves to a single tape should be performed in a series of WRITT operations to save the cartridges sequentially on the tape. If you want to add a cartridge save to an existing tape, use the FMGR CN command to position the tape past the last file. Use the IH parameter to inhibit the tape rewind before and after each save. With the rewind inhibited, you must use the CN command (CN,lu,RW) to rewind the tape after the last save operation

Examples

In the following example, group cartridge 32754 (LU 39) is saved to tape. The mag tape is LU 9 and the comment "RTE-6/VM RELOCATABLES" is added to the header.

```
RU,WRITT,32754,DC,VE,MT:9,"RTE-6/VM RELOCATABL
CR 32754  HPDISK SAVED   3:59 PM  FRI., 24 SEPT, 1982  GR
RTE-6/VM RELOCATABLES
From LU      39
/WRITT: Verifying tape
/WRITT:  STOP
```

In the following example, WRITT is invoked to save private cartridge GU to tape. The DC parameter is not specified so WRITT scans the tape and, since an existing cartridge save file will be overlaid, issues a message identifying the tape file to be overlaid and requesting an OK to continue. The response is NO, and WRITT exits without saving the cartridge.

```
RU,WRITT,GU,VE,MT:9,"GROUP UTILITIES CRN (LAT
****** CAUTION ******
Do you want to
 overlay
CR 32754  HPDISK SAVED   3:59 PM  FRI., 24 SEPT, 1982  GR
RTE-6/VM RELOCATABLES
   with
CR    GU  UTILS  SAVED   4:07 PM  FRI., 24  SEPT, 1982  PR
GROUP UTILITIES CRN (LATEST SORT)
(Yes or No)?
NO
*** DISC NOT SAVED ****
/WRITT:  STOP
```

Error Messages

WRITT can generate the following error messages:

WRIT 001 MAG TAPE DOWN

The requested tape LU is not active. UP the appropriate EQT to enable the device.

WRIT 002 ONLY THE SYS MNGR MAY SAVE SYSTEM DISCS

System disc subchannel 2 and auxiliary disc subchannel 3 can only be saved by the System Manager.

WRIT 003 LU LOCKED

The specified tape unit is locked to another process.

WRIT 004 ILLEGAL MAG TAPE LU

The LU was not specified correctly (was not specified as a number between -63 and 63).

WRIT 005 MT OFF LINE

The tape unit is off line. Press the on-line switch to bring the unit on line.

WRIT 006 NO WRITE RING

The write ring must be installed before you can write to tape.

WRIT 007 PARITY ERROR

A parity error was detected when writing to the tape. If this error recurs, the tape may be irrecoverable.

WRIT 008 END OF TAPE

The EOT was sensed while writing a cartridge to tape. Mount another tape and enter AB to continue.

WRIT 009 FILE OPEN OR WRIT DISC LU LOCK REJECTED

The cartridge to be saved has an open file or is locked to another program.

WRIT 010 DISC NOT FOUND

The specified disc is not mounted.

WRIT 011 ILLEGAL DISC LU

The specified negative LU number is less than -63, the LU is not included in your Session Switch Table (SST), or the LU driver is not a disc driver.

WRIT 012 ONLY THE SYS MNGR MAY SAVE LU 2 OR LU 3

Only the System Manager may save system disc subchannel 2 or auxiliary disc subchannel 3.

WRIT 013 BAD TRANSMISSION--DISC TO MEMORY TRK xxx

Transmission is irregular in reading data from disc. If this error recurs, the disc may have a bad track.

WRIT 014 BAD TRANSMISSION--MEMORY TO MAG TAPE REC xxx

Transmission of data from memory to tape may be faulty. If this error recurs, see the System Manager.

WRIT 016 BAD TRANSMISSION - MAG TAPE TO MEMORY,REC xxxxx

An error was detected in transmission of data from the magnetic tape to memory. If this error recurs, the tape may be faulty. See your System Manager.

WRIT 020 VERIFY ERROR ON TRACK xxxxx

A compare error was encountered when verifying the listed track.

READT

READT restores a cartridge, previously saved on tape by WRITT, to a selected disc cartridge. Tracks are reformatted if required to satisfy the track/sector value of the saved file, and the directory tracks are moved if the destination cartridge is smaller than the cartridge from which the data was saved.

Calling READT

Invoke READT with the runstring

```
RU,READT[,disc[,MT:lu[,type[,SI:nnn[,IH[,VE[,CO]]]]]]]
```

where:

disc — is the disc cartridge to which the saved FMP cartridge is to be restored. This parameter can be either a negative LU number or a positive CRN number. The defaults are defined below.

MT:lu — is the LU of the magnetic tape unit on which the file is stored. The default is to LU 8. This parameter also may be given using the LU number only (positive or negative).

type — is the type of cartridge to be restored: P for private cartridge, G for group cartridge. (System cartridges can only be restored by the System Manager.) If the type specified differs from that in the tape header, the cartridge type given in the command runstring is assigned. The default is to use the cartridge type specified in the tape header. This parameter is meaningless in a non-session environment.

SI:nnn — is the size (specified in tracks) of the disc to which the magnetic tape contents are to be restored. The default is to the first pool cartridge large enough to accept the saved-cartridge file; the entire cartridge is mounted. This parameter also can be given without the SI designator, using only the decimal number of tracks required.

NOTE

The following optional parameters are order-independent and can be specified in any sequence.

IH inhibit tape rewind. The default is to rewind tape before and after the disc cartridge restoration.

VE verify the integrity of the data after the restore operation. The VE and CO options are mutually exclusive; only one may be specified.

CO perform a word-by-word comparison of the tape file and a previously restored cartridge.

When you are under Session Monitor and the specified disc CRN is not mounted, READT mounts a cartridge from the cartridge pool and assigns it the specified CRN number; if the disc is specified by LU number, READT mounts that LU if necessary. In a non-session environment, the disc specified by CRN must be a mounted cartridge as there is no non-session cartridge pool. If, however, the disc is specified by LU number, READT mounts that LU if necessary. In either environment, if the cartridge has an open file or is locked to another user, READT returns the appropriate error message and exits.

READT locks the cartridge during the restore operation, and unlocks it after the successful restore. Following the restore, the tape-file header created by WRITT and the LU or CRN to which the cartridge was restored is displayed at your terminal, as

```
CR   289    MANUF  SAVED 10;45 AM FRI.,18 SEP.,1981

RESTORED TO LU  53
READT:  STOP
:
```

If you are going to overlay a currently mounted cartridge, be aware that the type (P or G) of the cartridge to be overwritten can't be changed, and the overlaying cartridge must be at most the same size as the overlayed cartridge (the last track of the overlaid cartridge can't be moved). When the restore will overlay an existing cartridge, READT issues the message and prompt

```
DO YOU WANT TO OVERLAY CRN nnn ON LU nnn
WITH CRN nnn  (YES OR NO)?
```

If the available disc cartridges are not large enough for the cartridge to be restored, and if by moving the file directory the saved cartridge can be restored to an available cartridge, READT issues the following message and prompt

```
CRN nnn WAS SAVED FROM A xxxxxx TRACK DISC
LAST TRACK USED IS          xxxxxx
WOULD LIKE TO RESTORE TO A  xxxxxx TRACK DISC
IS IT OKAY TO MOVE DIRECTORY TRACKS (YES OR NO)?
```

A YES response allows READT to restore the saved cartridge to the available disc. A NO response terminates the utility.

If the saved cartridge file extends to more than one tape, READT issues the message

```
PLEASE MOUNT SUBSEQUENT TAPE
AFTER MOUNTING ENTER "GO" OR "ABORT"
```

READT checks the first record of the continuation tape to determine if it contains the tape count calculated by WRITT. If the tape count is not in the first record, the MOUNT TAPE message is repeated.

The size parameter specifies the number of tracks required to restore the cartridge. For example, if you specify 203 tracks, READT obtains a 203-track disc LU and places the first directory track at track 202.

If a restore is made to a disc LU that has a sector/track value that is different from the sector/track value on the tape file, READT issues a message indicating that reformatting of the directory is necessary to maintain the integrity of the file structure

```
TRACKS REFORMATTED FROM xxx SEC/TRK TO xxx SEC/TRK
```

READT then proceeds to restore the cartridge.

When the IH parameter is selected to inhibit tape rewind before and after the restore, the FMGR CN command (CN,lu,RW) must be used to rewind the tape. To position a tape to a particular file, repeat the CN,lu,FF command for each file to be skipped on the tape.

If the verify parameter (VE) is selected, the tape is rewound after the restore operation (using the CN command if the IH parameter is selected) and a word-by-word comparison is made between the tape file and the restored cartridge. If a verify error is encountered, READT issues the message

```
VERIFY ERROR ON TRACK nnn
```

The compare parameter (CO) does a word-by-word comparison between the tape file and a cartridge that was restored by a prior invocation of READT. This command does not restore a tape file to the cartridge.

Examples

In the following example, an attempt is made to restore a cartridge with the same CRN as a cartridge already mounted.

```
:RU,READT,289

 DUPLICATE CRN LABEL OR LU ALREADY MOUNTED
 DO YOU WANT TO OVERLAY CRN 289 ON LU 53
 WITH CRN 289 (YES OR NO)?NO
 /READT: STOP
:
```

In the following example, READT is called to restore a cartridge to LU 53, which is already mounted. READT issues the OVERLAY prompt and, following the YES response, overlays LU 53 with the CRN 289 cartridge file.

```
:RU,READT,-53

 DISC ALREADY MOUNTED
 DO YOU WANT TO OVERLAY LU 53
 WITH CRN 289 (YES OR NO)?YES
 /READT: CONTINUE

 CR  289  MANUF  SAVED 10:45 AM FRI.,18 SEP.,1981 PR
 RESTORED TO LU 53
 /READT: STOP
:
```

In the following example, READT is called to store a disc cartridge and change the CRN number by specifying another CRN number in the runstring.

```
:RU,READT,1000

 CRN 289 MANUF SAVED 10:45 AM FRI.,19 JAN.,1980 PR
 CRN 289 HAS BEEN CHANGED TO CRN 1000

 /READT:  STOP
:
```

When restoring to a disc containing previously restored program files, the following messages is issued and the tape is not restored.

```
:RU,READT,-2

 CR  2  SYS  SAVED 4:00 PM FRI., 4 FEB 1980

 THE FOLLOWING PROGRAMS HAVE ID SEGMENTS
 POINTING TO FMP TRACKS  YOU'RE REPLACING.
 THESE PROGRAMS MUST BE REMOVED BEFORE READT WILL
 REPLACE THESE TRACKS.

 FTN4
 RT6GN
 MERGE

 /READT:  STOP
:
```

Error Messages

READT can generate the following error messages:

READT 001 MAG TAPE DOWN

The requested magnetic tape LU is not active. UP the appropriate EQT to enable the device.

READT 002 BAD TAPE FORMAT

The tape contains information in a format not restorable by READT. The tape may have been constructed with another utility, or using the FMGR DU command, or may not be the correct volume in a multi-volume tape set.

READT 003 LU LOCKED

The tape device is locked to another process.

READT 004 ILLEGAL MAG TAPE LU

The magnetic tape device was incorrectly specified. Was not specified as a number between -63 and 63, or the driver of the specified LU is not a tape device driver.

READT 005 MT OFF LINE

The tape unit is off line. Press the on-line switch to bring the unit on line.

READT 006 ILLEGAL DISC LU

READT has rejected the use of the specified disc LU. The specified number is less than -63, the driver is not in your Session Switch Table (SST), or the driver of the specified LU is not a disc device driver.

READT 007 PARITY ERROR

The driver detected a parity error when reading rom the tape. If this error recurs, the tape may be irrecoverable.

READT 008 END OF TAPE

Mount the next tape in the set to complete the restoration.

READT 009 FILE OPEN OR READT'S DISC LU LOCK REJECTED

The disc has a file open or is locked to another program.

READT 010 NO SESSION; LU MUST BE NEGATIVE

You are operating in a non-session environment. A negative LU must be specified; there is no non-session disc pool.

READT 011 SIZE ERROR

The size parameter was not specified correctly, or does not specify sufficient tracks to restore the cartridge file.

READT 012 MOUNT ERROR, FMGR xxx

The listed FMGR error was encountered:

FMGR 012 Duplicate label or CRN already mounted. Remove disc or CRN and run READT again.

FMGR 056 The size requested is too large for the disc LU specified. Run READT again with a smaller size parameter.

FMGR 063 The maximum number of disc cartridges has already been mounted. Remove one and run READT again.

FMGR 064 There are no more free LUs in the cartridge pool.

FMGR 065 There is a conflict in SST definition. You are trying to mount a disc LU that has a session LU number assigned to a different device. Check your SST for the LU to which the number is assigned, change the specified number or choose another LU.

FMGR 066 There is no more room in your SST for an entry.

READT 013 SPECIFIED LU OR FREE LU NOT BIG ENOUGH TO MOUNT CRN

The disc LU specified in the runstring, and all available pool cartridges, are too small to restore the cartridge file.

READT 014 ILLEGAL RESTORE TO LU 2 OR LU 3

Only the system manager may save system disc subchannel 2 or auxiliary disc subchannel 3.

READT 015 PRIVATE/GROUP OPTION INCOMPATIBLE WITH EXISTING CARTRIDGE

The P/G attribute of the cartridge tape file conflicts with the attribute of the mounted cartridge. Run READT with the correct P/G option, or mount a compatible cartridge.

READT 016 BAD TRANSMISSION - MAG TAPE TO MEMORY, REC xxxxx

An error was detected in transmission of data from the magnetic tape to memory. If this error recurs, the tape may be faulty; see the System Manager.

READT 017 INTERNAL BUFFER TOO SMALL!

The tape records are longer that the READT internal buffer. The buffer size must be increased and READT must be reloaded.

READT 018 ABORTED BY USER

This message is produced when you respond NO to any prompt, or when READT is halted using the BReak command.

READT 019 DISC ERROR ON LU xx, TRACK xxxxx

READT encountered an error when reading the listed track of the listed LU.

READT 020 VERIFY ERROR ON TRACK xxxxx

A compare error was encountered when verifying the listed track.

READT 021 INVALID PARAMETER

An invalid parameter was specified in the READT command runstring. Check the runstring and re-enter the parameter.

Chapter 6
On-Line, Off-Line Physical Backup Utilities

Introduction

The on-line and off-line physical backup utilities PSAVE, PRSTR, and PCOPY, are used for transferring data between disc and tape transports or tape cartridges. The data transfer is on a track-by-track basis and does not assume that the disc subchannel contains valid FMGR information. You can specify the transfer of all tracks on a named subchannel or all subchannels on a named disc unit. Track sparing is provided and mounted FMP cartridges are protected.

PSAVE (SA in off-line mode) saves discs to tape; PRSTR (RE in off-line mode) restores the saved tapes to discs, and PCOPY (CO in off-line mode) copies data from disc to disc. In off-line mode, the program BCKOF schedules the appropriate backup utility and provides the help function and I/O configuration information.

The off-line mode basically supersets on-line mode by allowing track-map table modification. Unit saves also may be specified in pushbutton-restorable format, either off-line or on-line, if the destination device is a CS80 tape cartridge and the source is a CS80 disc.

Compatibility with Other Disc Backup Utilities

The physical backup utilities replace the ICD/MAC and 7900 backup utilities. Tapes created by PSAVE are not compatible with these utilities; however, PRSTR can read tapes created in either the ICD/MAC or 7900 formats. The physical backup utilities are not compatible with tapes created by the READT/WRITT, READR/SAVER, FC, or other file backup utilities.

Compatibility Among Disc Drives

The physical backup utilities do not use any file structure information. This means that source and destination discs used in copy and in restore operations must have the same track size (except as noted below). If you must transfer between discs with different track sizes, use the file backup utility (FC) to transfer the files individually or in groups, rather than on the track-to-track basis.

The track size in the subchannel definitions for CS80 discs are logical track sizes, and may not necessarily be equal to the physical track size of the disc. Using the off-line utility, any CS80 disc can be restored to any other CS80 disc regardless of the physical track size. The on-line utility must protect the system integrity by not allowing a disc to be restored if it cannot be referenced properly by the existing system subchannel definitions.

Table 6-1 defines the source and destination compatibilities of the various disc drive models.

Table 6-1. Disc Drive Source/Destination Compatibility

	Destination						
Source	7900	7905	7906	7910	7920	7925	CS80
7900	O.K.	*	*	N/A	*	N/A	+
7905	*	O.K.	O.K.	N/A	O.K.	N/A	+
7906	*	*	O.K.	N/A	O.K.	N/A	+
7910	N/A	N/A	N/A	O.K	N/A	N/A	+
7920	*	*	*	N/A	O.K.	N/A	+
7925	N/A	N/A	N/A	N/A	N/A	O.K.	+
CS80	+	+	+	+	+	+	O.K.

CS80 Discs include the 7908, 7911, 7912, 7914 and 7933.

N/A Not available.

* If restore is off-line, you must supply subchannel definition.

\+ Logical track sizes for the CS80 sealed discs have been set at 96 sectors per track. This is variable, so any CS80 source can go to any CS80 destination off-line. On-line operations use the current subchannel definitions.

Note: Since PRSTR only knows the disc types returned by LDTYP, you must be very careful when restoring a tape off-line using the track map definitions on the tape; Table 6-1 should be consulted. The off-line option UD (refer to the section Restore Tape File <RE> for details), forces you to specify the track map definitions. This is automatic when the disc types (CS80, MAC, ICD) are different.

Tape Handling

Tape handling considerations differ between tape transports and CS80 cartridges, since the cartridge is a streaming mode, blocked record device, and the transport is a random-length record storage device. (Refer to the section Data Structures for details of the tape formats.)

Efficient handling of cartridge tape is provided by the CS80 device driver disc-caching routines that control on-line transfers to the cartridge. Data to be transferred is buffered on the integrated disc drive and transferred in a steady stream to the cartridge tape, thus allowing continuous movement of the tape. Since caching is not possible off-line, PB (pushbutton) saves and restores are substantially faster in this mode. (Refer to the DVM33/DVN33 Driver Manual for details of the disc-caching scheme.)

When the source or destination for the backup operation is a CS80 cartridge tape (CTD), the utility will prompt with

```
DEFINE CTD LU SUBCHANNEL -

CURRENT CTD DEFINITION IS:

ADDRESS,   UNIT#,   VOLUME#,

   0,        1,        0,

ADDRESS,   UNIT#,   VOLUME#,
```

and wait for your response. Enter the correct configuration, followed by a carriage return. The HP-IB address for the CTD is set using the address switch at the rear of the drive unit. Refer to the Operating and Installation Manual provided with your CS80 drive for details of the switch settings.

Tape Positioning

When the backup device is a tape transport, the save and restore functions start at the current tape position unless a file number is specified. If specified, the number is interpreted as relative to the start of the present tape, and the tape is rewound. (Note that the desired file number need not be on the currently mounted tape volume of a multi-volume set.) For the SE (selective restore) and UN (unit restore) options, all subsequent file numbers are relative to this initial positioning. Since the tape is not rewound after a save or restore, you can save more than one LU on a tape. On-line, you can use the FMGR CN command (control non-disc devices) to position the tape before an operation. Off-line, the tape movement commands are a part of the utility tasks that can be specified.

If an end-of-tape mark (EOT) is detected during an operation, a message is issued and the utility waits for you to mount a new tape and enter the command to continue.

When the backup device is a CS80 cartridge, PSAVE always assumes the save is to start at the first logical file on the cartridge unless you specifically designate a file number greater than one. Since file seeks on cartridges always start at block 0, it is advisable to use the multiple LU save option (MU) if multiple saves are going to the same cartridge. End-of-cartridge is handled the same as for tape transport EOT.

CAUTION

> If a magnetic tape position is specified beyond the end of all data on the tape, a tape runaway condition may be encountered.

The last save written by PSAVE is followed by two End-of-File (EOF) marks. This constitutes an end-of-data mark to the utilities. A warning is issued if you specify a file beyond a double EOF mark.

Unit Save Tape Files

In a unit save, which is simply a series of LU saves, each disc subchannel is saved on tape as one file. LU saves are assigned sequential tape file numbers relative to the current position of the tape: last save file# + 1, or file #1 if the tape was rewound or is a CS80 cartridge. The save file numbers on the tape are thus independent of the LU number of the subchannel saved to tape. As a result, you must specify the tape file number, not the subchannel number, when you are doing a selective restore either on-line or off-line.

On a CS80 cartridge, for any save not in pushbutton-restore format (PB option), the cartridge contains an EOF mark at the beginning of the tape. The EOF mark prevents inadvertant restoring of the cartridge using the front-panel pushbutton controls, since the manual restore operation stops at file marks. The empty file is interpreted as file 0, thus the first save file on a CS80 cartridge is file 1, just as it is for tape transports.

Multiple-Volume Tape Sets

Volumes in a multivolume tape set are numbered consecutively, and each begins with a full set of headers. If a single LU save crosses over tapes, the save definition header contains the track and sector at which the save resumes on the tape. If the destination is a tape cartridge, the current checksum record is written in the last two blocks of the cartridge.

Data Verification

The on-line utilities PSAVE and PRSTR both provide for data verification as an option (VE). The off-line commands SA and RE also allow the VE option. Data verification is identical for both off-line and on-line saves and restores, except for PB operations. When the operation is a restore, specifying the VE option also causes the destination disc tracks to be spared in a prepass over the disc (except CS80 discs).

Verification of Saves

Verification of the save is done in two steps: 1) verify that the disc was read correctly and 2) verify that the tape was written correctly.

The first step is accomplished during the save operation. When the disc is read, a checksum is calculated. The disc is then re-read and a new checksum is calculated. If these two checksums agree, the data is assumed to be correct. If they do not agree, the above process is repeated until two succesive reads yield the same checksum or until the calculation has been tried ten times. If no two reads are the same, the disc is read sector by sector to recover as much of the save data as possible.

The second step, that of verifying the tape write, is accomplished in a second pass. This pass occurs when the last save in a series is completed, or when a new tape must be mounted. The utility rewinds the tape whenever possible, but if the first save in the series did not start at the beginning of the tape, the files are backspaced to the beginning of the first save. Each tape record is verified by testing for a correct checksum. Note that the save utility does not recognize the break flag (BR) during the verify pass.

Verification of Restores

Verification of the restore is done in two steps: 1) verify that the tape was read correctly and 2) verify that the disc was written correctly.

The first step is accomplished by testing for correct checksums as the tape records are read. In the second step, the disc is re-read when it is written to determine if the correct data was transmitted to the disc. This process is repeated up to ten times; if it is still not successful, a sector-by-sector write is done in an attempt to correctly restore as much data as possible. Note that the restore is verified as the disc is written; the tape does NOT need to be re-read in a second pass.

Pushbutton Save/Restore Verification

Data integrity for pushbutton save/restore operations is verified using the cyclic redundancy check performed by the CS80 controller. If, however, a bad region exists on the media (tape or disc), the PB operation could fail. The VE option, in conjunction with a PB restore, ensures that a verify pass will take place on the destination disc after the restore operation.

On a PB save, the VE option causes a verify pass to take place on the tape at the completion of the save operation. If a bad block should occur, the number is reported and a verify error is returned. (Sparing is automatic if a data error occurs.) Thus, if a PB save fails to verify the data, repeating the operation will not fail to verify the restore operation for the block.

Pushbutton (PB) Operations

The physical backup utilities are capable of emulating the off-line Cartridge Tape backup capability of many of the CS80 disc drives. This manual-backup operation is referred to in this subsection as a PB (PushButton) operation.

The 7908, 7911, 7912, and 7914 disc drives in the standard configuration have integrated Cartridge Tape Devices (CTDs), and can perform PB operations either manually or by using the utilities. However, there are options with these disc drives which will delete the cartridge tape device or provide separate controllers for the disc and CTD. This subsection does not apply to these configurations, nor to the 7933 disc drive, which does not have an integrated CTD for backup. None of these drives are capable of performing a PB operation either manually or by using the utilities.

A PB operation on a CS80 disc with CTD is performed manually from the front panel of the disc drive. This operation is also emulated from the physical backup using the PB option when saving or restoring. These utilities provide the capability of a post save or restore verify pass (and in the case of a restore, a pre-verify and spare pass) if the user specifies the VE option. Some devices are capable of performing a PB save or restore using two tapes, and this feature is also emulated by the physical backup utilities.

The PB save or restore operation from the utilities is merely a command to the device controller to perform a copy from a source to a destination for a unique length. This means that if the user specifies a disc LU which is on a different bus or a different bus address than the Cartridge Tape Device LU, then the operation cannot be performed. The disc and the CTD must be integrated on a single controller, which is at a single HPIB address on a bus, in order for the PB operation to take place.

Please note the capacities of the tapes which you are using as specified in the Operating and Installation manual supplied with your disc drive. If you are using a 64-megabyte disc drive, then to perform a PB backup of the entire disc you will need a cartridge tape capable of holding 64 megabytes.

Below is shown an example of using the PB option from the physical backup utilities for a save operation. In this example, the save operation completes successfully after reporting that a block that had been found bad in some earlier operation to the tape has now been spared from further use. On the post verify pass, one block was found to have required a retry to get the correct data (as determined by internal CRCs). Note that unless the message CS/80 ERROR: xxxxxxx is printed out (where xxxx is some type of message), any status printed from the utilities on a PB request is just an informative message, not intended to indicate that an error has occurred.

```
:RU,PSAVE
ENTER OPTIONS
vepb                    ! Verify data; PushButton operation
ENTER SOURCE DISC LU
2                       ! CS/80 Disc
ENTER TAPE LU
13                      ! CS/80 CTD on same bus & HPIB address
ENTER HARDCOPY LU
1                       ! Default to terminal scheduling PSAVE

SCHEDULING PUSHBUTTON OPERATION     ! PSPAR being scheduled

COPY FROM DISC TO TAPE PROCEEDING.  ! Save has started.
CS/80 NOTE:  AUTO SPARING WAS INVOKED.  NO DATA WAS LOST.
CS/80 - FULL DEVICE STATUS:
000060B: IDENTIFICATION FIELD.
PARAMETERS - 000000B, 000000B, 130201B, 239481B, 293212B.
(SEE DVM33/DVN33 REFERENCE MANUAL FOR MORE INFORMATION.)

VERIFYING TAPE.
CS/80 - FULL DEVICE STATUS:
000020B: IDENTIFICATION FIELD.
PARAMETERS - 000000B, 000000B, 001010B, 121212B, 121212B.
(SEE DVM33/DVN33 REFERENCE MANUAL FOR MORE INFORMATION.)
COPY COMPLETED SUCCESSFULLY.
PSAVE NORMAL END OF JOB.
```

The following is an example of using the PB option in a restore operation. There are no errors on either the pre-verify and sparing pass, the restore, or the post-verify pass on the disc.

```
:RU,PRSTR
ENTER OPTIONS
pbve
ENTER DEST DISC LU
14
ENTER TAPE LU
13
ENTER HARDCOPY LU
<cr>

SCHEDULING PUSHBUTTON OPERATION

COPY FROM TAPE TO DISC PROCEEDING.

VERIFYING DISC.
COPY COMPLETED SUCCESSFULLY.
PRSTR END OF JOB.
```

Using the BReak Command

Both the on-line and off-line utilities check the break flag prior to each tape or disc I/O operation. The current operation is ended and all resources are released when a break is detected. The one exception is during data verification, when the break condition has no effect.

The on-line backup utility terminates when the break flag is detected, while the off-line utility returns to the interactive TASK? prompt state. When the BR command is used, the interrupted operation must be considered invalid.

Breaking a save causes the utility to write double EOF's at the end of the current save to mark the end of data on the tape.

On-Line Physical Backup Utilities

The on-line utility programs PSAVE, PRSTR and PCOPY run under control of RTE-6/VM, thus the logical units and disc subchannel definitions are determined by the tables in the current system. Each of the utilities can be run interactively, by using a command runstring, or by means of a command file. When the utilities are invoked using a runstring and you specify an erroneous parameter, the utilities enter interactive mode and prompt for the correct parameter.

To prevent suspending or aborting critical operations that require access to a cartridge, PSAVE does not lock the cartridge during the save operation. It is recommended that you ensure that no other program writes to a disc LU that is in the process of being backed up by PSAVE. In a multi-CPU environment, you should ensure that no other CPU has the disc mounted when you attempt to save or restore it.

PRSTR and PCOPY lock the disc being restored and will not restore to an already mounted FMP cartridge. To restore a disc LU on-line, you must first dismount the cartridge with the FMGR DC command. To restore a disc unit on-line, you must dismount all LUs on that unit. However, PRSTR issues a message listing the mounted LUs and allows you the option of restoring those LUs that are not mounted. If all cartridges are mounted, PRSTR terminates with an error message after listing the LUs that must be dismounted.

Note however, that the system and auxilary discs, LU 2 and LU 3, cannot be dismounted, and any attempt to do an on-line LU, UNit, or SElective restore to these LUs will result in an error. A UNit or SElective restore will attempt to restore those LUs that were dismounted and lockable, if the user responds with Yes to the OK TO PROCEED (Y or N)? prompts. The tape files containing the PSAVE of LU 2 and LU 3 will be skipped in the restoration process.

The PushButton option does allow the user to restore LU 2 and LU 3 on-line. The following warnings are printed for the user, and the user must answer Yes to all the questions before the restore will take place.

```
*WARNING* SYSTEM LU <n> WILL BE OVERLAYED
OK TO PROCEED (Y or N)?

 ******  W A R N I N G  ******
 ALL ACTIVITY MUST BE TERMINATED BEFORE RESTORE PROCEEDS.
OK TO PROCEED (Y or N)?
```

The first question shown above may be suppressed using the DE option, but the latter must be answered interactively by the user. The user can enter EX or /E to any of these questions and the utility will terminate. After the restoration on the system disc completes, the system will be HALTed to allow the user to reboot. This is because the system on the disc and the system in memory can not be guaranteed to be the same. If PRSTR is executing under Session environment, the user must have a capability level greater than, or equal to 60 to do an on-line PushButton restore to the disc containing LU 2 or LU 3.

A unit restore in CS80 pushbutton format is not restorable selectively since the pushbutton controls on the drive unit allow only full save or full restore. Therefore, all cartridges on the CS80 disc must be dismounted before a data restore can be done.

Loading the On-Line Utilities

The loading sequence allows the retrieval of some code space for use as an I/O buffer by loading the routine $BEGGT last. On-line loading includes the relocation of a $ONLIN module that resolves references to special routines needed only in the off-line environment.

The on-line utilities PSAVE and PCOPY are loaded interactively as

```
:RU,LOADR
 /LOADR: FM,CP          !format current page links
 /LOADR: OP,LB          !load program as large background
 /LOADR: SZ,27
 /LOADR: RE,%COMM
 /LOADR: RE,%PSAVE      !load PSAVE, or
 /LOADR: RE,%PRSTR      !load PRSTR, or
 /LOADR: RE,%PCOPY      !load PCOPY
 /LOADR: RE,$ONLIN      !load routine to resolve references needed
                        !only in off-line utility
 /LOADR: SE,$BCKUP      !load system-independent physical backup
                        !library relocatable modules
 /LOADR: SE,$DTCLB      !load CS80 disc library relocatable modules
 /LOADR: SE,$PLIB       !load Pascal library relocatable modules
 /LOADR: SE             !search system libraries to assure that
                        !$BEGGT is relocated last
 /LOADR: RE,$BEGGT      !this set of routines should be relocated
                        !last,  as it can be overlaid for use as
                        !an I/O buffer
 /LOADR: /E
```

The PSPAR utility, scheduled by PRSTR and PCOPY when the VErify option is selected, is loaded as

```
:RU,LOADR
 /LOADR: FM,CP
 /LOADR: OP,LB
 /LOADR: SZ,20
 /LOADR: RE,%COMM
 /LOADR: RE,%PSPAR
 /LOADR: RE,$ONLIN
 /LOADR: SE,$BCKUP
 /LOADR: SE,$DTCLB
 /LOADR: SE,$PLIB
 /LOADR: /E
```

You can also use the LOADR command files #PSAVE, #PRSTR, #PCOPY, and $PSPAR to load the physical backup utilities into memory and then use the SP command to place them on LU 2.

PSAVE

PSAVE saves one disc LU, a group of disc LUs or an entire disc unit to tape transports or tape cartridges. As an option, you can save an entire disc unit in CS80 pushbutton-restorable format.

Invoke PSAVE with the runstring

[RU,]PSAVE[,input[,srceLU[,destLU[,file#[,opts[,hcpy[,title]]]]]]]

where:

input — is the LU (or file) from which the PSAVE parameter inputs are to be read. If input is from a file or or a non-interactive device, no other parameters may be specified. The default is to your log device.

srceLU — is the LU of the disc subchannel to be saved. If the UN (unit save) option is specified, the srce LU parameter is a target to the unit and may be any LU on the disc. If the MU (multiple save) option is specified, this parameter is meaningless and will be ignored. In any other case, this is a required parameter.

destLU — is the LU of the tape transport or tape cartridge to receive the saved data. Under session monitor, the LU specified must be in your Session Switch Table (SST). The default is to LU 8.

file# — is the integer number to be assigned to the saved file. This parameter specifies the start location on the tape for the save. Before writing, file# - 1 tape files are skipped from the start of the tape. The default is to the current tape position for tape transports or file #1 for CS80 cartridges.

opts — are any of the following two-character ASCII option codes. The options can be specified in any order, and may not have intervening characters (including blanks) between options. If no options are specified, the default is to LU save without verify.

VE Turn on verify option. Track sparing is done only when the verify option is specified.

LU Save is to be an LU save.

UN Save is to be a disc unit save. The source LU parameter can be any LU on the disc unit.

PB Unit save in CS80 pushbutton-restorable format. The destination LU must be a CS80 cartridge, and the source LU must be a CS80 disc. PSPAR must be loaded into memory for this option.

MU Multiple LU saves in one pass. The source LU parameter should be omitted; PSAVE will prompt for the source LUs. All LUs must be from the same class of discs, such as all MAC disc LUs.

hcpy is the LU of the device on which information about the save is to be printed as a record of the save operation. The default is to your log device.

title is the title (to a maximum of 40 ASCII characters) that will be placed in the tape header.

If you enter at least one parameter in the runstring, PSAVE defaults all optional parameters and prompts you for all required parameters not included in the runstring.

Calling PSAVE Interactively

To run PSAVE interactively, just enter the utility call PSAVE or RU,PSAVE. Enter a valid response to each prompt or enter a carriage return to use the default value. If a response is not valid, an appropriate error message is issued and the prompt is repeated. PSAVE prompts for input in the following order. (Note: This is the order that should be used in preparing an input command file.)

Prompt	Response
ENTER OPTIONS	Enter any of the above option codes. Since the UN, LU, PB and MU options are mutually exclusive, only two options are possible, at most. The options must be specified without any intervening characters, including blanks. (If you respond with "??". PSAVE will display a list of available options.)
ENTER SOURCE DISC LU	If the selected option is UN or PB, this parameter can be any LU on the disc unit. If the selected option is LU, this should be the LU to be saved. If the selected option is MU, a string of LUs may be entered, separated by commas. To continue the list on a second line, end the line with a comma.
ENTER TAPE LU	Enter the LU of the destination tape transport or CS80 cartridge. Enter a carriage return to specify the default, tape LU 8.
ENTER FILE NUMBER	Enter the file number to specify the point at which the save is to begin on the tape. (File# - 1 files are skipped.) Enter a carriage return to specify the defaults: current tape position on tape transport, file #1 on CS80 cartridge.
ENTER HARD COPY LU	Enter the LU of the device on which the save operation information is to be printed. Enter a carriage return to specify the default LU, the log device.
ENTER TITLE	Enter up to 40 ASCII characters of label information to be stored in the tape header. Enter a carriage return if no title is desired. The title may be entered in upper-case characters, lower-case characters, or a combination of both.

A response of AB, /A or /E to any of the above prompts causes the program to terminate cleanly, releasing all resources. Once the program has started processing the tape, it may be terminated by entering break mode (hitting any key on the keyboard) and entering BR,PSAVE. The OF command should not be used to terminate PSAVE.

Before beginning the save operation, PSAVE checks that the tape is on-line and write-enabled. If either condition is not met, the appropriate error message is issued and the utility prompts with

```
TYPE 'GO' WHEN READY ('PA' TO SUSPEND)
```

When the tape is on-line and write-enabled, enter GO and PSAVE will continue. If you enter PA, the utility suspends itself after issuing the message

```
TYPE  GO, PSAVE TO CONTINUE
```

PSAVE prints the tape number, file number, and the tape header records on the log device and the hard-copy LU (if specified) for each save as it is processed. If any serious errors are encountered, the utility terminates with an appropriate error message.

If PSAVE is run from a command file, information is returned to the father program by a PRTN call. The first parameter is either a zero, signifying valid completion, or an error number. (The error number returned corresponds to the error numbers assigned to the error messages.) If the scheduling program is FMGR, the error is returned in the 1P global. If you are under session monitor, an error is posted to the SCB. The error number and the subroutine in which the error occurred are displayed on the log device.

Examples

In the following example, PSAVE is invoked to perform multiple LU saves in a single pass (MU option) and to verify the saved data. Although the source LU is specified in the runstring, PSAVE ignores it and prompts for the parameter. Two files are specified in the prompt response, however the MU option also can be specified to save a single LU. The destination is defaulted to tape transport LU 8, and the starting file is specified as file #1. The title "PBU Test 091581" is added to the tape header for both files. In this example, the TAPE UNIT OFF-LINE message is issued after PSAVE checks the status of the destination LU.

```
:RU,PSAVE,,,,,VEMU,1,PBU Test 091581

 TAPE LU DEFAULTED TO           8
 ENTER DISC LU(S)
 45,61

 TAPE UNIT OFF-LINE
 ENTER "GO" WHEN READY ("PA" TO SUSPEND)
 go

 SAVING DISC LU   45 TO FILE   1  TAPE    1

 TAPE NUMBER:   1   SAVE FILE:   1   USER: BILL.JT
 PROGRAM:  PSAVE    OPTIONS:  VE  MU   DATE: WED 9 SEPT, 1981
 DISC LU:  45       TITLE:  PBU TEST 091581
 SECTION:   1       (TRK    0    SEC     0)

 LU 45 SAVED ON FILE   1   TAPE   1

 SAVING DISC LU   61 TO FILE   3  TAPE    1

 TAPE NUMBER:   1   SAVE FILE:   2   USER: BILL.JT
 PROGRAM:  PSAVE    OPTIONS:  VE  MU   DATE: WED 9 SEPT, 1981
 DISC LU:  61       TITLE:  PBU TEST 091581
 SECTION:   1       (TRK    0    SEC     0)

 LU 61 SAVED ON FILE   2   TAPE   1

 PSAVE NORMAL END OF JOB
:
```

(text deleted)

In the following example, the FMGR command CL is used to first obtain a listing of all available LUs, and PSAVE is then invoked interactively. The "??" response to the ENTER OPTIONS prompt causes PSAVE to display all of the options allowed with this utility. The VE option is then specified to verify the LU save (the default if the LU, UN, or MU option is not given). Since the file number is defaulted, the number assigned to the saved LU will be determined by the current position of the magnetic tape on LU 8.

```
:cl
   LU  LAST TRACK   CR    LOCK  P/G/S

   14      00202      L           P
   32      00202      S           P
   19      00202      O           P
   22      00202      E4          P
   37      00202      PV          P
   16      00202      CS          G
   02      00255   00002          S
   03      00255   00003          S
   50      00424   00050          S
   51      00202   00051          S

:psave
 ENTER OPTIONS
 ??
 ALLOWED OPTIONS (SAVE):
 (1) VE => VERIFY
 (4).LU => LU SAVE/RSTR/COPY
 (4) UN => UNIT (FULL VOL) SAVE/RSTR/COPY
 (4) PB => PUSHBUTTON UNIT IMAGE SAVE/RSTR
 (4) MU => SAVE OF MULTIPLE LUS

 THE NUMBERS IN ()'S REPRESENT CATEGORIES AND
 ONLY ONE OUT OF EACH CATEGORY CAN BE USED.

 ENTER OPTIONS
 ve
 ENTER SOURCE DISC LU
 37
 ENTER TAPE LU
 8
 ENTER FILE NUMBER
 <cr>

 ENTER HARD COPY LU
 6
 ENTER TITLE
 This is my private cartridge.
```

```
 SAVING DISC LU  37 TO FILE   1 TAPE   1
 TAPE NUMBER:   1  SAVE FILE:     1 USER: BILL.JT
 PROGRAM: PSAVE    OPTIONS: VE  LU  DATE: 10:36 AM TUE., 18 MAY,1982
 DISC LU:    37    TITLE: This is my private cartridge.
 SECTION:     1    (TRK      0  SEC      0)

 LU  37 SAVED ON FILE   1 TAPE   1

 VERIFYING TAPE   1

 PSAVE NORMAL END OF JOB
:
```

PRSTR

NOTE

When restoring tapes saved using the ICD/MAC or 7900 utilities, be aware that the USAVE and SAVE UNIT utilities store a disc unit as a single tape file. These unit saves are therefore not selectively restorable; efforts to position the tape beyond the start of the save will fail.

PRSTR restores a single subchannel LU, a group of LUs, an entire disc drive unit, or selected files from a tape transport or tape cartridge previously saved by PSAVE. This utility also can be used to read tapes written in SAVE, LSAVE, and USAVE formats. When the verify option is selected, PRSTR spares the destination disc tracks in a prepass over the disc before data is written. (Sparing is not done for 7900 discs.)

Invoke PRSTR with the runstring

```
[RU,]PRSTR[,input[,dest LU[,srce  LU[,file#[,opts[,hcpy]]]]]]
```

where:

input is the LU (or file) from which the parameter inputs are to be read. If the input is from a file or a non-interactive device, no other parameters may be specified. The default is to the log device.

dest LU is the LU of the disc subchannel on which the data is to be restored. If the UN (unit save) option is specified, the LU is a target and may be any LU on the disc. If the selective restore (SE) option is specified, this parameter must be omitted. In all other cases, the parameter is required.

srce LU is the LU of the tape transport or CS80 cartridge from which the save file is to be read. Under session monitor, the LU specified must be in your Session Switch Table (SST). The default is to LU 8.

file# is the integer number of the file to be read. File# - 1 files will be skipped on the tape before the read begins. Default is to the current tape position for tape transports, or file #1 for CS80 cartridges.

opts are any of the following two-character ASCII option codes. The options can be specified in any order, and may not have intervening characters (including blanks) between options. If no options are specified, the default is to not verify and LU restore if the save was an LU save or UN if the save was a unit save.

VE Turn on data verify option. PRSTR also spares the destination disc tracks in a prepass over the disc before data is written. PSPAR must be loaded into memory for this option.

DE Do not prompt for verification of tape file header. Do not prompt for OK TO CONTINUE? when disc source and destination sizes do not match. (PRSTR will continue with the restore.)

UN Unit restore. This is the default if the tape is a unit save or a from-to save. (Note that from-to saves performed using !DSKUN must use the UN option to restore the save.)

LU Lu restore. This is the default if the tape is an LU or MU save.

SE Selective restore from a unit save, or multiple LU restores in one pass. If this option is selected, the destination LU parameter must be omitted; PSAVE will prompt for the file:LU pairs. Not valid for a unit save in pushbutton- restorable format or a unit save in formats other than those created by PSAVE. All LUs must be of the same disc class.

PB Unit save in CS80 pushbutton-restorable format. PSPAR must be loaded into memory for this option.

hcpy is the LU of the device on which information about the restore is to be printed. The default is to your log device.

If you enter at least one parameter in the runstring (beyond the input parameter), PRSTR defaults all of the optional parameters and prompts for the remaining required parameters not included in the runstring.

If the SE option is selected, PRSTR prompts with the message

ENTER FILE:LU PAIRS

and waits for you to enter the file number and destination LU. The file numbers can be entered in random order; they are reordered and restored in ascending order. A string of file/LU pairs may be entered, using a comma to separate the entries. To continue the list on another line, end the current line with a comma. If duplicate file/LU pairs are entered, only the first entry pair is restored; duplicates are ignored.

Calling PRSTR Interactively

To run PRSTR interactively, enter the utility call PRSTR or RU,PRSTR. Enter a valid response to each prompt or enter a carriage return to use the default value. If a response is not valid, an appropriate error message is issued and the prompt is repeated. PRSTR prompts for input in the following order. (Note: This is the order that should be used in preparing an input command file.)

Prompt	Response
ENTER OPTIONS	Enter any of the above option codes. Since the SE, PB, UN, and LU options are all mutually exclusive, only three options are possible, at most. The options must be specified without any intervening characters, including blanks. (If you respond with "??", PRSTR will display a list of available options.)

If you select the UN, PB or LU option:

ENTER DEST DISC LU — If the PB or UN option is selected, this parameter can be any LU on the disc drive unit. If the LU option is selected, this must be the destination LU for the restore operation.

If you select the SE option:

ENTER FILE:LU PAIRS — Enter the restore file number and the destination LU, separated with a colon. File numbers can be entered in random order; they are reordered and restored in sequence. A string of entries must be separated with commas. To continue the list on another line, end the current line with a comma. If a file number:LU pair is duplicated, only the first entry is restored; subsequent entries are ignored.

ENTER TAPE LU — Enter the LU of the source tape transport or CS80 cartridge. Enter a carriage return to specify the default tape transport LU 8.

If you do not select the SE option:

ENTER FILE NUMBER — Enter the number of the saved file to be restored. Enter a carriage return to specify the defaults: current tape position on tape transport, file #1 on CS80 cartridge. If the UN option is selected, the tape must be positioned at the first file in the original unit save, or an error will be returned.

ENTER HARD COPY LU — Enter the LU of the device on which the restore operation information is to be printed. Enter a carriage return to specify the default LU, your log device.

A response of AB, /A or /E to any of the above prompts cause the program to terminate cleanly, releasing all of its resources. Once the program has started processing the tape, it may be terminated by entering break mode (hitting any key on the keyboard) and entering BR,PRSTR. The OF command should not be used to exit PRSTR.

Before restoring any files, PRSTR reads the first record header and reports those LUs that cannot be restored because: they cannot be locked, they are mounted FMP cartridges or, in the case of a unit restore, there is no subchannel on the target disc corresponding to that on the tape. PRSTR issues a prompt giving you the option of aborting the operation or continuing with those saves that can be done.

When restoring to any disc unit, all drive protect switches must be turned off and the format switch must be turned on. (Since PRSTR does not recognize upper- and lower-platter protect, both protect switches must be turned off when restoring a 7905 or 7906 disc LU.) If either condition is not met, an error message is issued and PRSTR prompts with:

```
TYPE 'GO' WHEN READY ('PA' TO SUSPEND)
```

When the switch setting is correct, enter GO and the program continues. If you enter PA, the program suspends itself after issuing the message

```
TYPE  GO, PRSTR TO CONTINUE
```

After the tape header is printed on the log device, PRSTR issues a prompt asking you to verify that this is the correct file (unless the DE option was selected). If your response is YE, the tape file is restored to disc. If your response is NO, PRSTR rewinds the tape and again requests the file number. The file number entered is now relative to the start of the present tape. The tape is advanced the entered number of files minus one and the header is printed. PRSTR again asks for verification and, after a YE response, restores the file to the original disc LU. Subsequent file advances on SE or UN restores will then be relative to this new position.

The rewind feature allows you to position the tape to a file preceding the one that was rejected. This capability should be used with caution in SE and UN restores. Specifying a file number using the reject/select process will impact all subsequent restores since files are restored by their position on the tape relative to the first file and are restored to the absolute LU specified when PRSTR is invoked.

If, for example, a tape contains four files, you specify an SE restore using the pairs 1:20 and 2:22, and then find when the first file header is displayed that you really really wanted to restore the second file. You can either enter /E to terminate PRSTR, or respond to the file number prompt with "2" to bypass file #1 and restore file #2 as the first file. In this case, file #2 will be restored to LU 20, and the next file on the tape (#3) will be restored to LU 22. Further, if the original SE pairs were 1:20 and 3:22, repositioning the first restore to file #2 would cause file #4 to be restored to LU 22, and file #3 would not be restored.

Note, however, that PRSTR prompts with the file verification message and the OK TO PROCEED?. This allows you to reposition the tape to absolute file #3, if desired, and continue.

On all restores, tapes must be mounted sequentially, and no tape volumes in a set may be skipped. Internal checks are made to ensure this proper sequencing of tapes.

As each restore is processed, PRSTR displays the tape number, file number and the tape title information on the log device. If fatal errors are encountered, PRSTR terminates with an error message.

If PRSTR is run from a command file, information is returned to the father program by a PRTN call. The first parameter is either a zero, signifying valid completion, or an error number. (If the scheduling program is FMGR, the error number is returned in the IP global.) If you are under session control, an error is posted to the SCB.

CAUTION

Be aware that the directory tracks might not be copied when a disc LU with FMGR files and directory tracks is restored to a destination disc LU with fewer tracks than the source disc. If the directory tracks are not written, the restore operation will not be valid.

If the source disc LU saved on the tape file and the destination disc LU do not have the same number of tracks, PRSTR issues the message

```
xxxxx TRACKS IN SOURCE LU
yyyyy TRACKS IN DEST   LU
OK TO PROCEED? (YE/NO)
```

If you respond YE, PRSTR copies tracks until it has filled the last track of the destination LU or has restored the last track of the source file LU, whichever is smaller. (The question is not asked if the DE option is specified, but the condition is reported.)

When restoring to a destination disc LU with more tracks than the source disc LU, you must do a FMGR MC command and specify the number of tracks restored from the source disc LU to allow FMGR to locate the correct directory tracks.

Examples

In the following example, the SE (selective restore) option is specified, and PRSTR prompts for the file/LU pairs. In this example, the first file is continued on another volume of the tape set, so PRSTR issues the MOUNT NEXT TAPE message and waits. Note that the file/LU pairs are specified separated with commas. If the list were to continue on a second line, the first line would end with a comma. The XXXXX-XXXXX entry in the file header display signifies the PSAVE and PRSTR software part numbers.

```
:RU,PRSTR,,12,,SE,1

 ENTER FILE:LU PAIRS
 2:28,3:28

 CREATED USING: PSAVE  92084-16656 REV.2121 <810905.1311>
 READ USING:    PRSTR  92084-16657 REV.2121 <810714.0904>
 TAPE NUMBER:     2    SAVE FILE:   2    USER: BILL.JT
 PROGRAM: PSAVE        OPTIONS:   MU     DATE: WED,9 SEPT. 1981
 DISC LU:    28        TITLE: PBU TESTING PRSTR
 SECTION:     1        (TRK     0   SEC     0)

 OK TO PROCEED? (YE/NO)
 ye
 RESTORING DISC LU 28
```

```
MOUNT NEXT TAPE
ENTER "GO" WHEN READY ("PA" TO SUSPEND)
pa

ENTER GO,PRSTR WHEN READY
go,prstr

RESTORING DISC LU 28
RESTORED DISC LU 28

CREATED USING:  PSAVE  92084-16656 REV.2121  <810905.1311>
READ USING:     PRSTR  92084-16657 REV.2121  <810714.0904>
TAPE NUMBER:      2    SAVE FILE:   3    USER: BILL.JT
PROGRAM: PSAVE         OPTIONS:   MU     DATE: WED,9 SEPT. 1981
DISC LU:    28         TITLE:  PBU TESTING PRSTR
SECTION:     2         (TRK      1   SEC     0)

OK TO PROCEED? (YE/NO)
ye
RESTORING DISC LU 28
RESTORED DISC LU 28
PRSTR NORMAL END OF JOB
:
```

The following example illustrates two error conditions that can arise to abort PRSTR: source disc mounted, and PSPAR not scheduled. In both cases, the condition is corrected and PRSTR rescheduled. PRSTR is invoked in this example to perform a selective restore (SE option) with verify (VE option). In the third call to PRSTR, both the tape LU and the hard-copy LU parameters are defaulted; the tape LU defaults to LU 8 (magnetic tape transport) and the hard-copy LU defaults to the log terminal. The DE option (suppress file-header verification and OK TO CONTINUE prompts) also is specified in the third call to PRSTR.

```
:prstr
 ENTER OPTIONS
 ?
 ALLOWED OPTIONS (RSTR):
 (1) VE => VERIFY
 (2) DE => DEPRESS USER QUESTIONS
 (4).LU => LU SAVE/RSTR/COPY
 (4) UN => UNIT (FULL VOL) SAVE/RSTR/COPY
 (4) PB => PUSHBUTTON UNIT IMAGE SAVE/RSTR
 (4) SE => SELECTIVELY RESTORE LUS

 THE NUMBERS IN ()'S REPRESENT CATEGORIES AND
 ONLY ONE OUT OF EACH CATEGORY CAN BE USED.
```

```
 ENTER OPTIONS
 vese
 ENTER 'FILE:LU' PAIRS
 1:37
 ENTER TAPE LU
 8
 ENTER HARD COPY LU
 1

 CREATED USING: PSAVE 92084-16656 REV.2121 <810905.1311>
 READ USING:    PRSTR 92084-16657 REV.2121 <810714.0904>
 TAPE NUMBER:  1  SAVE FILE:    1  USER: BILL.JT
 PROGRAM: PSAVE   OPTIONS: VE  LU  DATE: 10:36 AM TUE.,18 MAY,1982
 DISC LU:    37   TITLE: This is my private cartridge.
 SECTION:     1   (TRK      0  SEC      0)
 OK TO PROCEED? (YE/NO)   ye
 LU    37 IS MOUNTED.
 DISC LU IS MOUNTED
 PRS68 ERROR#23                    *disc-mounted error; PRSTR exits
 REPORTING MODULE='MANDL '.
:dc,-37                            *run DC to dismount LU
 DISC CRN    PV  LU  37 INACTIVE
:prstr                             *rerun PRSTR
 ENTER OPTIONS
 vese
 ENTER 'FILE:LU' PAIRS
 1:37
 ENTER TAPE LU
 8
 ENTER HARD COPY LU


 CREATED USING: PSAVE 92084-16656 REV.2121 <810905.1311>
 READ USING:    PRSTR 92084-16657 REV.2121 <810714.0904>
 TAPE NUMBER:  1  SAVE FILE:    1  USER: BILL.JT
 PROGRAM: PSAVE   OPTIONS: VE  LU  DATE: 10:36 AM TUE.,18 MAY,1982
 DISC LU:    37   TITLE: This is my private cartridge.
 SECTION:     1   (TRK      0  SEC      0)
 OK TO PROCEED? (YE/NO)   ye
 VERIFING DISC LU W/SPARING
 PSPAR SCHEDULE FAILURE
 PRS68 ERROR#43                      *PSPAR not-scheduled error;
 REPORTING MODULE='SPARE '.          *PRSTR exits
:rp,pspar                            *RP PSPAR
:prstr                               *sigh, re-rerun PRSTR
 ENTER OPTIONS
 vesede
 ENTER 'FILE:LU' PAIRS
 1:37
 ENTER TAPE LU
```

```
TAPE LU DEFAULTED TO      8
ENTER HARD COPY LU


CREATED USING: PSAVE 92084-16656 REV.2121 <810905.1311>
READ USING:    PRSTR 92084-16657 REV.2121 <810714.0904>
TAPE NUMBER:  1  SAVE FILE:    1  USER: BILL.JT
PROGRAM: PSAVE   OPTIONS: VE  LU  DATE: 10:36 AM TUE.,18 MAY,1982
DISC LU:    37   TITLE: This is my private cartridge.
SECTION:     1   (TRK      0  SEC      0)
VERIFING DISC LU W/SPARING
RESTORING DISC LU       37
RESTORED  DISC LU       37
PRSTR NORMAL END OF JOB
:
```

PCOPY

The utility PCOPY allows you to execute a fast disc-to-disc copy operation. The source and destination disc LUs need not be on the same disc unit, but they must have the same track size. (Note, however, that CS80 tracks are logical tracks, not physical tracks; thus a CS80-to-CS80 copy is valid regardless of the physical track size, as long as the logical track sizes are identical.)

Invoke PCOPY with the runstring

```
[RU,]PCOPY[,input],srce LU,dest LU[,VE[,hcpy]]
```

where

input — is the LU (or file) from which the parameter inputs are to be read. If input is from a file or a non-interactive device, no other parameters may be specified in the runstring. The default is to the log device.

srce LU — is the LU of the disc subchannel to be copied.

dest LU — is the LU of the disc subchannel to which the data is to be copied.

VE — is the verify option. When this option is selected, PCOPY spares the destination subchannel tracks in a prepass over the disc before data is written. The default is to suppress verify. PSPAR must be loaded into memory for this option.

hcpy — is the LU of the device on which the read or verify errors are logged. The default is to the log device.

Calling PCOPY Interactively

To run PCOPY interactively, just enter the utility call PCOPY or RU,PCOPY. Enter a valid response to each prompt or enter a carriage return to use the default value. If a response is not valid, an appropriate error message is issued and the prompt is repeated. You may enter AB, /A or /E to any of the prompts to terminate the utility. PCOPY prompts as

Prompt	Response
ENTER OPTIONS	Enter VE if the verify option is desired. Enter a carriage return to bypass the VE option.
ENTER SOURCE DISC LU	Enter the LU of the disc subchannel from which the data is to be copied. This parameter may not be defaulted.
ENTER DEST DISC LU	Enter the LU of the disc subchannel to which the data is to be copied. This parameter may not be defaulted.
ENTER HARD COPY LU	Enter the LU of the device on which the disc copy or verify errors are to be logged. Enter a carriage return to specify the default LU, your log device.

Once PCOPY has started copying the data, you may terminate the utility by entering break mode (by hitting any key on the keyboard) and entering BR,PCOPY. PCOPY should not be terminated with the OF command.

CAUTION

> Be aware that the directory tracks might not be copied when a disc LU with FMGR files and directory tracks is copied to a destination disc LU with fewer tracks than the source disc LU. If the directory tracks are not written, the copy operation is invalid.

If the source disc LU and the destination disc do not have the same number of tracks, PCOPY issues a warning and waits for your decision to continue or to abort the copy. If you respond YE to continue, PCOPY copies tracks until it has filled the last track of the destination disc or has copied the last track of the source disc, whichever is smaller.

When copying to a destination disc with more tracks than the source disc, you must do a FMGR MC (Mount Cartridge) command on the destination disc and specify the number of tracks copied from the source disc to allow FMGR to locate the correct directory tracks.

Example

In the following example, PCOPY is invoked interactively. The source and destination discs do not have the same number of tracks, so PCOPY issues a warning.

```
:RU,PCOPY

 ENTER OPTIONS
   <cr>

 ENTER SOURCE DISC LU
 16

 ENTER DEST DISC LU
 30

 ENTER HARD COPY LU
   <cr>

 203 TRACKS IN SOURCE LU
 100 TRACKS IN DEST   LU
 OK TO PROCEED? (YE/NO)
 YE

 PCOPY NORMAL END OF JOB
:
```

Off-Line Physical Backup Utility

Off-line backup is implemented as a set of stand-alone save and restore tasks, with subtasks to allow you to display the I/O configuration, to position the tape, or to specify the input device:

IO	Display current system I/O configuration.
TR	Read parameter inputs from a specified tape unit or terminal.
SA	Save disc to tape.
CO	Copy one disc subchannel to another.
RE	Restore tape file to disc.
RW	Rewind transport tape or seek to cartridge tape block 0.
FF	Forward space transport tape or cartridge tape a specified number of files.
BF	Backspace transport tape a specified number of files.

Since the off-line utility does not have access to any of the on-line system tables, information about system LUs and disc subchannel definitions must be supplied. If the operation is a restore, the utility can retrieve the subchannel definition from the disc definition record in the tape header. (Refer to the section Data Structures for details of the tape format.) If the operation is a save or copy, however, you must supply all subchannel definitions. The information can be read in from a minicartridge or tape transport. (Refer to the SA and CO commands for details of the subchannel definition process.)

Referencing Devices Off-Line

LUs for devices such as terminals, tape transports, or tape cartridges are referenced in the same manner both on-line and off-line. However, LUs that refer to off-line disc units actually are target LUs that point to the associated disc driver select codes. Each disc type has one target LU; that is, one target LU relates to the 7900 disc driver select code, one LU relates to the CS80 driver select code, and separate target LUs relate to the MAC and ICD disc driver select codes. Once the target LU has been selected, specific subchannels of that LU can be defined and operations performed selectively on the defined subchannels. A target LU must be specified under the following conditions:

1. When the destination disc type for the restore is not the same as the save disc type. For example, if you want to restore an ICD disc to a MAC disc, you must specify the target LU of the MAC disc-driver select code in the disc LU parameter of the RE command runstring.

2. When you generate an off-line utility with more than one target LU for a given type of disc. For example, if target LUs 10 and 11 both point to ICD disc select codes, you must specify the LU that points to the driver select code of the disc you intend as the destination of a restore operation or as the source (or destination) of a copy operation.

Loading the Off-Line Utility

The physical backup utilities may be supplied in either of two ways, as described below.

1. Three cartridge tapes, where:

   a. the first cartridge tape, !BCK01, contains the memory-based (off-line) operating system, reconfigurator, and the startup program BCK0F;

   b. the second cartridge tape, !BCK02, contains the off-line utilities FORMT and PSPAR that are scheduled for execution in restore or copy operations where the VErify option has been selected, or in restore operations where the pushbutton option (PB) has been selected;

c. the third cartridge tape, !BCK03, contains the off-line PRSTR, PCOPY, and PSAVE utilities.

If you want to create your own copy of the Off-Line Physical Backup Utilities, enter the command

```
TR,*BCKCT
```

from your RTE-6/VM Primary System and follow the directions given in the system response to the command.

2. One magnetic tape reel that contains all of the files !BCK01, !BCK02, and !BCK03, together with a cartridge tape containing !MTLDR.

If you want to create your own copy of the off-line physical backup utilities on magnetic tape, use the transfer file *BCKMT supplied with the RTE-6/VM Primary System. Mount and load a magnetic tape and then enter the command

```
TR,*BCKMT,<mt>
```

where <mt> is the LU of your magnetic tape device. To create your own copy of the cartridge tape !MTLDR, insert a cartridge tape in the left CTU drive of a 264x type terminal (LU 4) and enter the command

```
ST,!MTLDR,4,BA
```

from your RTE-6/VM Primary System. The file !MTLDR will be copied to your cassette.

Loading Off-Line Utility from Cartridge Tape

The following steps should be performed in the order given to load the Off-Line Physical Backup Utilities from cartridge tape into memory.

1. Place the cassette labeled !BCK01 in the left CTU of your 264x terminal (LU 4) and load the memory-based operating system as follows:

HALT the computer.

Select and CLEAR the S-Register, then

set bits 15-14 to the location of the Cassette Loader ROM.

set bits 11-6 to the 264x terminal select code.

set bit 5 (to indicate a slow boot is desired).

Press STORE, PRESET, IBL, then PRESET again.

Press RUN. A HLT 77 should occur (102077 octal displayed).

2. You are now ready to load the reconfigurator and startup program into memory. See the section Loading and Using the PBU I/O Reconfigurator for loading instructions.

Loading Off-Line Utility from Magnetic Tape

Two procedures are described below for loading the Off-Line Physical Backup Utilities from magnetic tape into memory: Procedure One should be followed if your system does not contain a Magnetic Tape Loader ROM; it includes instructions for placing a Magnetic Tape Loader, !MTLDR, from cassette tape into memory. Procedure Two applies only to systems that include the Loader ROM.

Procedure One: Loading Utilities WITHOUT a Magnetic Tape Loader ROM.

1. Place a cassette tape containing !MTLDR in the left CTU of your 264x terminal and load !MTLDR into memory as follows:

   HALT the computer.

   Select and CLEAR the S-Register, then

   set bits 15-14 to the location of the Cassette Loader ROM.

   set bits 11-6 to the 264x terminal select code.

   Press STORE, PRESET, IBL, then PRESET again.

   Press RUN. A HLT 77 should occur (102077 octal displayed).

2. Mount and place the Off-Line Backup Utility tape on-line.

3. Load the memory-based operating system as follows:

   Select and CLEAR the P-Register, then

   set bit 1 (the P-Register now contains 2 octal).

   Press STORE.

   Select and CLEAR the S-Register, then

   set bits 11-6 to the select code for the magnetic tape unit.

   Press STORE.

   Press RUN. A HLT 77 should occur (102077 octal displayed).

4. You are now ready to load the reconfigurator and startup program into memory. Refer to the section Loading and Using the PBU I/O Reconfigurator for instructions.

Procedure Two: Loading Utilities Using the Magnetic Tape Loader ROM.

1. Mount and place the Off-Line Backup Utility Tape on-line.

2. Load the memory-based operating system as follows:

   HALT the computer.

   Select and CLEAR the S-Register, then

   set bits 15-14 to the location of the Magnetic Tape Loader ROM.

   set bits 11-6 to the magnetic tape unit select code.

   set bit 5 (to indicate a slow boot is desired).

   Press STORE, PRESET, IBL, then PRESET again.

   Press RUN. A HLT 77 should occur (102077 octal displayed).

3. You are now ready to load the reconfigurator and startup program into memory, as described in the following section.

Loading and Using the PBU I/O Reconfigurator

Loading the PBU I/O Reconfigurator. The reconfigurator and startup program should be loaded into memory as follows:

1. Select and CLEAR the P-Register, then

   set bit 1 (the P-Register now contains 2 octal).

2. Press STORE.

3. Select and CLEAR the S-Register, then

   set bit 15 (to indicate that you are going to reconfigure)

   set bits 11-6 to:

   select code of the H-Series disc (ICD) you will be using, or

   0 if you will NOT be using an H-Series disc.

   set bits 5-0 to the system console select code.

4. Press STORE.

5. Press RUN.

After the reconfigurator and startup program have been read from tape, the reconfigurator will automatically start executing with display and input at your system console.

Using the PBU I/O Reconfigurator. The initial I/O configuration (known to !BCKOF) now must be changed to the configuration of your system. It is important to have your CPU I/O configuration readily available for this procedure.

The initial !BCKOF configuration is shown below, followed by directions for modifying that configuration. Note that your configuration may not be identical to that shown, due to the automatic reconfiguration that took place for your system (and H-series disc, if used) when you last set the S-Register.

```
START RECONFIGURATION
LIST DEVICE LU #?                          *Direct output to your
1                                        *  terminal
I/O RECONFIGURATION ALREADY PERFORMED:
CURRENT SELECT CODE#,NEW SELECT CODE#?
15,15          *SYSTEM CONSOLE
                                           *ICD disc if S-Register set
CURRENT I/O CONFIGURATION:

SELECT CODE 11= TBG                        *Time Base Generator card
SELECT CODE 15= EQT  1,TYPE  5             *System Console
SELECT CODE 17= EQT  8,TYPE 23             *Magnetic Tape Unit card #1
SELECT CODE 20= EQT  8,TYPE 23             *Magnetic Tape Unit card #2
SELECT CODE 50= EQT  9,TYPE 31             *7900 Disc card #1
SELECT CODE 51= EQT  9,TYPE 31             *7900 Disc card #2
SELECT CODE 52= EQT  3,TYPE 32             *79xx MAC Disc
SELECT CODE 53= EQT  4,TYPE 32             *79xx ICD Disc
SELECT CODE 54= EQT  5,TYPE 33             *79xx CS80 Disc
SELECT CODE 60= EQT  2,TYPE  0  PRMPT      *Interactive Terminal
SELECT CODE 61= EQT  7,TYPE  0  PRMPT      *Interactive Terminal
SELECT CODE 62= EQT  6,TYPE  5  PRMPT      *Interactive Terminal
I/O RECONFIGURATION? (YE/NO)
YE
```

Reconfigure your system as follows:

NOTE

It is not necessary to reconfigure the select code of your system console as it was done when you last set the S-Register.

1. You may have to reconfigure the select code of your Time Base Generator (TBG) if it is not the same as the one shown above (11 octal). In this case enter

   11,<sc>

   where <sc> is the octal select code of the TBG in your system.

2. You may have to reconfigure the select codes of your magnetic tape unit if they are not the same as the ones shown above (17 and 20 octal). Note that you must reconfigure TWO select codes for ONE magnetic tape unit. Enter

   17,<sc lc>
   20,<sc hc>

   where <sc lc> is the lower numbered select code of the magnetic tape unit and <sc hc> is the higher numbered select code.

3. You may have to reconfigure the select code of the disc you want to use. If you will use only an H-Series disc (ICD), the disc will have been configured when you last set the S-Register and does not need to be reconfigured. If you are NOT using an H-Series disc or will be using other, different discs in addition to an H-Series disc, you must reconfigure the appropriate discs as shown below:

   <!BCKOF disc sc>,<your disc sc>

   where <!BCKOF disc sc> is the select code of the type of disc (CS80, MAC, or 7900 disc) you want to reconfigure, and <your disc sc> is the select code of the same type of disc that you want to use in your system. Be aware that if you are reconfiguring a 7900 disc, there are TWO disc select codes for ONE 7900 disc unit. The lower numbered select code in the !BCKOF system should be reconfigured to the lower numbered select code in your system.

4. Enter /E to terminate the I/O reconfiguration. The current configuration is then displayed as shown below. When the OK TO PROCEED? prompt appears, review your changes to the original configuration. If the changed configuration is correct, enter YE to continue. If the changes are incorrect, enter NO and correct any errors in the configuration (steps 1 through 4 above, as required). When the configuration has been corrected, enter /E to terminate the reconfiguration, and respond YE to the OK TO PROCEED? prompt.

5. Since memory reconfiguration is not necessary, enter NO when the MEM RECONFIGURATION? (YE/NO) prompt is displayed.

6. The operating system will now start its initialization pass and then start executing the startup (and user control) program BCKOF.

A sample system configuration display is shown below.

```
SET TIME
   LU#   EQT#   SUBC#    S.C.   TYPE   DESCRIPTION
     1      1       0     15B     5B   INTERACTIVE TERMINAL
     4      1       1     15B     5B   LEFT C.T.U.
     5      1       2     15B     5B   RIGHT C.T.U.
     8      8       0     17B    17B   MAG TAPE/MASS STORAGE
     9      5       1     11B    33B   CS80 CARTRIDGE TAPE
    10      2       1     60B     0B   INTERACTIVE TERMINAL
    11      7       0     61B     0B   INTERACTIVE TERMINAL
    12      9       0     50B    31B   7900 DISC
    13      3       0     52B    32B   79XX (MAC) SERIES DISC
    14      4       0     53B    32B   79XX (ICD) SERIES DISC
    15      5       0     11B    33B   79XX (CS80) SERIES DISC
   Please enter tape-LU for reading !BCK02: n
```

Your response to the "Please enter tape-LU" prompt above is determined by the tape media on which the Physical Backup Utilities are supplied. If the utilities are supplied on cartridge tapes, refer to the next section for loading instructions; if the utilities are supplied on magnetic tape, refer to the section Loading Utilities from Magnetic Tape.

Loading Utilities from Cartridge Tape

After the memory-based operating system, reconfigurator, and startup program have been read in from cartridge tape, enter LU 4 (if the cartridge is to be installed in the left CTU of the system console) or LU 5 (if the cartridge is to be installed in the right CTU of the system console).

A prompt now appears to insert the next cartridge tape, !BCK02, and to enter a space character and carriage return. (See the description of the contents of !BCK02 at the beginning of this section.) When the cartridge is properly inserted, the Off-Line Backup Utilities startup program, BCKOF, reads in two utilities that are scheduled by a save, copy, or restore operation.

After these are loaded into memory, you are prompted to insert the next cartridge tape, !BCK03, and to enter a space character and carriage return. This cartridge tape contains the utility programs that perform the three operations, REstore, COpy, and SAve. Note that these programs are not read into memory until the desired operation is specifically requested.

At this point, the following prompt is displayed to indicate that the physical backup utilities are available for the off-line save, restore, or copy operations:

```
OFFLINE DISC-TAPE SAVE/RESTORE/COPY SYSTEM, !BCKOF
TASK?
```

You now may specify any of the off-line utilities by entering the associated response to the TASK? prompt. Refer to the individual utility runstring descriptions in the remainder of this chapter.

Loading Utilities from Magnetic Tape

After the memory-based operating system, reconfigurator, and startup program have been read in from magnetic tape, enter LU 8 to load all of the programs for the different off-Line operations into memory from the tape. When the load is complete, the following prompt is displayed:

```
OFFLINE DISC-TAPE SAVE/RESTORE/COPY SYSTEM, !BCKOF
TASK?
```

You may now specify any of the off-line utilities by entering the associated response to the TASK? prompt. Refer to the individual utility runstring descriptions in the remainder of this chapter.

The TASK? prompt indicates that the off-line utility set has been loaded and you can begin specifying the save, restore, or copy tasks as defined in the following sections. If an error occurs when processing any task, the processing of that task is aborted. The off-line utility issues the appropriate error message and returns to the TASK? prompt.

Any task in process can be halted by entering break mode (hitting any key on the keyboard) and entering BR,task. The OF command should not be used to abort any of the off-line tasks.

Off-Line System Console Operation

When using the physical backup utilities off-line, you can enter the non-session operating environment by exiting the utility (/EX in response to the TASK? prompt) and striking any key on the system console. When the asterisk prompt (*) appears, you can execute any RTE-IVB system command that is valid in your system configuration. Refer to the RTE-IVB Terminal User's Reference Manual for a description of the system commands.

Display I/O Configuration (IO)

Entering IO as the task causes the current I/O configuration to be displayed in the following form. The configuration shown is presented as a format example only; I/O assignments will vary among systems.

```
TASK?

IO

LU#   EQT# SUBC#   S.C.  TYPE  DESCRIPTION
 1      1     0     25B    5B  INTERACTIVE TERMINAL
 4      1     1     25B    5B  LEFT CTU
 5      1     2     25B    5B  RIGHT CTU
 6      6     0     20B   12B  LINE PRINTER
 8      8     0     23B   23B  MAG TAPE/MASS STORAGE
 9      2     1     15B   33B  CS80 CARTRIDGE TAPE
11      7     0     25B    0B  INTERACTIVE TERMINAL
12      2     0     15B   33B  79XXA (CS80) SERIES DISC
13      3     0     16B   32B  7905  (MAC) SERIES DISC
14      4     0     21B   31B  7900 DISC
15      5     0     77B   32B  7905H (ICD) SERIES DISC
TASK?
```

Transfer to Input LU (TR)

When the utility is initially booted, inputs are expected from the console device named in the configuration procedure. If a TRansfer command is entered, as

```
TR,lu
```

the utility transfers to the specified LU to read subsequent parameter inputs. A TR command can be entered in response to any prompt at the console, or can be specified in the input parameter of the SA, RE, or CO commands. TR commands also can be nested; that is, if another TR command is encountered in the input, the preceding prompt is repeated and the utility transfers to the named LU for the response.

If an error is encountered in the input or an end of file is read on the transfer LU, control returns to the console device (a TR,1 command is effectively executed) with an appropriate error message. The current activity is aborted and the TASK? prompt is issued.

If the transfer LU is the same as the destination LU on a save or the source LU on a restore, the utility will request that you mount the tape and enter GO when ready. The effect is that of a TR,1 command, and subsequent parameter input is expected to come from the console device.

Restore Tape File (RE)

The RE command restores an LU save or unit save tape file to disc. The off-line utility also recognizes the RE command if it is entered as "PR" or "PRSTR". When restoring to disc, be sure that the saved-file disc type and the restore disc type are compatible; refer to Table 6-1 to determine disc compatibility before specifying the destination disc.

The RE command takes the form

```
RE[,input[,targ LU[,srce LU[,file#[,opts[,hcpy]]]]]]
```

where:

input is the LU of the device from which the parameter inputs are to be read. If input is a non-interactive device, no other parameters may be specified in the runstring. The default is to the system console, LU 1.

targ LU is the LU of the target disc. Refer to the section Referencing Devices Off-Line for a description of the use of this parameter. There is no default for this parameter.

srce LU is the LU of the tape transport or CS80 cartridge from which the save file is to be read. The default is to tape transport LU 8.

file# is the integer number of the file to be read. File# - 1 files will be skipped before the read begins. Default is to the current tape position for tape transports, or file #1 for CS80 cartridges.

opts are any of the following two-character ASCII option codes. Options can be specified in any order, and may not have any intervening characters (including blanks) between options. If no options are specified, the defaults are no verify, read subchannel definitions from the tape file record (TA), prompt for file verify, and LU restore if the save was an LU or MU save or UN restore if the save was a unit save. If you respond with "??" RE will display a list of available options.

VE Turn on data verify option. Destination track sparing is performed only when this option is selected.

DE Do not prompt for verification of tape file.

UN Unit restore. This is the default if the save tape contains a unit save.

LU LU restore. This is the default if the save tape contains an LU or MU save.

SE Selective restore from a unit save, or multiple LU restores in one operation. This is not a valid option if the save was a unit save in pushbutton-restorable format, or a unit save in LSAVE, USAVE, or SAVE formats.

PB Restore a CS80 cartridge in pushbutton-restorable format. This is a unit restore.

TA Restore according to the subchannel definitions on the tape. This is the default option. The subchannel definitions are displayed at the console as they are read.

UD Read subchannel definitions from LU defined by input parameter.

hcpy is the LU of the device on which errors are loggged and track map definitions are echoed. The default is to the system console, LU 1.

When the UD option is selected, you are prompted to enter all subchannel definitions. These may be entered interactively or by TRansferring to a tape file containing the definitions. Where the destination disc type is not the same as the source disc listed in the tape header, the subchannels may need to be redefined. Subchannel definitions are also forced when the operating systems are not the same for the save and restore.

RE prompts for the subchannel definitions in one of the following formats. The same prompt is used for both the source and destination disc subchannel definitions, with the word SOURCE or DESTINATION in the DEFINE TRACK MAP statements.

For CS80 discs, the following prompt sequence is issued.

```
DEFINE (srce,dest) TRACK MAP
DEFINE SUBCHANNELS, LU =      n
DEVICE (MODEL,HP-IB ADDRESS,UNIT,VOLUME)
mmmm,aaaaa,uu,vvv

xxxxxx BLOCKS REMAINING
SUBCHANNEL   nn (TRACKS,BLOCKS/TRACK)
ttt,bbb
```

For MAC/ICD source and destination discs, the following prompt sequence is issued. (UNIT# is included in the definition prompt for ICD discs only.)

```
DEFINE (srce,dest) TRACK MAP
DEFINE SUBCHANNELS,LU = n

MODEL,#TRKS,FIRST CYL,HEAD,#SURF,ADDRESS,#SPARES,UNIT#:
00?
mmm,tt,c,h,ss,aaaaaa,sss,u
```

For 7900 discs, the following prompt sequence is issued.

```
DEFINE (srce,dest) TRACK MAP
DEFINE SUBCHANNELS,LU =     n
FIRST 7900 SUBCHANNEL NUMBER (0=<n<=s)
0
#TRACKS, 1ST CYL
SUBCHANNEL 00?
ttt,c
```

Unless the UN restore is selected, only the current track map definition is displayed as each tape file record is encountered. With the UN option selected, all track map definitions are displayed. Since you must specify all subchannels prior to the first restore when the UD option is selected, The SE option should be used with the UD option only when all of the source tape and destination disc track map information is known. If this information is not known, the restore should be performed as multiple tasks using the LU option.

Calling RE Interactively

To run interactively, enter RE; the utility prompts for the parameters in the following sequence. Enter a carriage return to default any of the optional parameters.

Prompt	Response
ENTER OPTIONS	Enter any of the valid option codes. Since the UN, LU, SE, and PB options are mutually exclusive, and TA and UD are mutually exclusive, only four options are possible, at most. If you respond "?" or "??" RE displays a list of all allowable options.
ENTER DEST DISC LU	Enter the target LU of the disc to which the files are to be restored.

If you select the SE option:

ENTER FILE:SUBCHANNEL PAIRS	Enter the file number and target subchannel to which the file is to be restored. If the track map definitions come from tape (TA option), the subchannel numbers given are irrelevant and are ignored. The subchannel numbers may be given sequentially starting with 0, and the track maps may then be defined in the same order. Multiple entries should be separated by commas. To continue on the next line, end the current line with a comma.
ENTER TAPE LU	Enter the LU of the tape unit containing the saved file. The default is to the tape transport, LU 8.

If you select the LU option:

ENTER FILE NUMBER	Enter the tape file number of the file to be restored. The default is file #1 on CS80 cartridges and the current tape position on tape transports.
ENTER HARD COPY LU	Enter the LU number of the device on which errors will be logged and track map definitions displayed. The default is the system console, LU 1.

If you do not select the DE option:

```
 - tape header -
OK TO PROCEED?(YE/NO)
```

Enter YE if this is the correct tape. If your response is NO, the ENTER FILE NUMBER prompt is repeated.

When restoring to any disc unit, the protect switch must be turned off and the format switch must be turned on. If either condition is not met, an error message is printed and the utility prompts with:

```
TYPE 'GO' WHEN READY ('PA' TO SUSPEND)
```

When the unit switch setting is correct, enter GO and the restore continues. If you enter PA, the utility suspends itself after printing:

```
TYPE  GO,PRSTR  TO CONTINUE
```

Example

In the following example, note that RE is invoked with the call "PRSTR"; the off-line utility also accepts a call of the form "PR". In this example REstore is called interactively, with the first response a "?" to request a listing of allowable options (either a single "?" or double "??" can be entered as the request). The VE, TA AND SE options then are specified and the file:subchannel pairs are defined for the selective restore operation. The tape header is then displayed, followed by the subchannel definition and the OK TO PROCEED? prompt. The response is YE and the restore operation begins by first verifying the destination disc and sparing tracks if required.

While the restoration is in process, the message RESTORING DISC LU 2 (in this example) is displayed. When the restoration is complete the RESTORED DISC LU 2 message is displayed. Since this example is a selective restore of two discs, the tape header of the second disc is then displayed and the sequence of prompt/verify/restore is repeated, followed by the TASK? prompt.

In the example, the TASK? prompt response is a garble of characters to which REstore responds with the UNRECOGNIZED COMMAND message. Following the available command display (?? response), the EX command is issued and BCKOF terminates. By entering a carriage return (any keyboard character can be used), the system enters non-session mode and returns the * prompt. At this point, any of the RTE-IVB system commands that are valid for your configuration can be executed. To return to the off-line backup utilities, enter a carriage return and enter RU,BCKOF, as shown in the example.

```
TASK?
PRSTR
ENTER OPTIONS
?
ALLOWED OPTIONS (RSTR):
(1) VE => VERIFY
(2) DE => DEPRESS USER QUESTIONS
(3).TA => USE TAPE SUBCHANNEL DEFS
(4) UD => GET USER SUBCHANNEL DEFS
(4).LU => LU SAVE/RSTR/COPY
(4) UN => UNIT (FULL VOL) SAVE/RSTR/COPY
(4) PB => PUSHBUTTON UNIT IMAGE SAVE/RSTR
(4) SE => SELECTIVLY RESTORE LUS

THE NUMBERS IN ()'S REPRESENT CATEGORIES AND
ONLY ONE OUT OF EACH CATEGORY CAN BE USED.

ENTER OPTIONS
VETASE
ENTER DEST DISC LU
15
ENTER 'FILE:SUBCHANNEL' PAIRS
1:0,2:1
ENTER TAPE LU
<cr>
TAPE LU DEFAULTED TO      8
ENTER HARD COPY LU
<cr>

CREATED USING:  PSAVE 92084-16656 REV.2121 <820623.1038>
READ USING:     PRSTR 92084-16657 REV.2121 <820623.1038>
TAPE NUMBER:      1   SAVE FILE:      1  USER: FUBAR.TEST

PROGRAM: PSAVE  OPTIONS: VE MU  DATE: 5:16 PM  MON., 22 FEB.,1982
DISC LU:     2    TITLE: PS#604 CS80 REV.2226
SECTION:     1    (TRK      0  SEC      0)
OK TO PROCEED? (YE/NO)  YE
VERIFING DISC LU W/SPARING
RESTORING DISC LU        2
RESTORED  DISC LU        2

CREATED USING:  PSAVE 92084-16656 REV.2121 <820623.1038>
READ USING:     PRSTR 92084-16657 REV.2121 <820623.1038>
TAPE NUMBER:      1   SAVE FILE:      1  USER: FUBAR.TEST

PROGRAM: PSAVE  OPTIONS: VE MU  DATE: 5:16 PM  MON., 22 FEB.,1982
DISC LU:     3    TITLE: PS#604 CS80 REV.2226
SECTION:     1    (TRK      0  SEC      0)
OK TO PROCEED? (YE/NO)  YE
VERIFING DISC LU W/SPARING
RESTORING DISC LU        3
```

```
RESTORED  DISC LU        3
TASK?
LKSJDF
UNRECOGNIZED COMMAND (ENTER ?? FOR HELP)
TASK?
??
IO - Display current I/O configuration.
SA - Save disc regions to tape.
RE - Restore tape file to disc.
CO - Copy one disc subchannel to another.
RW [,lu] - Rewind magnetic tape, seek to
    CTD block zero. (Default lu=8)
FF [,[lu],[n]] - Forward space magnetic tape or
    CTU n files. (Default lu=8, n=1)
BF [,[lu],[n]] - Backspace magnetic tape n files.
  NOP for CTU. (Default lu=8, n=1)
TR,lu - Read commands from 'lu'.
/E, EN, or EX - Terminate program.
TASK?
EX
BCKOF TERMINATED BY USER
<cr>
*RU,BCKOF

OFFLINE DISC-TAPE SAVE/RESTORE/COPY SYSTEM, !BCKOF
TASK?
```

Disc-to-Disc Copy (CO)

The CO command allows you to copy one disc subchannel to another according to the subchannel information you enter. The off-line utility also recognizes this command if it is entered as "PC" or PCOPY". Optionally, the subchannel definitions may come from a tape file. The CO command takes the form

```
CO[,input],targ LU,dest LU[,VE[,hcpy]]
```

where

input is the LU of the device from which the parameter inputs are to be read. If input is a non-interactive device, no other parameters should be specified in the command. The default is to the system console, LU 1.

targ LU is the LU of the target disc from which the subchannel is to be copied. This parameter is required for CO to select the format of the subchannel definition prompt set.

dest LU is the LU of the disc to which the subchannel is to be copied. This parameter is required for CO to select the format of the subchannel definition prompt set.

VE Turn on the data verify option. Track sparing is done only when this option is specified.

hcpy is the LU of the device on which copy errors are to be logged. The default is to the system console, LU 1.

CO prompts for the subchannel definitions in one of the following formats. The same prompt is used for both the source and destination disc subchannel definitions, with the word SOURCE or DESTINATION in the DEFINE TRACK MAP statements. After the subchannels are defined, CO begins the copy task and displays the message COPYING. At the end of the task, CO issues the message PCOPY NORMAL END OF JOB and exits to the TASK? prompt.

For CS80 discs, the following prompt sequence is issued.

```
DEFINE (srce,dest) TRACK MAP
DEFINE SUBCHANNELS, LU =      n
DEVICE (MODEL,HP-IB ADDRESS,UNIT,VOLUME)
mmmm,aaaaa,uu,vvv

xxxxxx BLOCKS REMAINING
SUBCHANNEL 0    (TRACKS,BLOCKS/TRACK)
ttt,bbb
```

For MAC/ICD source and destination discs, the following prompt sequence is issued. (UNIT# is included in the definition prompt for ICD discs only.)

```
DEFINE (srce,dest) TRACK MAP
DEFINE SUBCHANNELS,LU = n

MODEL,#TRKS,FIRST CYL,HEAD,#SURF,ADDRESS,#SPARES,UNIT#:
00?
mmm,tt,c,h,ss,aaaaaa,sss,u
```

For 7900 discs, the following prompt sequence is issued.

```
DEFINE (srce,dest) TRACK MAP
DEFINE SUBCHANNELS,LU =     n
FIRST 7900 SUBCHANNEL NUMBER (0=<n<=s)
0
#TRACKS, 1ST CYL
SUBCHANNEL 00?
ttt,c
```

When a CS80 copy operation is to begin at a disc location other than block 0 (as does the first disc subchannel on a system disc), enter a negative value as the TRACKS parameter of the subchannel definition. This, in effect, creates a "hole" on the disc and allows you to access the desired block for the copy. As an example, if the copy is to begin at block 33600 of a model 7908 CS80 disc, the sequence of definitions would be

```
64750 BLOCKS REMAINING
SUBCHANNEL 0 (TRACKS,BLOCKS/TRACK)
-300,48                                 *sets starting block of
                                        *next subchannel to 14400

50350 BLOCKS REMAINING
SUBCHANNEL 0 (TRACKS,BLOCKS/TRACK)
-200,48                                 *sets starting block of
                                        *next subchannel to 24000

40750 BLOCKS REMAINING
SUBCHANNEL 0 (TRACKS,BLOCKS/TRACK)
-200,48                                 *sets starting block of
                                        *next subchannel to 33600

31150 BLOCKS REMAINING
SUBCHANNEL 0 (TRACKS,BLOCKS/TRACK)
200,48                                  *positive value defines
                                        *subchannel at block 33600
```

The negative track values can be taken from the subchannel definitions contained in your system generation listing.

Refer to the RTE-6/VM System Manager's Reference Manual for further details of the CS80 disc configuration.

To run the copy command interactively, enter CO; the utility prompts you for the parameters in the following sequence.

ENTER OPTIONS	If verification and track sparing are desired, enter VE to specify the data verify option.
ENTER SOURCE LU	Enter the target disc from which the subchannel is to be copied. The source and destination can be the same disc LU.
ENTER DEST LU	Enter the disc to which the subchannel is to be copied.
ENTER HARD COPY LU	Enter the device that is to receive data error messages.

After the utility has been specified interactively, CO prompts for the subchannel definitions as above.

Save Disc to Tape (SA)

The off-line save function may be used to save disc information in LU save, unit save format, or pushbutton-restorable format. Note that if using the CTD, it is more efficient to save the disc on-line since the CTD cannot be cached off-line. (Refer to the section Tape Handling for a description of the CTD caching scheme.)

Invoke SA using the runstring

```
SA[,input],srce LU[,dest LU,[file#[,opts[,hcpy[,title]]]]]
```

where

input — is the LU of the device from which the parameter inputs are to be read. If input is a non-interactive device, no other parameters may be specified. The default is to the system console, LU 1.

srce LU — is the LU of the disc subchannel from which the information is to be saved. This parameter is required for SA to select the format of the subchannel definition prompt set.

dest LU — is the LU of the tape transport or CS80 cartridge to receive the saved data. The default is to the tape transport, LU 8.

file# — is the integer number to be assigned to the saved file. This parameter specifies the start location on the tape for the save file. Before saving, file# - 1 files are skipped from the start of the tape. The default is to the current tape position on tape transports, and file#1 on CS80 cartridges. This parameter is meaningless if the option is MU, and will cause an error if specified.

opts — are any of the following two-character ASCII option codes. Options must be specified with no intervening characters (including blanks). If no options are specified, the default is to LU save and no verify.

VE Turn on data verify option. Track sparing is performed only when this option is selected.

UN Unit save.

LU LU save.

PB Unit save in CS80 pushbutton-restorable format. The destination LU must by CS80 cartridge, and the source LU must be a CS80 disc.

hcpy is the LU of the device on which information about the save is to be printed. The default is to the system console, LU 1.

title is the title (to a maximum of 40 ASCII characters) that will be placed in the tape header. The title may be entered in either upper- or lower-case letters or a combination of both.

SA prompts for the subchannel definitions in one of the following formats.

For CS80 discs, the prompt sequence is

```
DEFINE SUBCHANNELS, LU =    n
DEVICE (MODEL,HP-IB ADDRESS,UNIT,VOLUME)
mmmm,aaaaa,uu,vv
NUMBER OF BLOCKS ON DEVICE
bbbbb
```

The "NUMBER OF BLOCKS ON DEVICE" prompt appears only if the model number is not recognized.

For MAC and ICD discs, the following prompt sequence is issued. (UNIT# is included in the definition prompt for ICD discs only.) The prompt is repeated for each subchannel on the disc, or until you enter /E to indicate the end of the subchannel definitions for the save.

```
DEFINE SUBCHANNELS, LU =    n

MODEL,#TRKS,FIRST CYL,HEAD,#SURF,ADDRESS,#SPARES,UNIT#
00?
mmmm,ttt,c,h,ss,aaaaaa,sss,uu
```

For 7900 discs, the following prompt sequence is issued. The prompt is repeated for each subchannel on the disc, or until you enter /E to indicate the end of the subchannel definitions for the save.

```
#TRACKS,  1ST CYL,
SUBCHANNEL     00?
ttt,c
```

Calling SA Interactively

To run the save command interactively, enter SA. The utility prompts for the parameters in the following sequence. Enter a carriage return to specify the default conditions. If you enter an invalid response, the question is repeated.

Prompt	Response
ENTER OPTIONS	Enter any of the above option codes. Since the UN, LU, PB and MU options are mutually exclusive, only two options are possible, at most. If you respond with "??", SA will display a list of available options.
ENTER SOURCE DISC LU	Enter the LU of the disc to be saved.

If you select the MU option:

ENTER DISC SUBCHANNEL	Enter the subchannels to be saved, separated with commas. To continue on the next line, end the line with a comma. Normally, the entries will be sequential starting with 0, but you can specify subchannels (up to a total of 64) in any order.
ENTER TAPE LU	Enter the LU of the tape transport or CS80 cartridge. The default is to the tape transport, LU 8.
ENTER FILE NUMBER	Enter the file number at which the save is to begin. The The default is to the current tape position for tape transports, and file #1 for CS80 cartridges.
ENTER HARD COPY LU	Enter the LU of the device to record information concerning the tape saves. The default is to the system console, LU 1.
ENTER TITLE	Enter the title, up to 40 ASCII characters, to be stored in the tape header.

After the utility has been specified, SA prompts for the subchannel definitions as above.

Before beginning the save operation, SA checks that the tape is on-line and write enabled. If either condition is not met, an error message is printed and the utility prompts with:

```
TYPE 'GO' WHEN READY ('PA' TO SUSPEND)
```

When the switch settings are correct, enter GO and SA continues. If you enter PA, SA suspends itself after printing the message

```
TYPE  GO, PSAVE TO CONTINUE
```

Example

In the following example, SA is invoked interactively and the defaults are specified by entering a carriage return in response to the prompts. The UN unit save option is selected and the tape LU is defaulted to tape transport LU 8. The file number is defaulted and is determined by the current tape position of the tape transport (in this example the tape is at file position 3). The title OFFLINE 7900 UNIT SAVE is assigned to both files. The subchannel definition prompt is then issued.

Since only two subchannels are to be saved, the SUBCHANNEL 02? prompt response is /E to indicate the end of the definition. The SAVING SUBCHANNEL message, followed by the tape header, then is displayed as the save is in process. At the end of each save, the SUBCHANNEL SAVED message is issued. When the save is complete, SA exits and the TASK? prompt is issued.

```
TASK?
SA

ENTER OPTIONS
UN

ENTER SOURCE DISC LU
14

ENTER TAPE LU
   <cr>
TAPE LU DEFAULTED TO        8
ENTER FILE NUMBER
   <cr>
ENTER HARD COPY LU
   <cr>
ENTER TITLE
OFFLINE 7900 UNIT SAVE

DEFINE SUBCHANNELS,LU=    14
```

```
#TRACKS,  1ST CYL,
SUBCHANNEL    00?
203,0

SUBCHANNEL    01?
203,0

SUBCHANNEL    02?
/E

SAVING DISC SUBCHANNEL  0 TO FILE   3 TAPE   1

TAPE NUMBER:  1  SAVE FILE:   3  USER:
PROGRAM: PSAVE   OPTIONS:    UN   DATE:  8:00 AM TUE, 14 JUL,1981
DISC LU:    14      TITLE: OFFLINE 7900 UNIT SAVE
SECTION:     1      (TRK      0  SEC      0)

SUBCHANNEL  0 SAVED ON FILE   3 TAPE   1

SAVING DISC SUBCHANNEL  1 TO FILE   4 TAPE   1

TAPE NUMBER:  1  SAVE FILE:   4  USER:
PROGRAM: PSAVE   OPTIONS:    UN   DATE:  8:00 AM TUE, 14 JUL,1981
DISC LU:    14      TITLE: OFFLINE 7900 UNIT SAVE
SECTION:     1      (TRK      0  SEC      0)

SUBCHANNEL  1 SAVED ON FILE   4 TAPE   1

PSAVE NORMAL END OF JOB
TASK?
```

Tape Movement Functions

The off-line utility also includes the following tape-movement control commands:

RW[,lu]	Rewind tape to the beginning.
FF[,lu[,n]]	Forward space tape n files. The default is FF LU 8 one file.
BF[lu[,n]]	Backspace tape n files. The default is BF LU 8 one file.

Help Function

A list and brief description of all available commands can be invoked by entering ?? or HE in response to the TASK? prompt, as

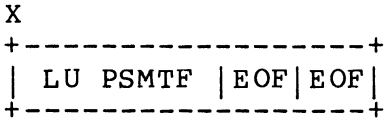
```
TASK?

??
IO - Display current I/O configuration.
SA - Save disc regions to tape.
RE - Restore tape file to disc.
CO - Copy one disc subchannel to another.
RW [,lu] - Rewind magnetic tape, seek to
      CTD block zero.  (Default lu=8)
FF [,[lu],[n]] - Forward space magnetic tape or
     CTU n files. (Default lu=8, n=1)
BF [,[lu],[n]] - Backspace magnetic tape n files.
     NOP for CTU. (Default lu=8, n=1)
TR,lu - Read commands from 'lu'.
/E, EN, or EX - Terminate program.
```

Data Structures

This section describes the data structures used when saving discs to magnetic tapes or to CS80 cartridge tapes (CTD). The data structures differ somewhat between the two device types, principally because the CTD is a fixed-block-length data-recording device and the 800/1600-bpi magnetic tape transport is a variable-block-length data-recording device.

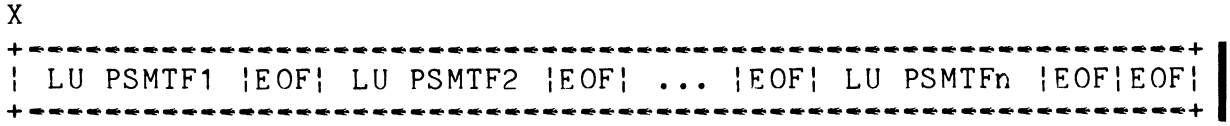
PSAVE Format on Magnetic Tape (Reel-to-Reel)

An LU save using PSAVE is written to magnetic tape in PSave Magnetic Tape File (PSMTF) format, where the end of the LU is indicated on tape by an end-of-file mark (EOF) and the end of the PSAVE operation (end of data) is indicated by a second EOF. This format is



where X (the tape position for the LU save starting point) is determined by the file number specified in the PSAVE runstring or by the current position of the tape if no file number is specified.

A UNit save is merely a series of LU PSave MT Files followed by EOFs. The save starts with the lowest number LU on the unit, proceeds through the highest number LU on the unit, and ends with a second EOF to mark the end of the PSAVE operation. A MUltiple LU save is in the same format except that the LUs are saved in the order in which they are specified in the command. The tape organization is



where

n is the number of LUs saved in the one PSAVE operation and

X (the tape position for the starting point) is determined by the file number specified in the PSAVE runstring, or is the current position of the tape if no file number is specified.

The PSave Magnetic Tape File (PSMTF) format for an LU consists of several different types of records, as shown below.

Records 1 through 6 are ASCII information (header) records:

```
CREATED USING: PSAVE xxxxx-xxxxx REV.xxxx <yymmdd.tttt>
READ USING:    PRSTR xxxxx-xxxxx REV.xxxx <yymmdd.tttt>
TAPE NUMBER:     n   SAVE FILE:      n  USER: BILL.JT
PROGRAM: PSAVE       OPTIONS: aa  aa    DATE: xxxxxxxxxxxxxxxxxx
DISC LU:    nn       TITLE: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
SECTION:     n       (TRK   nnnn  SEC   nnnn)
```

Records 7 and 8 are save and disc definition records in a format similar to those built by physical backup utilities (with the exception of the leading checksums). The content of the save definition record (record 7) is:

WORD	CONTENT
1	Checksum
2	Checksum
3	Checksum
4	Checksum
5	Tape number
6	Unused (initialized to binary zero)
7	Number of saves remaining (Unit save only)
8	Word size of tape records
9	Word size of disc definition record
10	Options set for save (MU, UN, or LU)
11	Relative file number
12	Disc LU number saved
13	Disc type (from LDTYP)
14	Section number
15	Starting track number
16	Subchannel
17-25	Unused: initialized to binary 0
26-EOR	Disc definitions for remaining saves (contained only in the first LU PSMTF of a UNit save)

The content of the disc definition record (record 8) is:

WORD	CONTENT
1	Checksum
2	Checksum
3	Checksum
4	Checksum
5-15	Disc definition for this LU (from DSCPR)
16	Subchannel 17-25 (Unused: initialized to binary 0)
26-EOR	Disc definitions for remaining saves (contained only in the first LU PSMTF of a UNit save)

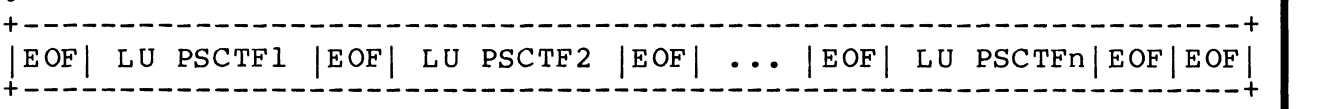
Records 9 through the end of the file contain data from the disc. Each of these records is an image of a certain number of sectors from the disc, except that the first four words of each record are used for checksums.

PSAVE Format on CS80 Cartridge Tape

The format of an LU PSave to CS80 Cartridge Tape File (PSCTF) is very similar to the PSAVE format for saves to magnetic tape. However, PSAVE places an end of file mark (EOF) in the first block of any cartridge tape that will contain PSAVE format data. This is included to provide protection from possible overlay of a disc if the pushbutton at the front of the CTD device is used. That is, when the pushbutton capability of the CTD is used on a PSAVE format tape, the EOF in the first block is read and, since this signifies the end of the file, the restore operation is immediately terminated and no data is overwritten on the disc.

The PSAVE Cartridge Tape File format for LU saves is shown below.

A UNit save is merely a series of LU PSave CT files followed by EOFs. The save starts at the lowest number LU on the unit, proceeds through the highest numbered LU on the unit, and ends with a second EOF to mark the end of the PSAVE operation. A MUltiple LU save is in the same format, except that the LUs are saved in the order in which they are specified in the command. The PSAVE Cartridge Tape File (PSCTF) format for MUltiple LU saves and UNit saves is shown below.



Note that any PSAVE operations on cartridge tape will begin in block 0 of the tape if either file number 1 is specified or if the file number is allowed to default. This file-numbering distinction is described in more detail in the section Tape Handling.

Since each record on a cartridge tape is a fixed length block of 512 words, the format of the tape files differs from that for magnetic tape. These differences in format for the PSave Cartridge Tape Files can be seen in the format of the blocks described below.

The first block of the file is a series of information (header) lines:

```
CREATED USING: PSAVE xxxxx-xxxxx REV.xxxx <yymmdd.tttt>
READ USING:    PRSTR
TAPE NUMBER:    n   SAVE FILE:      n  USER: BILL.JT
PROGRAM: PSAVE      OPTIONS: aa  aa    DATE: xxxxxxxxxxxxxxxxxx
DISC LU:    nn      TITLE: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
SECTION:     n      (TRK   nnnn  SEC   nnnn)
```

Note that the first word of this block is the length in words of the header lines. This length includes a carriage return and linefeed character that is added to the end of each line.

The save and disc definition blocks on cartridge tape (block 2 and block 3 of the file) are identical in format to the save and disc definition records on magnetic tape except that they are in fixed length (512-word) blocks.

The save definition block (block 2) format is:

WORD	CONTENT
1	Checksum
2	Checksum
3	Checksum
4	Checksum
5	Tape number
6	Unused (initialized to binary zero)
7	Number of saves remaining (UNit save only)
8	Word size of tape records
9	Word size of disc definition record
10	Options set for save (MU, UN, or LU)
11	Relative file number
12	Disc LU number saved
13	Disc type (from LDTYP)
14	Section number
15	Starting track number
16	Starting sector number
17	Operating System type (from .OPSY)
18-128	Unused (initialized to binary zero)

The content of the disc definition block (block 3) is:

WORD	CONTENT
1	Checksum
2	Checksum
3	Checksum
4	Checksum;
5-15	Disc definition for this LU (from DSCPR)
16	Subchannel number
17-25	Unused (initialized to binary 0)
26-EOR	Disc definitions for remaining saves (contained only in the first LU PSCTF of a UNit save).

Blocks 4 through the end of the file contain data from the disc. Each of these blocks is an image of 512 words of data from the disc, except that two identical checksum blocks are periodically written on the tape. These checksum blocks are written after every 480 blocks during the save operation, and at the end of a save, and at the end of tape (if the end of tape is reached). Each of the checksum blocks has the following format:

WORD	CONTENT
1-480	A one-word checksum for each corresponding preceding block. That is, word 1 is the checksum for the first block following the last checksum block or definition records for the save file, word 2 is the checksum for block 2, ... word 480 is the checksum for block 480.
481-511	Unused; initialized to binary 0.
512	Checksum for this checksum block.

Pushbutton Image Format on CS80 CTD

A pushbutton operation produces identical results whether it is performed using the front panel control on the Cartridge Tape Drive or by using the pushbutton (PB) option programmatically from the backup utilities. In both cases, the controller in the device receives a request to do a `Copy Data' operation for the entire CTD tape or the full disc.

For disc-to-CTD saves, four 128-word disc sectors are read sequentially from the disc starting with block address 0. The data are passed through the buffer of the disc controller and written sequentially to the tape in 512-word blocks, starting with CTD tape block 0. The save terminates when either the end of tape is sensed or the last block on the disc has been written to tape. If the end of disc terminates the save, an end of file is written to the tape to mark the end of the backup operation.

If you are REstoring a tape to disc, the copy terminates when the end of tape has been reached, the last block on the disc has been written, or an end of file mark is found on the tape.

When you use the backup utilities with the programmatic pushbutton option to perform the save or restore, you should specify the VE option to verify the destination data. The verification process is merely a request from the backup utilities to the disc controller to verify that all of the data on the tape/disc is correct, using the checksums internally generated with each block and stored on the tape/disc by the device controller.

With both copy methods, internal checksums moved with each block of data are used to verify (and sometimes correct) the data being read. This operation is automatic and is a function and feature of the hardware.

Error Messages

The following errors can be seen by the on-line and off-line save, restore, and copy utilities. If the scheduling program for the on-line utilities is FMGR, the error number is returned to the lP global. The utilities in which the error can be generated are indicated parenthetically following the definition of the message. Although only the on-line utilities are listed, the applicability of the error condition extends to the off-line SA, RE, and CO tasks. Those errors that can occur off-line only are identified at the start of the error-message description.

When an error results in operation abort, the error number, the name of the subroutine in which it occurred, and the utility that terminated is displayed at the log device.

1: SIZE UP THE PROGRAM!

Insufficient buffer space is available to perform the operation. (PRSTR,PSAVE,PCOPY)

2: INVALID PARAMETER

The utility terminated due to an invalid parameter input. This same error code is returned if the utility is terminated using /A or AB. (PRSTR,PSAVE,PCOPY)

3: TAPE MOUNT ERROR

The tape was not correctly mounted. (PSAVE)

4: TAPE COULD NOT BE LOCKED

The tape lock LURQ call failed. (PSAVE,PRSTR)

5: TAPE IS OFFLINE

The tape is not set to online. (PSAVE)

6: INVALID FILE NUMBER

A file number was given that was past the end of data mark. For a UN restore, this could mean that the file at which the tape is positioned is not the first file in the UN save. (PSAVE,PRSTR)

7: PUSHBUTTON [SAVE] [RESTORE] FAILED

A pushbutton save (PSAVE) or pusbutton restore (PRSTR) did not complete successfully. (PB option)

8: NO DISC PARAMETERS

A call to DSCPR returned no parameters. This is an internal error. (PSAVE,PRSTR,PCOPY)

9: BREAK SENSED

The break flag is set. (PSAVE,PRSTR,PCOPY)

10: UNRECOVERABLE VERIFY

The destination media failed to verify. (PSAVE,PRSTR)

11: SPARE ATTEMPT FAILED.

A disc that supports sparing could not be spared. (PRSTR,PCOPY, VE option only)

12: TAPE WRITE ERROR

An exec write request to the tape LU has failed. (PSAVE)

13: UNRECOGNIZED TAPE FORMAT

The tape is not in any supported format. (PRSTR)

15: NO MATCHING SUBCHANNEL

The destination LU of a unit restore has fewer subchannels than the source unit. Can be seen off-line for SE restores. This message is seen if you try to do an on-line restore of a SAVE from-to save tape. (PRSTR)

16: DEST TRACK MISMATCH

The destination LU has a differnt number of tracks than the source LU. The decision has been made to abort the operation after being prompted to continue. (PRSTR,PCOPY)

17: DEST SECT/TRACK MISMATCH

The destination LU has a different number of sectors per track than the source LU. (PRSTR,PCOPY)

18: TAPE READ ERROR - HARDWARE

An exec read request from a tape failed due to hardware problems. Tape read checksum errors do not abort the operation, but are reported. (PRSTR)

19: TAPE CHECKSUM BLOCKS BAD

Both checksum blocks on a cartridge restore contain bad data. This is a fatal error. (PRSTR)

20: DISC IS WRITE PROTECTED

The disc is write protected. If subsequent operations exist (i.e., an SE restore) you may elect to continue. (PRSTR,PCOPY)

21: FORMAT SWITCH IS OFF

The format switch for the destination disc LU is not on. If subsequent operations exist, you may elect to continue the operation. (PRSTR,PCOPY)

22: DISC LU LOCK FAILURE

An LURQ call to lock a disc LU failed, or a lock failed and you elect to terminate the operation. The discs are not locked off-line, so this check is not made when the utility is running off-line. (PRSTR,PCOPY)

If you are in session mode on-line, dismount any cartridges to be restored or copied to, but do not release them. If they are released, that disc LU is removed from your SST (Session Switch Table) and an LU that is not in your SST cannot be locked, resulting in this error.

23: DISC IS MOUNTED

A disc LU is mounted, and you elected to terminate any remaining operations. This check is not made when running the utilities off-line. (PRSTR,PCOPY)

24: BAD LU WAS ACCESSED

An internal error. If this error is encountered, contact your HP service representative for assistance. (PRSTR,PCOPY)

25: UNEXPECTED EXEC CALL FAILURE

An internal error in the same category as error 24. Off-line, however, it could indicate that incorrect track map entries were defined. This would likely be due to defining disc attributes in a way that contradicts the hardware characteristics, such as too many cylinders or too many sectors per track. (PRSTR,PCOPY)

26: UNEXPECTED END OF DATA

An end of data was encountered with some tape reads still expected. This should only occur if the tape save using PSAVE is halted by setting the break flag. (PRSTR)

27: UNEXPECTED END OF FILE

An end of file was encountered with some tape reads still expected. If this error occurs, contact your HP service representative. (PRSTR)

28: SE RESTORE, NOT PSAVE TAPE

An SE restore was selected, but the saved tape is not in PSAVE format. The SE option is not supported in this case. (PRSTR)

29: UNIT RESTORE, NOT UNIT SAVE

A UN restore was selected, but the saved tape is not a UN save. (PRSTR)

30: INVALID TAPE NUMBER MOUNT

An incorrect tape was mounted for a restore operation. Initially, this must be tape number 1, and subsequent tapes must be sequentially ordered. (PRSTR)

31: INVALID MU SAVE

An MU save was requested from discs of differing classes.

32: PB RESTORE, NOT PB SAVE

A PB restore of a non PB saved tape was requested. (PRSTR)

33: INVALID TRANSMISSION LOG

The transmission log on a tape read was not of the expected value. (PRSTR)

34: NOT ENOUGH SUBCHANNELS

(Off-line only) Insufficient track-addressing subchannel definitions were specified. The original operation cannot be completed; however, you can continue and the utility ignores any subchannel numbers larger than the maximum number defined. (PSAVE,PRSTR)

35: INVALID TR COMMAND

(Off-line only) An illegal TR command was issued. The command is ignored and the utility continues using the previous input device. (PSAVE,PRSTR,PCOPY)

36: UNRECOGNIZED MODEL NUMBER

(Off-line only) When specifying the track map, an unidentifiable model number was specified. The command is ignored. (PSAVE,PRSTR,PCOPY)

37: DISC MODEL TYPE CHANGED

(Off-line only) A disc model number was specified that is different from the source or destination LU. The command is ignored. (PSAVE,PRSTR,PCOPY)

39: PASCAL RANGE ERROR

(Off-line only) A specified track parameter is larger than its field in the track-map structure. This is a fatal error. (PSAVE,PRSTR,PCOPY)

40: SON PROGRAM NOT SCHEDULED PROPERLY

(Off-line only) The off-line utilities check for proper scheduling from the father program, to ensure that the common routines are correctly initialized and to prevent problems with sharing resources by more than one utility. Note that the off-line utilities may not be run using the FMGR RU command. (PSAVE,PRSTR,PCOPY)

41: UNEXPECTED ERROR

This error should not be seen in the debugged programs. However, if it should occur, contact your HP service representative for assistance. (PRSTR,PCOPY)

42: TAPE READ ERROR-CHECKSUM

The tape information was invalid. This error does not cause program termination. (PRSTR)

43: XXXXX SCHEDULE ERROR

The son program (for example, PSPAR) could not be scheduled. Program probably is not loaded. (PSAVE,PRSTR,PCOPY)

44: INVALID PB DEVICE(S)

A pushbutton operation was requested for non-CS80 devices, or for CS80 devices not on an integrated unit. (PSAVE,PRSTR)

45: TAPE NOT INITIALIZED

A save to a CS80 cartridge was requested to an uninitialized tape medium. FORMC should be used to initialize the tape, and the utility rerun. (PSAVE,PRSTR)

46: SON PROGRAM TERMINATED ABNORMALLY

A son program scheduled by a backup utility (for example, PSPAR or FORMC) was terminated, most likely using the FMGR OF command. (PSAVE,PRSTR,PCOPY)

47: CANNOT DEFAULT DICS LU

(Off-line only) The disc LU has been defaulted in an off-line system and the utility cannot find an LU of the appropriate type. (RE)

48: FROM-TO SAVE INVALID HERE

An attempt was made to restore a from-to SAVE tape on-line, or to specify an LU/SE/PB option. (PRSTR)

49: CAPABILITY <64

(On-line only) A user must have a capability greater than or equal to 60 in order to do a PushButton restore on-line to a disc that contains LU 2 or LU 3. (PRSTR)

Chapter 7
Logical Interchange Format Utility

Introduction

The LIF utility program allows you to interchange files between an HP 1000 system and other HP Computer Systems. It translates files to and from the RTE FMP format and a standard Logical Interchange Format (LIF), and provides a means for reading and writing files on various media (such as CS80 cartridge tapes and flexible discs) using the common LIF format. The utility can function with any supported HP disc using standard HP disc drivers as well as any CS80 disc-compatible cartridge tapes.

The utility executes commands similiar to FMGR commands, with similiar results. It can be run interactively or from a transfer file. The output can be stored to a legal RTE namr that can be another file or LU. This utility is not intended to duplicate the file management capabilities of the File Manager Program (FMP). It is intended only for manipulating files on LIF media and interfacing with the FMP.

LIF conforms completely to the HP standard Logical Interchange Format. Although the file descriptor (file name and subparameters) used in the standard LIF format differs from that for RTE, legality checking can be partially disabled to allow non-LIF file names to be written. This may be useful to preserve naming conventions if both the source and destination systems are capable of handling both conventions.

When reading files from other systems, checks are made to verify the legality of the format. The utility reports significant differences from the format but can be forced to continue reading the media if possible (refer to the SV command description for details). LIF conforms to the proper format when writing and, whenever possible, reading data not conforming to the standard format.

Utility Commands

The following LIF utility commands are provided:

CO Copy files from one medium to another

DL Directory list

EX Exit

HE Help

IN Initialize the LIF medium

LI List file on the list device

LL Logical list, specifying the list device

MC Mount cartridge

PK Pack cartridge (LIF medium)

PU Purge file from LIF medium

ST Store files to and from LIF medium

SV Severity level

TR Transfer to a LIF command file

The LIF files are all type 1 (Interchange type), therefore all FMP files are copied into type 1 LIF files. Conversely, all files copied from a LIF medium are copied into type 3 FMP files unless specified otherwise in the type subparameter of the namr. It is not required that type 1 files be ASCII; the type indicates that it is an interchange file and thus has the same record format as other interchange files.

Naming Conventions

Not all legal RTE file names are legal LIF file names because LIF file names consist of only capital letters and the digits 0-9. When the name of the LIF file is not specified, the name of the input FMP file is used. If that name is not a legal LIF name, the legality checking must be suppressed or an error is reported and the file not written to the LIF medium.

The LIF file names may be up to 10 characters long, thus several files with different FMP file names may map into the same LIF file name. If not specified, FMP files are given the name of the LIF file they were copied from, but the name is truncated to six characters. If there is a naming conflict it is treated as a duplicate name error and is not copied.

There are three different types of namr as used in the command parameter description given below. The first type is the FMP namr and is precisely the same as the namr used for file accesses.

The second type is the LIF namr, which is similiar to the first but with a few significant differences: the file name may be up to 10 characters long; the name must begin with a capital letter; the remaining characters must be either capital letters or digits. The LIF utility will accept any legal FMP namr, including 10 characters, as an LIF namr but it may fail to be a legal LIF name. If this is the case, error checking must be disabled. It is possible to write illegally named files (refer to the SV command description).

The third type is referred to simply as a namr; it is not specifically designated as an LIF or an FMP namr. The namr can be either an FMP namr or an LIF namr, whichever the context implies, and is determined by the program. Any ambiguity is resolved by the convention that the mounted LIF medium (there is at most one at any given time) is considered to be at the top of the cartridge list. Therefore, files that are not clearly specified as either LIF or FMP are first searched for (or created) on the LIF medium. If this fails to satisfy the search, the FMP cartridges are searched, using standard FMP calls.

For all namr types, the file descriptor CRN is defined. If that subparameter is a negative number, LIF checks to see if it matches the mounted LIF medium. If a match is found, it is treated as if the reference is to that medium. If a match is not found, it is treated as an RTE CRN (which may be either a negative LU or a positive CRN), and is handled as a normal FMP CRN.

Calling LIF

The LIF utility can be run either interactively or non-interactively. The entry and program prompt sequence for interactive operation is

```
:[RU,]LIF
 LIF:
```

You can now enter the commands for the desired operation. To run LIF without any operator intervention, you must create a command file containing all desired LIF program commands, and enter the command file namr in the runstring as

```
:RU,LIF[,command FMP namr[,list FMP namr]]
```

A list LU may be specified to report error messages or file/device listings. Both the command and list namr must be FMP files since LIF will not run from an LIF file. The default for both files is to your terminal. If the commands are from a file or a non-interactive device, the prompt is supressed and the commands are echoed to the list file or device.

The commands executed by the LIF utility are a subset of the commands executed by the FMGR program, with similiar results. In general, the commands execute the same in LIF as they do in the FMGR. In some cases, the LIF commands do not provide the full range of optional parameters.

HE — Help

The help command provides an on-line reference to all LIF commands and their formats. Call the help command as

```
LIF: HE (or ?, or ??)
```

This command prints a listing of all of the available commands with their parameters, with optional parameters shown within brackets, as

```
LIF: HE

CO,-lif lu,-rte lu/crn
DL,[mask],[level]
EX
HE
IN,lu,[volume label],[directory start],[directory length],[tracks]
LI,namr
LL,[rte namr]
MC,lu
PK
PU, lif file name
RN, lif file name, lif file name
ST,namr,[namr]
SV,[severity level]
TR,[rte namr]
REV. 2140  <811222.1654>
```

CO — Copy

The copy command copies all of the files on a LIF medium onto an RTE disc cartridge. The command syntax is:

```
CO,-LIF lu,-RTE lu (or CRN)
```

As each file is copied, the name is output to the list file or device. If there are any errors, the error message appears after the name of the file to which it applies. All files copied will be type 3 FMP files. There is no parameter to specify renaming or copying a subset of the files. Copying an FMP disc to an LIF medium is not allowed since, in general, the files on an FMP disc are not suitable for an LIF medium. Most FMP files will have illegal names (&,%,',...) and are not of a readily interchangeable type.

LIF file names are truncated to six characters in the FMP file names. If this results in naming conflicts, subsequent conflicting files are reported as FMP ERROR -2 (duplicate name). The files must be renamed or stored under a different name using the ST command.

DL — Directory List

The directory list command identifies the files on the mounted LIF medium and provides selected directory information about the files. The command syntax is:

```
DL[,mask[,level]]
```

The optional mask parameter allows you to specify wildcard characters that can match any letter. Thus if you enter DL,B----, LIF lists the names of all the files with five or less letters which start with a "B". The mask can be up to 10 characters long since the maximum length of an LIF file namr is 10 characters.

The second optional parameter specifies the level of directory information to be listed. The default is level 0 which lists the names of the files in the directory alphabetically, five names per row. The utility uses the memory space behind the program to store the names prior to sorting them, and there may not be enough space to hold all of the names. In this case the level defaults to level 1, which lists just the names of the files, one name per line, in the order in which they appear in the directory. You may also legally specify level 1.

Specifying level 2 provides a list that includes the names in the order they appear on the directory, one name per line (only level 0 alphabetizes the names) with their type and size in sectors listed on the same line.

Level 3 includes all of the information available in the directory about the file. The type and size, the starting track and sector, the date and time stamp (creation time of the file), and the volume number all are listed. The volume number indicates which volume of a multi-volume file this medium holds. One file may span several media, and the volume number indicates which section of the file it is. If this is the last volume of the file, an asterisk is placed before the volume number. (This is normally not used becuse the LIF utility does not support multi-volume files.) Specifying a level greater than 3 gives the same information as level 3 plus a list of purged files (identified as type 0 in the listing).

For example:

```
LIF: DL,,4

 NAME        TYPE  # SCTS  TRAK/SEC   DATE/TIME STAMP   VOL NUM

XYZZX12345  00001 000017  0037 013  13:38:42 10/14/81  *000001
```

IN — Initialize

The initialize command writes a volume label and blank directory on a LIF medium. The command syntax is:

```
LIF: IN,LU[,vol label[,direct. start[,direct. length[,tracks]]]]
```

This command can be used to prepare a blank medium for use with the LIF utility. It clears the medium of all the files and writes a volume label that includes the name of the volume (volume label), the sector address of the start of the directory, the number of sectors reserved for the directory, and the number of tracks on the medium. It also writes an end-of-directory mark over the first file on the medium and other information not specified by the user. When the command is executed it checks that the LU is an appropriate LIF medium, that it is up and available, and that it is not already mounted to the system as an FMP cartridge. The prompt

```
DO YOU WANT TO CLEAR LU XX
```

is then issued. A Yes answer causes LIF to initialize the medium. Any other response displays the message and prompt

```
NOT INITIALIZED
LIF:
```

Any parameters not specified are set to the default values:

```
Volume label      - DEFALT
Directory start   - logical sector 2
Directory length  - 14 sectors
Number of tracks  - determined from system tables
```

After the information is written, the medium is mounted and made available to LIF. Thus it is not necessary to mount an LIF medium before initializing it.

LI — List

The list command copies files, either FMP or LIF, to the list file or device. The command syntax is:

```
LIF: LI,namr
```

This command searches in the LIF directory first unless a CRN is specified in the namr. If the file is not found in the LIF directory, the RTE directory is searched for the file. If it is not found, an error is reported. If it is found, the file is listed (without line numbers) to the current list file or device.

LL — Logical List

The logical list (LL) command changes the list file or device. The command syntax is:

```
LIF: LL[,FMP namr]
```

The LL command can be used to specify the file or device to which listings and error messages will be sent. If a specified file already exists, it is opened and information is appended to it. If the file does not exist it is created (default type 3, size 24, on the first available FMP LU). Outputs to the list device are generated by the DL command (directory listings), the LI command, and the help and other routines (error messages, etc.) The default list device is the scheduling LU, normally your terminal.

MC — Mount Cartridge

The mount cartridge (MC) command mounts an LIF cartridge, specifying that it is the cartridge to be referenced by subsequent commands. The command syntax is:

```
LIF: MC,LU
```

The MC command does not mount an FMP cartridge; it checks that the LU being accessed is an appropriate LIF medium, that it is up and available, and that it is not mounted to FMP. Then it reads the volume label from the cartridge and saves the information that specifies the cartridge size and type, the number of directory tracks, etc. At this point the cartridge is checked for conformity to the LIF standard if the severity level is specified as positive. (Refer to the SV section for a description of positive/negative severity.) Any errors in format are reported.

The mounted LIF cartridge is used by all subsequent commands that implicitly refer to the LIF cartridge. Only one LIF cartridge can be mounted at a time; if a cartridge is already mounted, it is dismounted when another is mounted.

PK — Pack Cartridge

The pack cartridge (PK) command packs all files on the LIF cartridge to recover free space. The command syntax is:

```
LIF: PK
```

The PK command moves the files next to each other, eliminating the space previously occupied by purged files. It also consolidates the directory, removing all entries for the purged files. Before a PK operation, the purged files are still listed in the directory as type 0 files (shown by a DL,,4). The PK command removes these entries by sliding all subsequent directory entries down. The end of directory mark is also moved down to the last active file.

PU — Purge

The purge (PU) command removes a file from the LIF medium. The command syntax is:

```
LIF: PU,LIF namr
```

The PU command changes the type of a file to be purged to 0, which is defined by the LIF standard to be a purged file. This is the only action the PU command takes. A purged file can not be listed, only appears on a level 4 or higher DL and is physically written over when the medium is packed.

RN — Rename

The rename (RN) command changes the name of an LIF file. The command syntax is:

```
LIF: RN,old LIF namr,new LIF namr
```

The RN command rewrites the directory entry for the file with the first name, inserting the second name. No other information except the name has been changed. Before the name is changed, the new name is checked to make sure it is a legal LIF name, unless the SV level is negative. (Refer to the SV command description for details.)

ST — Store

The STore command is the main command used to create files from existing files on FMP/LIF media to FMP/LIF media. The command syntax is:

```
LIF: ST,source namr[,destination namr]
```

This command is similar to the File Manager ST command. LU numbers can be used as well as any FMP file. If a CRN is specified, it is searched for the file. If no CRN is specified for the source namr, the LIF medium is searched and, if the file is not found, the FMP discs are searched. If the file is still not found, the error messages

```
LIF ERROR -06
FMP ERROR -06
```

are displayed (refer to the Error Messages section for a description of the errors). If no CRN is specified for the destination namr, the file is placed on the LIF medium (since it is at the top of the psuedo cartridge list). If the destination file already exists the store is not executed and an error is reported (LIF or FMP ERROR -02).

If the second parameter is defaulted, the source is searched as before. If the source file is found, the destination file will have the same name but be placed on the medium of the other format. Thus if the source file was found on an FMP disc, that file will be stored on the LIF medium. If the source file was found on the LIF medium, it will be copied to the first FMP disc.

It is useful to note that the above specification does not imply that you can store only from LIF to FMP and back. You can also store FMP files (or LUs) to other FMP files (or LUs), and LIF files to other LIF files. All four combinations are possible with this command:

FMP to FMP, FMP to LIF, LIF to FMP, LIF to LIF.

If an FMP namr is specified, the full namr is used, either as a source or a destination. Thus you can individually specify the type (default is 3), the size (default is twice the number, of sectors occupied on the LIF medium), the CRN (default is the top of the cartridge list), and the security code (default is 0).

The ST command does not allow the specialized control, format and file-skipping functions to be specified as in the FMGR ST command. Therefore it is normally not used for storing FMP files to different types of FMP files. For instance, storing a binary relocatable from cartridge tape into a type 5 file does not create a loadable type 5 file. The same naming restrictions apply as in the CO command. This is not a dump command; it will not copy over an existing file. If the destination file already exists, an error will be indicated.

SV — Severity

The severity (SV) command modifies the amount of error checking done by the utility to determine conformity with the HP Logical Mass Memory Format Standards. Two levels of checks can be specified. Level 0, the default level, checks for conformity with Revision 0 of the standard. SV level 1 checks for conformity with Revision 1 of the standard. Note that media in conformance with level 0 also conform with level 1 requirements; however, media that conform with level 1 may not conform with level 0 requirements since level 1 (LIF Standard Revision 1) contains features not found in level 0. If error-checking finds the media not in conformance, an appropriate error message is issued and the related command is not executed.

You can override the error check by specifying a negative integer as the severity level. A negative SV turns off all error checking, thus allowing non-conforming file names and unusual configurations. The override also can be used for storing FMP files with names that are illegal to LIF. If the destination system is also capable of reading non-legal names, it may be easier to maintain the illegal but descriptive FMP names. The utility can read any medium it writes, however other systems may not be able to read files written with a negative severity level.

The SV command takes the form

```
LIF:SV[-[level]]
```

If the SV command is issued without specifying a severity level, LIF returns the current severity level, as

```
LIF: SV
 SV LEVEL IS xxxxxx
```

where xxxxxx is the positive or negative severity level. If you encounter difficulties in reading media from another system, setting the severity level to -1 (LIF: SV,-1) may solve the problem.

TR — Transfer Control

The TRansfer command enables the transfer of control from one RTE file or LU to another. It allows you to specify the source of commands the LIF utility will execute. There is no ":" before each line. Note that there is no command for calculations (as in the FMGR TR command) or to transfer back, except to do an explicit TR command to return to the original file. Thus there is no command file stacking. The command syntax is:

```
LIF: TR[,FMP namr]
```

The FMP namr is the command file or LU where the LIF commands are to be entered. The LIF utility maintains an interactive device flag such that, if commands are not coming from an interactive device, the prompts are suppressed. The default command file is the scheduling LU.

Exit Command

The exit commands terminate LIF, closing open files and performing other proper termination functions. Any of the following commands also terminate LIF:

```
LIF: EX or EN or /E or ex or en or /e
```

Errors

When an error is detected, a message is written to the list device indicating the type of error. All of the LIF errors are matched as closely as possible to an error number from the FMGR error codes. Any errors printed to the list device are also posted in the 2P global. A FMGR transfer file then can call the LIF utility and determine if any errors occurred during execution. (The 2P-5P globals are all cleared when the program begins executing.) Thus, you can call HE in the FMGR to get a brief description of the error. The error format is:

```
   LIF ERROR [-]XXX
or
   FMP ERROR [-]XXX
```

FMP errors are the standard errors described in the RTE-6/VM Terminal User's Reference Manual. The meaning of each error should be interpreted according to the actions the utility was taking. As an example, LIF error -105 corresponds to FMGR error -105, and means that the program did not have enough room in memory for sufficient buffer space to execute the command. The FMP error codes are summarized below; refer to the Terminal User's Reference Manual for a detailed description of each error code.

FMGR ERROR CODES

ERROR	MEANING
-101	ILLEGAL PARAMETER IN D.RTR CALL
-102	ILLEGAL D.RTR CALL SEQUENCE
-105	D.RTR DIRECTORY BUFFER TOO SMALL
-99	DIRECTORY MANAGER EXEC REQUEST ABORTED
-52	SPOOL SHUT DOWN. SPOOL FILE SETUP FAILED.
-49	COPY VERIFY FAILED
-48	SPOOL NOT INITIALIZED OR SMP CANNOT BE SCHEDULED
-47	NO AVAILABLE SESSION LU FOR SPOOL FILE
-46	GREATER THAN 255 EXTENTS
-41	NO ROOM IN SST
-40	LU NOT FOUND IN SST
-39	SPOOL LU NOT MAPPED TO THE SPOOL DRIVER
-38	ILLEGAL SCRATCH FILE NUMBER
-37	ATTEMPT TO PURGE AN ACTIVE TYPE 6 FILE
-36	LOCK ERROR ON DEVICE
-35	MORE THAN 63 DISCS
-34	DISC ALREADY MOUNTED
-33	NOT ENOUGH ROOM ON DISC CARTRIDGE
-32	DISC CARTRIDGE NOT FOUND
-30	VALUE TOO LARGE FOR PARAMETER
-26	QUEUE FULL OR MAX PENDING SPOOLS EXCEEDED
-25	NO SPLCON ROOM
-24	NO MORE BATCH SWITCHES
-23	NO AVAILABLE SPOOL FILES
-22	NO AVAILABLE SPOOL LU'S
-21	ILLEGAL DESTINATION LU
-20	ILLEGAL ACCESS LU
-19	ILLEGAL ACCESS TO SYSTEM DISC
-18	ILLEGAL LU
-17	ILLEGAL READ/WRITE ON TYPE 0 FILE
-16	ILLEGAL TYPE OR SIZE=0
-15	ILLEGAL NAME
-14	DIRECTORY FULL
-13	DISC LOCKED
-12	EOF OR SOF ERROR

FMGR ERROR CODES (Cont.)

ERROR	MEANING
-11	DCB NOT OPEN
-10	NOT ENOUGH PARAMETERS
-09	ATTEMPT TO USE APOSN OR FORCE TO 1 A TYPE 0 FILE
-08	FILE OPEN OR LOCK REJECTED
-07	BAD FILE SECURITY CODE OR ILLEGAL WRITE ON LU 2 OR 3
-06	FILE NOT FOUND.
-05	RECORD LENGTH ILLEGAL
-04	RECORD SIZE OF TYPE 2 FILE IS 0 OR UNDEFINED
-03	BACKSPACE ILLEGAL
-02	DUPLICATE FILE NAME
-01	DISC ERROR
000	BREAK
001	DISC ERROR-LU REPORTED
002	INITIALIZE LU 2!
003	INITIALIZE LU 3!
004	ILLEGAL RESPONSE TO 002 OR 003
005	REQUIRED TRACK NOT AVAILABLE - RELATIVE TAT POSITION REPORTED
006	FMGR SUSPENDED
007	CHECKSUM ERROR
008	D.RTR NOT LOADED
009	ID-SEGMENT NOT FOUND
010	INPUT ERROR
011	DO OF,XXXXX,8 ON NAMED PROGRAMS
012	DUPLICATE DISC LABEL OR LU
013	TR STACK OVERFLOW
014	REQUIRED ID-SEGMENT OR ID-EXTENSION NOT FOUND
015	LS TRACK REPORT
016	INSUFFICIENT SYSTEM TRACKS FOR RP
017	ID SEGMENT NOT SET UP BY RP
018	PROGRAM NOT DORMANT
019	FILE NOT SET UP BY SP ON CURRENT SYSTEM
020	ILLEGAL TYPE 0 LU
021	ILLEGAL DISC SPECIFIED
022	COPY TERMINATED
023	DUPLICATE PROGRAM NAME.
038	ATTEMPT TO REMOVE ACTIVE TYPE 6 FILE
041	PROGRAM CANNOT BE A SEGMENT
042	LU CANNOT BE SWITCHED
043	LU NOT FOUND IN SST
044	NO MESSAGES WAITING
045	SESSION COMMAND ONLY
046	INSUFFICIENT CAPABILITY
047	SPOOL SETUP FAILED
048	GLOBAL SET OUT OF RANGE
049	CAN'T RUN RP'ED PROG. OR PARTITION TOO SMALL

FMGR ERROR CODES (Cont.)

ERROR	MEANING
050	NOT ENOUGH PARAMETERS
051	ILLEGAL MASTER SECURITY CODE
052	ILLEGAL LU.
053	ILLEGAL LABEL OR ILABEL
054	DISC NOT MOUNTED
055	MISSING PARAMETER
056	BAD PARAMETER
057	BAD TRACK NOT IN FILE AREA
058	LG AREA EMPTY OR TOO SMALL!
059	REPORTED TRACK UNAVAILABLE
060	DO YOU REALLY WANT TO PURGE THIS DISC? (YES OR NO).
061	DO A "DC" AND A "MC" ON THIS CR
062	MORE THEN 63 DISCS
063	EXCEEDING SESSION DISC LIMIT
064	NO DISCS AVAILABLE FROM DISC POOL
065	CONFLICT IN SST DEFINITION
066	NO ROOM IN SST
067	PROGRAM NOT FOUND
068	LU NOT IN VARIABLE PART OF SST
069	JOB LOGON FAILED
070	SECTORS/TRACK VALUE TOO LARGE
071	DO "EX,SP" TO SAVE OR "EX,RP" TO RELEASE PRIVATE CARTRIDGES
072	LU NOT INTERACTIVE
073	ACCOUNT NOT FOUND
074	JO COMMAND EXPECTED
075	CAN'T RESTORE TYPE 6 PGM (USER PROTECTED)
076	CAN'T RESTORE TYPE 6 PGM (GROUP PROTECTED)
077	CAN'T RESTORE TYPE 6 PGM (INSUFFICIENT CAPABILITY)
078	CAN'T RESTORE TYPE 6 PGM (INTERNAL ERROR)
079	WARNING - RECORDS TRUNCATED TO 128 WORDS
080	CANNOT FIND NAMED EMA

Chapter 8
Disc Formatting Utilities

Introduction

The on-line disc-formatting utilities, FORMT and FORMC, allow you to format discs and to verify the disc integrity. FORMT formats MAC/ICD discs and flexible discs; FORMC performs the same functions for the CS80 sealed discs and fixed-block tape cartridges.

FORMT

The utility FORMT allows you to format a flexible disc, initialize and reformat a hard disc, and spare hard disc tracks on-line. A verify function allows you to confirm the data integrity of both the flexible discs and the hard discs. FORMT can be used with all ICD (Integrated Controller Disc) and MAC (Multiple Access Controller) discs, except the HP 7900. Only the HP 9895 double-sided, dual-density flexible disc can be formatted using FORMT.

All hard discs are preformatted (track and sector addresses and timing surface are written to the disc) before shipment from HP. However, you must format a flexible disc before initial use. The FORMT FO function is provided for this operation and can only be used with the HP 9895 flexible discs.

The preformatted hard discs can be initialized, spared, or reformatted with FORMT. Initialization (IN function) is the process of removing the protected status of all tracks in the disc, cleaning up the spare-track pool, and sparing any bad tracks found during the initialization process. Bad tracks reported during subsequent use of the disc can be spared individually using the SP function. Reformatting, in which the status bits are cleared and zeros are written into the data blocks (thus wiping the disc clean), is performed using the RE function. A detailed description of the initializing, formatting, and sparing processes is contained in Appendix A.

Loading FORMT

NOTE

FORMT locks itself into a memory partition. Therefore, the system should be generated to allow background partitions with memory lock.

Load FORMT from the relocatable file %FORMT (92067-16554) during generation, or on-line using LOADR. FORMT also uses $DSCLB, the Disc Utility Library (92067-12002), during the load procedure. In specifying the load, you must override the program page requirements to allow a track buffer for the disc transfer. FORMT itself requires 10 pages; the override program size for each disc type is

```
9895     - 11 pages
7905(H)
7906(H)
7920(H)  - 16 pages
7925(H)  - 18 pages
```

Load FORMT on line with the command sequence

```
:RU,LOADR
 /LOADR: SZ,18          ! override program size (7925[H] disc)
 /LOADR: RE,%FORMT      ! load FORMT utility
 /LOADR: SE,$DSCLB      ! search  Disc Utility Library
 /LOADR: END
:
```

FORMT can be loaded as a type 1, 2, 3, or 4 program with no SSGA required (type 1=fixed, 128-word records; type 2=fixed, user-defined records; type 3=variable records, any data type; type 4=ASCII source code). After FORMT is loaded, it should be saved as a type 6 file (executable memory-image code) using the FMGR SP command and restricting access to the System Manager only.

Calling FORMT

CAUTION

No other program should access the disc LU being used by FORMT. The INitialize, SPare, VErify and REformat tasks lock all disc drives with the same EQT as the MAC and ICD disc LU being accessed, thus system performance is downgraded while these tasks are executing.

When running any FORMT function under Session Monitor, the disc LU to be processed must be defined in your Session Switch Table (SST). Use the FMGR SL command if necessary to enter the LU in your SST. Dismount the LU, using the FMGR DC command, and make sure that there are no other users for this LU. When the disc LU has been processed, mount the disc using the complete FMGR MC command. (Refer to the RTE-6/VM Terminal User's Guide for a description of the FMGR commands.)

Invoke FORMT with the runstring

```
:[RU,]FORMT[,input[,cmd,disc lu[,n]]]
```

where

input is the LU of the device from which the FORMT parameters are to be read. Default is to the log device.

cmd is one of the following FORMT function commands:

FO,disc lu,n - Format a flexible disc using n as the fill-sector value.

IN,disc lu - Initialize the named disc LU.

SP,disc lu,n - Spare track n on the named disc LU.

VE,disc lu - Verify the named disc LU.

RE,disc lu - Reformat the named disc LU.

The disc LU must be specified as a positive integer >0 <64. Note that the system disc subchannel (LU 2) and the auxiliary disc subchannel (LU 3) cannot be initialized. Be aware, however, that these LUs can be reformatted, which would destroy the system disc. Exercise caution in specifying the disc LU when the operation is to REformat a disc.

If you invoke FORMT using only the utility call, or if all parameters are not supplied (only the input LU can be defaulted), FORMT goes into interactive mode, prompts with

```
TASK?
```

and waits for you to enter the desired function. All responses must be entered in upper-case letters; FORMT does not accept a lower-case command entry. If you enter ?? in response to the prompt, FORMT displays the valid function commands as

```
ENTER FO(RMAT),IN(ITIALIZE),SP(ARE),VE(RIFY),RE(FORMAT),EN(D)
```

You can terminate FORMT with EN, /E or EX in response to any prompt. After you have defined the function to be performed, FORMT prompts for the disc as

DISC LU?

Enter the LU of the disc you want to spare, format, initialize, reformat or verify. FORMT will then execute the function. If you specify FORMT interactively, the TASK? prompt is displayed following successful completion of the function and you can either enter another function or exit. If you invoke FORMT with a runstring, the utility exits after completing the function.

Each function has its own prompt set, in addition to the TASK?/DISC LU? prompts. These are defined in the function descriptions.

Format a Flexible Disc (FO)

The flexible disc physical sectors are 128 words each (64-word logical sectors) plus the sector/block preamble and postamble. The postamble holds information needed for data error checking. Formatting the HP 9895 flexible disc includes writing the track and sector addresses and the interleaving fill number in the preamble, and identifying defective tracks.

The interleaving fill number (if supplied in the disc LU parameter) defines the logical order in which consecutive sectors are to be read from the disc. For example, if a fill number = 1 is supplied and sector 1 is read first, the next consecutive physical sector is passed over and the next sector (physical sector 3) is read. The sector is identified in the preamble as sector #2. If the fill number is omitted, the default is to read sectors consecutively (fill number = 0).

If a defective block is found during the formatting process, the status is entered in the preamble. The entire track containing the bad block is skipped and will be invisible to you during use of the flexible disc. The first good track formatted following the skipped track is assigned the next consecutive track number. For example, if tracks 0, 1, and 2 are good tracks, but the next two tracks (3 and 4) are found to be defective and are skipped, the next good track (physical track 5) will be identified as track #3 in the sector preambles of that track.

NOTE

> In the formatting process, all previous information on the disc LU is lost; binary zeroes are written on every track, including the directory track.

Using FO

The FO function can be invoked using the runstring

```
[RU,]FORMT[,input],FO[,disc lu,n]]
```

or can be called interactively as

```
[RU,]FORMT,,FO
```

This interactive form of the call defaults the input device to your terminal. After the command is entered, FORMT checks the System Track Map table to verify that the LU specified is a flexible disc. Under Session Monitor, FORMT also checks to verify that the LU is in your SST. If there is a discrepancy, the prompt

```
INVALID DISC LU
ENTER DISC LU < 64
```

is issued and FORMT waits for you to enter a valid disc LU. If the correct LU was entered, FORMT formats the flexible disc. If you are in interactive mode, FORMT prompts with

```
DO YOU REALLY WANT TO FORMAT THIS DISC?
```

before beginning the format, since the process will destroy the current contents of the disc. Enter YES or NO. Only the first two characters of the response are examined, so YE is sufficient for YES.

Once the disc LU has been determined to be valid and the YE response has been given, FORMT scans the DRT (Device Reference Table) and finds all other disc LUs that point to the flexible disc drives on the controller. All of these LUs are then locked for the duration of the FORMT process. Both the left side and right side drive LUs are locked, therefore it is not possible for other programs to access either side even though only one flexible disc LU is to be formatted. FORMT then prompts with

```
# OF FILL SECTORS?
```

Enter the fill number, a value between 0 and 28. A fill value of 1 will give the fastest transfer rate with the existing hardware, and is recommended. Refer to the performance graph in Appendix A for a comparison plot of formatting time to fill value.

FORMT makes five passes over the disc in order to detect any potential defective areas. Since each pass can take several minutes, the following is displayed at the beginning of each pass.

```
FORMAT PASS # nn
```

During each pass, FORMT issues commands to the controller hardware that may take several minutes to execute. During execution of these commands, the controller physically locks the door on the drive so that the disc cannot be removed. If you halt FORMT while the these commands are being executed, the drive access door remains locked until all hardware commands are executed.

After formatting the disc, FORMT reports the number of good tracks on the medium as

```
# OF GOOD TRACKS = nnn
```

In interactive mode, FORMT issues the TASK? prompt. You may enter another task or enter EN, /E, or EX to exit. If FORMT was called through a runstring, the utility exits after the successful FO operation.

Example

In the following example, the FMGR SL command is first used to map flexible disc LU 45 to the system LU currently in the user SST. The FMGR DC command then is used to dismount the LU and return the resource to the system (RR parameter). FORMT is then invoked interactively. Following the successful format operation, the flexible disc is mounted to the user session with the FMGR MC command.

```
:SL,45,45                    ! map disc LU into SST.
:DC,45,RR                    ! dismount disc and make sure no
                             ! one is using LU (RR parameter).

 DISC CRN  45  LU   45 DISMOUNTED FROM SYSTEM

:RU,FORMT

 TASK?
 FO

 DISC LU?
 45

 DO YOU REALLY WANT TO FORMAT THIS DISC?
 YES
```

```
# OF FILL SECTORS?
1

READY DISC - ENTER " ",CR
  <cr>
FORMAT PASS # 01
FORMAT PASS # 02
FORMAT PASS # 03
FORMAT PASS # 04
FORMAT PASS # 05

# OF GOOD TRACKS = 130

TASK?
EN

FORMT FINISHED
:MC,45,P,,LU 45,1,45          ! mount cartridge to get
                              ! directory entry.
```

Initialize an LU (IN)

In initializing a preformatted ICD or MAC hard disc, the existing contents of the disc (including the directory tracks) are overwritten with a pattern of all zeroes.

The spare-track pool is initialized first and the tracks, including the sector preambles and postambles, are set to zero. On verification, bad tracks are flagged by setting the defective track bit (D) in the preamble of the track's sector 0.

The mapped tracks are then initialized. If track has been previously flagged as bad, the track is spared and the defective track/spare-track addresses are cross-referenced in the sector 0 preamble of both tracks. As part of the initialization process, each track is then verified block-by-block, and bad tracks encountered in the verification also are spared.

CAUTION

The IN command locks all disc drives sharing the EQT of the specified disc LU. The lock is issued on a track-by-track basis for each track on the disc. All loads, swaps, and other accesses to any of the discs sharing the same EQT are severely limited during execution of the IN command. If the system disc shares the EQT, the IN command should be used only when severely degraded system performance can be tolerated.

Using IN

The initialization function can be invoked using the runstring

```
[RU,]FORMT[,input],IN,disc lu
```

or can be called interactively as

```
[RU,]formt,,IN
```

The form of the above interactive call defaults the input device to the log terminal. After the command is entered, either interactively or by runstring, FORMT verifies the disc LU and type. If you attempt to initialize LU 2 (system disc), LU 3 (auxiliary disc), or a flexible disc, FORMT issues the message

```
INVALID DISC LU
ENTER DISC LU < 64
```

and waits for you to enter a valid disc LU. When the correct LU is entered, FORMT initializes the disc. In interactive mode, FORMT prompts with

```
DATA WILL BE DESTROYED, OK TO PROCEED?
```

Only the first two letters are significant, so YE is sufficient for YES. If the answer is YES, FORMT initializes the disc LU and reports the initialized state as

```
BAD TRACKS SUBCHANNEL xx

LU XX       LOGICAL CYL HEAD UNIT/ADDR
BAD TRACK    xxxx  xxxx  xx     xx
SPARED TO    xxxx  xxxx  xx     xx

xxxx SPARE TRACKS AVAILABLE
```

In interactive mode, FORMT then issues the TASK? prompt. You may enter another task or enter EN, /E, or EX to exit. If FORMT was called through a runstring, the utility exits after the successful IN operation.

Examples

In the following example, the FMGR command SL is first used to map disc LU 54 to the system LU currently within the user SST. The FMGR command DC is then used to dismount the LU and return the resource to the system (RR parameter). Following this, the FORMT utility is invoked interactively. After the successful initialization operation (track sparing is not required in this example), FORMT displays the number of remaining spare tracks and returns to the TASK? prompt. When FORMT exits, the MC command is then used to mount the initialized disc as a group cartridge.

```
:SL,54,54                       ! map disc LU into SST
:DC,54,RR                       ! dismount disc and make sure
                                ! no one else is using LU 54
                                ! (RR parameter)

 DISC CRN    54  LU  54 DISMOUNTED FROM SYSTEM

:RU,FORMT

 TASK?
 IN

 DISC LU?
 54

 DATA WILL BE DESTROYED, OK TO PROCEED?
 YE

 0004 SPARE TRACKS AVAILABLE

 TASK?
 /E

 FORMT FINISHED
:MC,-54,G,203,CRN54,1,54        ! mount cartridge to get
                                ! directory entry.
```

In the following example, the FMGR SL, DC, and MC commands are used as above; the utility is called using a runstring, so no operator intervention is required. The results of the track sparing are displayed, the number of remaining spare tracks is listed, and FORMT exits.

```
:SL,14,14                          ! map disc LU into SST
:DC,14,RR                          ! dismount LU and make sure LU is
                                   ! not in use (RR parameter)

 DISC CRN    14 LU  14 DISMOUNTED FROM SYSTEM

:RU,FORMT,,IN,14

 BAD TRACKS SUBCHANNEL 05          ! bad tracks found on LU 14,
                                   ! Subchannel 5.
                                   ! results of track sparing are
                                   ! displayed following initialization

 LU 14       LOGICAL CYL HEAD UNIT/ADDR

 BAD TRACK    0002  0210  02     00
 SPARED TO    0198  0406  02     00
 BAD TRACK    0089  0297  02     00
 SPARED TO    0199  0407  02     00

 0003 SPARE TRACKS AVAILABLE

 FORMT FINISHED
:MC,14,P,,LU14,1,14                ! mount cartridge to get
                                   ! directory entry.
```

Spare a Track (SP)

When a bad track is found during a disc read, an error is returned to the calling program, identifying the defective track. The FORMT SP function is used to spare these bad tracks individually. In track sparing, a known good track is substituted for the defective track and as much data as possible is recovered to the spare track. The defective track address and the spare track address are then cross-referenced in the preamble of both tracks' sector 0.

Using SP

CAUTION

Be aware that once a track is spared, it is permanently marked as defective. The process is irreversible.

CAUTION

All discs sharing the EQT of the specified disc LU are locked during the SP operation. All loads, swaps, and other accesses to any of the discs on the EQT are not allowed while a track is being spared.

Invoke SP using the runstring

```
[RU,]FORMT,[,input],SP,disc lu,n
```

where

input	is the device from which inputs are to be received and prompts are to be displayed.
SP	calls the track SPare function.
disc lu	is the LU of the disc containing the bad track to be spared.
n	is the bad track to be spared.

SP also can be called interactively as

```
[RU,]FORMT,,SP
```

The form of the above interactive call defaults the input device to the log device. After the command is entered, either interactively or by runstring, FORMT verifies the disc LU and type. If you attempt to spare a flexible disc or if the disc LU is not in your SST (under Session Monitor), FORMT issues the message

```
INVALID DISC LU
ENTER DISC LU < 64
```

and waits for you to enter a valid disc LU. When the correct LU is entered, FORMT spares the track specified in the runstring. In interactive mode FORMT prompts with

```
TRACK TO BE SPARED?
```

If you enter ?? to this prompt, FORMT returns the track range of the disc. Enter the logical track number to be spared. Note that if you enter the logical number of a good track, it is spared and permanently labeled as a defective track. If the specified track is not within the bounds of the disc LU specified, FORMT prompts with

```
INVALID TRACK #
ENTER BAD TRACK # x - xxxx
```

and waits for you to enter a logical track number within the specified range.

Data is copied block by block from the bad track to the spared track. If all the information on the track cannot be recovered, FORMT issues the warning

```
WARNING: ALL INFORMATION ON TRACK NOT SUCCESSFULLY RECOVERED
```

and continues sparing the defective track. When the sparing is complete, FORMT reports the successful operation as

```
BAD TRACKS SUBCHANNEL xx

LU xx       LOGICAL CYL HEAD UNIT/ADDR

BAD TRACK    xxxx  xxxx  xx     xx
SPARED TO    xxxx  xxxx  xx     xx

xxxx SPARE TRACKS AVAILABLE
```

In the interactive mode, FORMT then issues the TASK? prompt. You may enter another task or enter EN, /E, or EX to exit. If FORMT was called through a runstring, the utility exits.

Examples

In the following example, the FMGR SL is used to map disc LU to the system LU currently in the user SST. The FMGR command MC is then used to mount LU 14. (If the disc is already mapped and mounted, these commands are not necessary, and will cause a FMGR error.) The utility is then called interactively. After the successful sparing operation, FORMT reports the results and returns to the TASK? prompt.

```
:SL,14,14                          ! map LU 14 into SST.
:MC,14,P                           ! mount LU 14 to your session
                                   ! as a private cartridge.
:RU,FORMT

 TASK?
 SP

 DISC LU?
 14

 TRACK TO BE SPARED?
 3                                 ! logical bad track on LU 14

 BAD TRACKS SUBCHANNEL 05          ! LU 14 is on subchannel 05

 LU  14      LOGICAL CYL HEAD UNIT/ADDR

 BAD TRACK    0003  0211  02      00
 SPARED TO    0202  0410  02      00

 0000 SPARE TRACKS AVAILABLE       ! no more spare tracks left.

 TASK?
 /E                                ! exit FORMT.

 FORMT FINISHED
```

In the following example, the FMGR SL and MC commands are used as above. FORMT is called with a runstring, so no operator intervention is required and FORMT exits after sparing the bad track.

```
:SL,23,23                          ! map LU 23 into SST.
:MC,23,P                           ! mount LU 23 as a private
                                   ! cartridge.

:RU,FORMT,,23,SP,1                 ! spare logical track #1 on
                                   ! LU 23.
WARNING! ALL INFORMATION ON TRACK NOT SUCCESSFULLY RECOVERED

BAD TRACKS SUBCHANNEL 03

LU 23       LOGICAL CYL HEAD UNIT/ADDR

BAD TRACK    0001  0410  01      00
SPARED TO    0204  0410  00      00

0002 SPARE TRACKS AVAILABLE

FORMT FINISHED
```

Reformat a Disc (RE)

The FORMT REformat function clears the entire disc area designated by the LU, except for the servo and timing data written to the inter-sector gaps when the disc was preformatted prior to shipment from HP. The directory track is overwritten with zeros, zeros are written to all sector preambles, data areas, and postambles, and the spare track pool is cleared. The reformatted disc is not verified, and bad tracks are not spared during the RE operation.

If the PRSTR VErify option is not specified when restoring the disc, or if the disc is restored using READT, the FORMT IN function should be used before restoration, to initialize the disc and to spare tracks according to the system track map. HP-designated bad tracks must be spared using the FORMT SP function even if the tracks are not found to be bad during initialization. Bad tracks defined during disc preformatting are identified in the HP documentation supplied with the disc. (If the records are not available when the disc is to be reformatted, contact your HP customer engineer for assistance.)

CAUTION

> If the system disc (LU 2) and auxiliary disc (LU 3) are reformatted, the operating system is lost and the system will go down. This capability should be used only with the full and complete understanding of how to back up the disc using PSAVE and how to restore it using the off-line physical back-up utility RE.

Any disc can be formatted, including the system disc and auxiliary disc. However, you must have a user capability of 60 (usually reserved for the system manager or other support personnel) before you you can reformat the system discs, and the reformatting can only be done interactively. (Refer to the RTE-6/VM Terminal User's Reference Manual for a discussion of user capabilities.)

In reformatting discs other than LU 2 or LU 3, the following steps are recommended:

1. Use the FMGR SL command if needed to be sure that the disc is listed in your SST.

2. Use PSAVE or WRITT to back up all data from the disc to be reformatted.

3. Dismount the disc using the FMGR DC command and make sure no one is using the disc.

4. Reformat the disc using RE.

5. If the data is not to be restored, or if the data is to be restored using READT, use the IN function to spare bad tracks.

6. Use the SP function to individually spare those tracks identified as bad during preformatting prior to shipment from HP.

7. Mount the disc using the FMGR MC command.

8. Restore the data, if desired, using PRSTR or READT. If PRSTR is used with the VErify option, the IN function is not needed: the PRSTR VE option spares the tracks in a prepass before the data is written.

In reformatting LU 2 or LU 3, the following steps are recommended:

CAUTION

Do not use WRITT to back up LU 2 or LU 3. This utility only saves the FMGR area of the disc, not the system area.

1. Use PSAVE to back up all data from LU 2 and LU 3. Note that you should back up both LU 2 and LU 3 even though reformatting is required for only one of the system discs. After reformatting, the system will attempt to come back and the results will be unpredictable.

2. Halt all system activity. Failure to halt all activity when reformatting the system discs could cause FORMT to be swapped out onto the disc between reformats, and the utility would be lost.

3. Reformat LU 2 and LU 3. Note that RE functions only in interactive mode when reformatting the system discs.

4. Use the SP function to individually spare those tracks identified as bad during preformatting prior to shipment from HP.

5. Load the off-line physical back-up utilities as described in chapter 6.

6. Restore LU 2 and LU 3 using the off-line RE command with the VErify option to spare bad tracks in a prepass over the disc prior to the restoration.

7. Re-boot the system.

CAUTION

> The RE command locks all disc drives sharing the EQT of the disc to be reformatted. The lock is issued on a track-by-track basis for each track on the specified disc LU. All loads, swaps, and other accesses to any of the discs sharing the same EQT are severely limited during execution of the RE. If the system disc shares the same EQT as the disc being reformatted, the RE function should only be used when severely degraded system performance can be tolerated.

The reformatting function can be invoked using the runstring

```
[RU,]FORMT[,input],RE,disc lu
```

where

input — is the device to which prompts and messages are to be directed. The default is to the log device.

disc lu — is the disc to be reformatted.

RE also can be invoked interactively as

```
[RU,]FORMT,,RE
```

The above form of the command defaults to the log device for parameter inputs. If LU 2 or LU 3 are to be reformatted, RE can only be called interactively. After the command is entered, FORMT verifies the disc LU and type. If you attempt to reformat a flexible disc, or if the disc is not in your SST (under Session Monitor), FORMT issues the message

```
INVALID DISC LU
ENTER DISC LU < 64
```

and waits for you to enter a valid disc LU. If you attempt to reformat LU 2 or LU 3 without a command capability of 60, or without being in interactive mode, FORMT issues the message

```
UNAUTHORIZED LU2,3 ACCESS - (COMMAND IGNORED)
```

and exits or, if in interactive mode, returns to the TASK? prompt. If you are reformatting LU 2 or LU 3, FORMT issues the prompt

```
DO YOU REALLY WANT TO REFORMAT THE SYSTEM DISC?
```

and waits for you to enter a YES or NO response. In interactive mode, FORMT then issues the message

DATA WILL BE DESTROYED. OK TO PROCEED?

and waits for you to enter a YES or NO response. If your response is YES, FORMT proceeds to reformat the disc.

In interactive mode, FORMT issues the TASK? prompt following the successful reformat operation. You may enter another task or enter EN, /E, or EX to exit. If FORMT was called through a runstring, the utility exits after the successful RE operation.

Verify a Disc (VE)

The FORMT verification process, the VE function, reads and verifies all data on either a flexible disc or a hard disc. While the directory track is checked, unused tracks in the spare-track pool are not verified. If a track has been previously spared, VE will verify that spared track. The results of the verification are reported, but track status is not changed. That is, tracks found to be bad are not spared; the SP function must be used to spare bad tracks individually.

CAUTION

> VE locks all disc drives sharing the EQT of the specified disc LU. The lock is issued on a track-by-track basis and all loads, swaps and other accesses to any of the discs sharing the same EQT are severely limited during execution of the VE function. If the system disc shares the same EQT as the disc to be verified, the VE command should only be used when severely degraded system performance can be tolerated.
>
> In verifying a flexible disc, only the LUs associated with a specific disc LU are locked and remain locked until the VE function is completed.

The verify function can be invoked using the runstring

[RU,]FORMT[,input],VE,disc lu

where

input	is the device to which messages are to be sent and, in interactive mode, prompts are displayed.
disc lu	is the flexible disc or hard disc to be verified.

The function also can be called interactively as

```
[RU,]FORMT,,VE
```

The above form of the command defaults the input device to your terminal. After the command is entered, either interactively or by runstring, FORMT checks the disc LU and type. If you attempt to verify a non-disc LU or a disc not in your SST (under Session Monitor), FORMT issues the message

```
INVALID DISC LU
ENTER DISC LU < 64
```

and waits for you to enter a valid disc LU. When the correct LU is entered, FORMT verifies the disc and reports any bad tracks without a valid spare as

```
BAD TRACKS SUBCHANNEL xx
LU xx       LOGICAL    CYL   HEAD   UNIT/ADDR
BAD TRACK    xxxxx     xxxx   xx       xx
```

In interactive mode, FORMT then issues the TASK? prompt. You may enter another task or enter EN, /E, or EX to exit. If FORMT was called through a runstring, the utility exits after the VE operation.

FORMT Utility Error Messages

The following error messages can be generated by FORMT:

```
BAD TRACKS SUBCHANNEL XX

LU  XX      LOGICAL CYL HEAD UNIT/ADDR
BAD TRACK    XXXX  XXXX  XX     XX
SPARED TO    XXXX  XXXX  XX     XX

XXXX SPARE TRACKS AVAILABLE
```

A defective track encountered during an IN or SP task is spared.

```
CYLINDER COMPARE ERROR
LU  XX         LOGICAL CYL HEAD UNIT/ADDR
TARGET TRACK    XXXX  XXXX  XX     XX
```

The disk controller was unable to seek to the target track. Be sure that the disc is formatted. The current task is aborted and FORMT returns to TASK? mode.

INVALID DISC LU

The disc LU is the wrong type, has zero tracks defined in it, or the response was not a numeric parameter. Under session monitor, the disc LU must be defined in your SST (Session Switch Table). LU 2 and LU 3 cannot be initialized.

INVALID DISC SPECIFICATIONS xx

The disc controller detects an out-of-bounds condition on either a cylinder, head, sector, or unit, based on the invalid subchannel definition specified in the message. The current task is aborted and FORMT returns to TASK? mode.

```
LU  XX        LOGICAL CYL HEAD UNIT/ADDR
BAD SPARE      XXXX  XXXX  XX     XX
```

A defective spare is encountered in either the IN or SP task. This message appears in conjunction with the BAD TRACKS heading.

MAX OF 20 BAD TRACKS EXCEEDED

Flexible disc only. FORMT returns to TASK mode.

NOT ENOUGH ROOM FOR TRACK BUFFER

The memory bounds specified for FORMT at load time did not provide sufficient free memory to serve as a track buffer for the task and disc LU requested. FORMT terminates. Reload FORMT and resize it to the recommended program size.

OUT OF SPARE TRACKS FOR THIS LU

All spare tracks have been used. The sparing operation cannot be performed. FORMT returns to TASK? mode.

READY DISC-ENTER " "CR

The disc drive is not ready or there is no disc in the drive. Enter a space and carriage return when the disc is ready, or enter /E to abort the task.

TURN OFF PROTECT OR READ-ONLY SWITCH-ENTER " "CR

The disc protect (or read-only) switch is on. This error also occurs if the floppy disc in the selected drive has the write-protect notch present. Turn off the switch and enter a space and carriage return to continue or enter /E to abort the task.

TURN ON FORMAT SWITCH-ENTER " "CR

The utility is initializing the disc, but the format switch is off. This error also occurs if a write operation is attempted to a track with its P-bit set while the format switch is off. Turn on the format switch and enter a space and carriage return to continue or enter /E to abort the task.

WARNING! ALL INFORMATION ON TRACK NOT SUCCESSFULY RECOVERED

Spare command only. The track about to be spared was not successfully preserved. FORMT continues.

WARNING! POSSIBLE BAD FLOPPY MEDIA -
RETRY FORMT OR DISCARD FLOPPY

Flexible disc only. An error occurred on the fifth and final FORMT/Verify pass. Re-starting the FO process may mask the defective track.

FORMC

FORMC, the on-line maintenance utility for CS80 discs and tapes, allows you to verify the integrity of data on the device using the VErify option. Errors detected can be corrected using the FORMC SPare and FOrmat options to spare bad disc tracks or to format the entire Cartridge Tape Drive (CTD) tape or disc volume.

If only an off-line environment is available, or if physical addressing of the disc is required, the CS80 disc exerciser should be used for disc maintenance. (Refer to the CS80 External Exerciser Manual (5955-3462) for details of the CS80 disc exerciser.)

FORMC Options

Verification (VE option) is a nondestructive process that checks the integrity of a specified disc LU or CTD tape through reading the CRC information (checksum) in each block of the disc or tape. The entire disc/tape, or only specified tracks/blocks, can be verified by FORMC. Note that a block as defined for a CTD tape consists of 512 words, while a CS80 disc block consists of 128 words. All track references are to RTE tracks, not physical disc tracks.

Sparing (SP option) is the process of removing a defective disc block from active service. This process individually spares defective blocks identified by the verify operation. Sparing is not required for CTD tapes because defective blocks are flagged both by the format operation and during a normal read operation when bad tracks are discovered. They are automatically spared on any subsequent write operation, a process known as skip sparing. However, the spare operation can be used to spare out one or more defective blocks within a track of the CTD disc cache memory area. (Refer to the DVM33/DVN33 Driver Reference Manual, Part No. 92084-90025, for details of the CTD disc cache memory scheme.)

Each CS80 disc has a specified number of physical tracks reserved for use as spares when a defective block is detected. Each physical track also contains a spare block. (The spare tracks and blocks are known only to the controller.) When a defective block is spared within a physical track, the track is rearranged to bypass the defective block and to use the spare block. If another defective block is subsequently spared on that physical track, the entire physical track is then bypassed and the data is written to another physical track from the spare-track pool. In either case,the existence of spared tracks/blocks is transparent to the user.

Formatting of a disc (FO option) consists of writing the preamble for each block of data on the medium. In addition, the data areas are initialized. Note that the format command issues the initialize-medium command to the entire disc volume; for most CS/80 discs, this means the entire disc unit. CS/80 discs do not allow the formatting of RTE subchannels, as with previous disc drives. Disc blocks found defective must be specifically spared using the spare command.

Formatting a CTD tape means something different than formatting a disc. If the tape has not been certified (the "initialized media" flag on the tape has not been set), a tape certification process is performed. This includes writing a known pattern on every

block of the tape and then rereading the tape to determine if there are any defective areas on it. In addition, every 512th block on the tape is allocated as a sparing block. A table of these allocated blocks is created on the tape. If any defective blocks have been discovered, they are flagged and subsequent writes to the tape will not use these blocks. This is known as "skip sparing". Finally, the "initialized media" flag is set on the tape. This certification process can take up to 20 minutes for a short tape (DC 150) or more than one hour for a long tape (DC 600). RTE-6/VM utilities that access CTD tapes will not use a tape until the Initialized Media flag is set.

If the tape has been previously certified (the flag is set), you have the option to either recertify the tape or to convert jump spares to skip spares. (In jump sparing, a specific defective block is spared to the first spare block following the bad block.) The recertification process is identical to the initial certification, and any data on the tape will be destroyed in the process. The spare conversion process takes the tape blocks that were spared using jump sparing and flags these blocks to be skipped on the next write to the tape. On the next write, the controller will simply renumber the tape blocks, starting at the next good block following the defective block. The spare conversion option will result in loss of data for those blocks that have been jump-spared.

Previously certified tapes should not have to be recertified on HP 1000 systems unless there are serious doubts as to the quality of the tape. Any defective areas on the tapes discovered after the certification process by either a normal read or verify operation are automatically flagged as defective and automatically skip-spared on any subsequent write operation. Since all HP 1000 utilities use skip sparing, you do not need to use the spare conversion option for HP 1000 CTD tapes. This option is included since other non-HP 1000 systems do use jump sparing and you may want to use this option when reformatting their tapes.

Calling FORMC

The utility can be run with or without user interaction. In interactive mode, FORMC prompts for each command and its associated parameters. For non-interactive operation, all parameters required for the specified command must be included in the runstring. FORMC terminates after successful completion of the command or, in the case of an error condition, after issuing the appropriate error message. FORMC also can be called programatically by issuing the appropriate program-scheduling EXEC call and passing the parameters through the runstring.

Invoke FORMC with the runstring

```
[RU,]FORMC[:IH],[list LU],option,LU[,option params]
```

where

:IH — inhibits automatic renaming of the program. :IH must be specified when FOrmat or SPare is selected since these options require that FORMC be run without renaming.

list LU — is the LU (Logical Unit number) of the list device to which FORMC will direct messages generated during the specified operation.

option — is one of the following commands:

- VE - Verify the integrity of the specified disc or tape.
- SP - Spare the specified disc track. (This requires a user capability of 60 or greater.)
- FO - Format the specified disc or tape. (This requires a user capability of 60 or greater.)

LU — is an LU of the CS80 disc or tape to be accessed by FORMC for the specified operation.

option params — are the parameters applicable to the specified option. Refer to the associated option for a description of the parameters.

Response to System BReak Command

FORMC checks the break flag only during the VErify operation, after each locate and read. If the flag is set, FORMC exits with the messages

```
/FORMC:  BREAK DETECTED
/FORMC:  ABORTED
```

The break flag is not checked during the FOrmat and SPare operations since, to protect the data integrity, these operations should complete without intervention after they are initiated.

Runstring Mode

In runstring mode, all FORMC parameters must be specified in the command string. If a required parameter is omitted, or is entered incorrectly, FORMC exits with the messages

```
/FORMC:  ILLEGAL RUNSTRING PARAMETER
/FORMC:  ABORTED
```

Interactive Mode

When FORMC is called interactively, the following sequence of messages is issued:

```
/FORMC: CS80 DISC FORMAT UTILITY
/FORMC: '?' WILL LIST THE LEGAL COMMANDS
/FORMC: TASK?
```

If you enter ?, FORMC issues the message

```
/FORMC: LEGAL COMMANDS ARE
   VE(RIFY)
   FO(RMAT)
   SP(ARE)
   ?(LIST COMMANDS)
   EN(D)
   EX(IT)
   /E(XIT)
   AB(ORT)
   /A(BORT)
/FORMC: TASK?
```

When the selected option is entered, FORMC issues the applicable series of parameter prompts, as defined in the sections describing each option. Before a command is executed, FORMC issues the message

```
/FORMC: COMMAND EXECUTING - <command>, <option parameters>
```

During execution, FORMC tests the driver status and, if a -1 is present in QSTAT (indicating driver timeout), issues the messages

```
/FORMC: DEVICE DRIVER TIMED OUT. CONTACT SYSTEM MANAGER.
/FORMC: ABORTED
```

and exits. Driver timeout can result if the disc drive is disconnected or if a hardware failure occurs. (Refer to the Error Messages section for a description of the CS80 device driver status information message.)

The exit or abort commands can be entered in response to any FORMC prompt and cause the utility to terminate execution. Do not use the RTE OF command to terminate a spare or format operation as the integrity of the disc data could be jeopardized.

Verify Command

In runstring mode, you must include the VE and media LU parameters in the string; the optional parameters default as defined below. Invoke the VErify option as

```
[RU,]FORMC,[list LU],VE,media LU[,start,numb]
```

where

list LU	is the LU of the hard-copy error-logging device. The default is to no hard-copy log.
VE	calls the verify option.
media LU	is an LU of the CS80 disc or tape to be verified.
start	is the number of the disc track or tape block at which the verify is to begin.
numb	is the decimal number of disc tracks or tape blocks to be verified. If the start and numb parameters are omitted in runstring mode, the entire disc or tape will be verified.

When you invoke the verify option as VE in response to the TASK? prompt, FORMC issues the parameter prompt

```
/FORMC: CS80 DISC OR TAPE LU?
```

Enter the LU of the disc or tape to be verified. If the number given does not correspond to a legal CS80 disc or tape LU, the message

```
/FORMC: ILLEGAL CS80 DISC OR TAPE LU
```

is issued and FORMC reprompts for the number. If a CTD tape is to be verified, FORMC tests to determine if the cartridge to be verified has been inserted into the drive. (The time required to execute this routine is from 10 to 15 seconds.) If the cartridge is not present, the message sequence

```
/FORMC: READY TAPE AND RESTART
/FORMC: FINISHED
```

is issued and FORMC exits after verifying the CTD disc cache memory area associated with the tape (if the cache exists). You must insert the CTD cartridge and restart the utility. If the cartridge has not been initialized (formatted), FORMC issues the messages

```
/FORMC: UNINITIALIZED TAPE MEDIA - FORMAT TAPE
/FORMC: TASK?
```

and returns to the interactive TASK state. You must enter the FO command to format the tape. (Refer to the Format Command section for details.)

When the formatted cartridge is in place and the correct LU has been entered, the following sequence of prompts is issued, as appropriate:

```
/FORMC: VERIFY ENTIRE DISC LU (Y,N)?
```

or

```
/FORMC: VERIFY ENTIRE TAPE (Y,N)?
```

If your response is Yes, FORMC proceeds in verifying the entire disc LU or tape. If you respond No, FORMC prompts,

```
/FORMC: START TRACK NUMBER?
```

or

```
/FORMC: START TAPE BLOCK NUMBER?
```

The number entered must be an integer between zero and the number of tracks or blocks on the medium minus one. In the case of tape verification, the number is between 0 and 65535; if a negative number is entered as the starting block, it is interpreted as positive (-2 = 65534). If an incorrect number is entered (a number larger than the disc/tape capacity or, in the case of disc verification, a negative number) the message

```
/FORMC: ILLEGAL TRACK NUMBER, LEGAL RANGE 0 TO <last track>
```
or
```
/FORMC: ILLEGAL BLOCK NUMBER, LEGAL RANGE 0 TO <last block>
```

is issued, where <last track> or <last block> is the number of the last track/block minus one.

When the correct number is issued, FORMC then prompts with

```
/FORMC: NUMBER OF DISC TRACKS?
```
or
```
/FORMC: NUMBER OF TAPE BLOCKS?
```

One or more tracks or blocks must be specified. If the number given is greater than the capacity of the medium, all tracks or blocks following the start number are verified. Note that a negative number may be specified for the number of tape blocks, as it is interpreted as positive (-5 = 65531). If zero tracks or blocks are specified, or a negative number of disc tracks is specified, the message

```
/FORMC: ILLEGAL NUMBER OF TRACKS
```
or
```
/FORMC: ILLEGAL NUMBER OF BLOCKS
```

is issued and the prompt is repeated. When the correct number is entered, the verification is performed. If the CS80 tape to be verified has an associated disc cache memory, the disc cache also is verified. When a defective track or block is encountered, FORMC issues the message

```
/FORMC: LU <disc LU> - BAD TRACK <number>
```
or
```
/FORMC: LU <tape LU> - BAD BLOCK <number>
```
or
```
/FORMC: LU <disc LU> - BAD TRACK <number> IN CACHE
```

where <number> is a decimal integer identifying the defective track or block.

After the verify is complete, FORMC issues the message

```
   /FORMC: DISC VERIFY COMPLETED <number> BAD TRACK(S).
or
   /FORMC: TAPE VERIFY COMPLETED <number> BAD BLOCK(S).
or
   /FORMC: TAPE CACHE VERIFY COMPLETED <number> BAD TRACK(S).
```

where <number> is either a decimal integer defining the number of defective tracks or blocks or is NO to indicate that no defective tracks or blocks were encountered. Bad tracks can be spared by passing the disc or tape LU and the track number to the SP command.

If an End of File error is encountered during the tape verification operation, FORMC issues the message

```
/FORM: LU <tape LU> - END OF FILE DETECTED IN TAPE BLOCK <number>
```

where <number> is a decimal integer identifying the tape block containing the End of File. FORMC will then resume the verification operation with the next tape block following the End of File.

Format Command (FO)

CAUTION

> The FOrmat command issues the Initialize Media command to the entire tape or disc volume. If a disc LU is specified, the entire disc volume will be formatted; CS80 discs do not allow formatting by subchannels.

NOTE

In order to execute the FOrmat command, FORMC must be scheduled without renaming (as FORMC:IH) and you must have a capability of 60 or greater. (Refer to the RTE-6/VM Terminal User's Reference Manual for a description of the user capability levels.) If the rename-inhibit (:IH) is not included in the call, FORMC aborts with the messages

```
/FORMC: MUST USE THE PROGRAM NAMED `FORMC'
        WITH THE FO AND SP COMMANDS
/FORMC: ABORTED
```

If your assigned user capability is less than 60 and you attempt to format a disc or tape, FORMC aborts with the messages

```
/FORMC: INSUFFICIENT CAPABILITY
/FORMC: ABORTED
```

To execute the format command in runstring mode, you must include the FO and media LU parameters in the string; the optional parameters default as defined below. The rename-inhibit parameter must be included in the program descriptor since the FO option does not allow renaming FORMC. Invoke the FOrmat option as

```
[RU,]FORMC:IH,[list LU],FO,media LU[,interleave]
```

where

list LU	is the LU of the hard-copy error-logging device. The default is no hard-copy log.
FO	calls the format option.
media LU	is an LU of the CS80 disc or tape to be formatted.
interleave	is the interleave factor (a decimal number between 1 and 32) to be assigned. The default is to a factor of 1. (Refer to Appendix A for a discussion of interleaving.)

In non-interactive mode, FORMC aborts with an ILLEGAL RUNSTRING PARAMETER message if you attempt to format a system disc (LU 2 or LU 3), or a disc that is currently mounted. In interactive mode, FORMC issues a warning and user-action prompt sequence before a currently mounted disc or the system discs can be formatted.

When you invoke the format option as FO in response to the interactive TASK? prompt, FORMC issues the parameter prompt

```
/FORMC: CS80 DISC OR TAPE LU?
```

If a tape is to be formatted (also referred to as tape certification or tape media initialization), enter the LU of the CTD drive on which the tape is mounted. Any LU on a disc volume may be entered; the entire volume is formatted by FORMC. If the LU entered does not correspond to a legal CS80 disc or tape drive, the message:

```
/FORMC: ILLEGAL CS80 DISC OR TAPE LU
```

is issued and FORMC reprompts for the LU number.

Tape Formatting

If the FO command is issued to the CTD, FORMC determines if the cartridge has been inserted in the drive; this process requires from 10 to 15 seconds to execute. If the cartridge is not present, the message sequence

```
/FORMC: READY TAPE AND RESTART
/FORMC: FINISHED
```

is issued and FORMC exits. You must insert the cartridge into the CTD tape drive and restart the utility.

When the cartridge is in place, FORMC then determines if the cartridge was previously initialized (certified). If the cartridge shows an UNINITIALIZED MEDIA status, FORMC issues the message sequence

```
/FORMC: TAPE INITIALIZATION(CERTIFICATION) MAY TAKE UP TO AN HOUR
/FORMC: DATA WILL BE DESTROYED ON TAPE LU xx
/FORMC: OK TO PROCEED (Y,N)?
```

If you respond N, FORMC exits. If your response is Y, FORMC proceeds to certify the entire tape, issues the message

```
/FORMC: TAPE FORMATTING COMPLETED
```

and exits.

If the tape is already initialized (certified), FORMC issues the warning and question

```
/FORMC WARNING: TAPE IS ALREADY INITIALIZED(CERTIFIED)
/FORMC: DO YOU WANT TO RECERTIFY OR CONVERT SPARES(CE,SP)?
```

If you respond with CE, FORMC issues the message sequence

```
/FORMC: TAPE INITIALIZATION(CERTIFICATION) MAY TAKE UP TO AN HOUR
/FORMC: DATA WILL BE DESTROYED ON TAPE LU xx
/FORMC: OK TO PROCEED (Y,N)?
```

If you respond with N, FORMC exits. If your response is Y, FORMC proceeds to recertify the tape, issues the message

```
/FORMC: TAPE FORMATTING COMPLETED
```

and exits.

If you respond to the recertification question with SP, FORMC issues the message sequence

```
/FORMC: JUMP SPARES WILL BE CONVERTED TO SKIP SPARES
/FORMC: DATA MAY BE DESTROYED ON TAPE LU xx
/FORMC: OK TO PROCEED (Y,N)?
```

If you respond with N, FORMC exits. If your response is Y, FORMC proceeds to update the tape spare table by converting jump spares to skip spares, issues the message

```
/FORMC: TAPE FORMATTING COMPLETED
```

and exits.

In non-interactive mode, FORMC will always perform a certification on new tapes, or a recertification on previously initialized tapes.

Disc Formatting

After the correct disc LU has been entered, the prompt

```
/FORMC: INTERLEAVE FACTOR FOR OPTIMAL THROUGHPUT (Y,N)?
```

is issued to determine if you want to specify a record interleave factor. (Refer to Appendix A for a discussion of interleaving.) An answer of Yes means that you want FORMC to supply the default optimal throughput factor of 1, which is sufficient for most applications. If you answer No, FORMC expects you to supply the interleave factor, and issues the prompt

```
/FORMC: INTERLEAVE FACTOR?
```

Your response to this question must be a decimal number in the range of 1 to 32. If an incorrect response is entered, the message/prompt set

```
/FORMC: ILLEGAL INTERLEAVE FACTOR, LEGAL RANGE 1 TO 32
/FORMC: INTERLEAVE FACTOR?
```

is issued and FORMC waits for the correct entry.

FORMC then locates all disc LUs that are on the same volume and thus subject to formatting. If any of the disc LUs are still mounted, FORMC issues the message set

```
/FORMC: DISMOUNT LU
   <lu 1>,<lu 2>,.....,<lu n>

/FORMC: DISMOUNT LU'S AND RESTART
/FORMC: FINISHED
```

and exits. The LUs listed in the message must then be dismounted using the FMGR DC command, and FORMC must be restarted. With the LUs dismounted, FORMC locks all LUs to be formatted and issues the message

```
/FORMC: DATA WILL BE DESTROYED ON LU
   <lu 1>,<lu 2>,.....,<lu n>
/FORMC: OK TO PROCEED (Y,N)?
```

A response of Yes causes FORMC to proceed to format the entire disc volume. The message

```
/FORMC: DISC FORMATTING COMPLETED
```

is then issued and FORMC exits. A response of No causes FORMC to unlock all LUs and to exit without formatting any discs.

If a system disc (LU 2 or LU 3) is to be formatted, the warning message and prompt

```
/FORMC WARNING: SYSTEM WILL BE DESTROYED.
   NOW IS THE TIME TO REPLACE THE SYSTEM CARTRIDGE.
/FORMC: OK TO PROCEED (Y,N)?
```

FORMC expects a Yes or No answer. The removable system disc should be replaced with the disc to be formatted before you respond to the prompt. Be aware that FORMC will format the entire disc volume, including the system subchannels if your response is Yes and you do not remove the system disc. FORMC terminates if your response is No.

After the formatting is complete, FORMC issues the message:

```
/FORMC: REPLACE SYSTEM CARTRIDGE TO RETURN TO SYSTEM.
/FORMC: OK TO RETURN (Y,N)?
```

FORMC expects a Yes or No answer. After the system disc is replaced, a Yes response causes a return to the host system. A No response causes FORMC to display the message

```
/FORMC: FINISHED
```

and to execute an infinite loop. The system must then be restored and rebooted.

Spare Command (SP)

NOTE

In order to execute the SPare command, FORMC must be scheduled without renaming (as FORMC:IH) and you must have a capability of 60 or greater. (Refer to the RTE-6/VM Terminal User's Reference Manual for a description of the user capability levels.)

If the rename-inhibit is not included in the call, FORMC aborts with the messages

```
/FORMC: MUST USE THE PROGRAM NAMED `FORMC'
        WITH THE FO AND SP COMMANDS
/FORMC: ABORTED
```

If your assigned user capability is less than 60, and you attempt to spare a disc track, FORMC aborts with the messages

```
/FORMC: INSUFFICIENT CAPABILITY
/FORMC: ABORTED
```

The SPare command is used to spare one or more defective blocks within a track of a CS80 disc. This command can be used only with CS80 discs, including the CTD disc cache memory area. The disc cache is divided into four logical tracks; SP can be used to spare one of these cache tracks. (Refer to the DVM33/DVN33 Driver Reference Manual (92084-90025) for a description of the disc cache memory scheme.)

In runstring mode, you must specify all of the media parameters. Invoke the SPare option as

```
[RU,]FORMC:IH,[list LU],SP,media LU,track
```

where

list LU	is the LU of the optional hard-copy error-logging device. The default is to no hard-copy log.
SP	calls the spare option.
media LU	is the LU of the CS80 disc or CTD disc cache memory to be spared.
track	is the track containing bad blocks to be spared.

When you invoke the spare option as SP in response to the interactive TASK? prompt, FORMC issues the parameter prompt

```
/FORMC: CS80 DISC OR TAPE LU?
```

Enter the LU of the disc that contains defective blocks to be spared. If a CTD tape LU is entered, the disc cache memory area of the CTD is spared. If the specified LU does not correspond to a legal CS80 disc or tape the message

```
/FORMC: ILLEGAL CS80 DISC OR TAPE LU
```

is issued and FORMC reprompts for the LU number. If the LU is that of a tape and the tape does not have a generated disc cache memory, the message

```
/FORMC: NO CACHE DEFINED FOR LU <number>
/FORMC: ABORTED
```

is issued and FORMC exits. When a legal LU number has been entered, FORMC prompts with

```
/FORMC: TRACK TO BE SPARED?
```

Enter the number of the track containing the bad block or blocks. If the number entered is not within the legal range of track numbers for that LU, the message

```
/FORMC: ILLEGAL TRACK NUMBER, LEGAL RANGE 0 TO <last track>
```

is issued, where <last track> is the number of tracks on the LU minus one. FORMC then reprompts for the track number. When a legal track number is entered, FORMC checks the track for bad blocks. Note that the SP option must be invoked separately for each track to be spared.

Once the LU number and track number have been determined, either interactively or through the runstring, FORMC performs an error-rate test on the specified track to determine which blocks, if any, are defective on the track. As each defective block is found, it is removed from active service and the message

```
/FORMC: PHYSICAL BLOCK <address> SPARED FOR <number> BLOCKS.
```

is issued. The <address> is the physical block address and <number> is the physical number of blocks spared. Since there may be more than one bad area covered by the specified track, more than one message may be issued. If the next to last spare track is used, the warning

/FORMC WARNING: ONE SPARE PHY. TRACK LEFT. CONTACT SYSTEM MANAGER.

is issued. Several more sparing operations may be performed if the disc only needs one spare block per track; however, the disc will only be able to use one more physical track for sparing if this need arises. The warning indicates that there is something seriously wrong with the drive or the disc. A complete diagnostic check of the drive and disc should be performed.

If the sparing operation cannot be performed to retain the data in a bad block, FORMC issues the warning

```
/FORMC: DATA MAY BE DESTROYED ON LU <number> TRACK
   <track 1>,<track 2>,.............,<track n>
/FORMC: OK TO PROCEED (Y,N)?
```

A response of Yes results in sparing without retaining the data on the disc tracks indicated by <track 1> through <track n>. A No response causes FORMC to terminate.

If the disc LU bound crosses a physical track that may actually cover another disc LU (such as another FMP directory), FORMC issues the warning

```
/FORMC: DATA MAY BE DESTROYED ON LU
     <lu 1>,<lu 2>,...................,<lu n>
/FORMC: OK TO PROCEED (Y,N)?
```

If the data has been backed up, a response of Yes can be entered and FORMC will lock all LUs that might cross physical tracks and perform the sparing operation. A response of No causes FORMC to terminate.

After a successful sparing operation, FORMC issues the completion message

```
/FORMC: DISC SPARING COMPLETED
```

If no bad blocks are found on the specified disc, the message

```
/FORMC: NO BAD BLOCKS - SPARING NOT PERFORMED
```

is issued and FORMC exits.

Error Messages

CS80 devices return 10 words of status information:

```
word 1 -      identification field
word 2 -      reject errors (severity #1)
word 3 -      fault errors (severity #2)
word 4 -      access errors (severity #3)
word 5 -      information errors (severity #4)
words 6-10 - parameter area for words 2-5
```

One or more error message may be displayed, in the following format:

```
/FORMC: eeeeee ERROR xxxxxxB
   <mnemonic error code>
   IDENTIFICATION FIELD: yyyyyyB     QSTAT: q
   PARAMETER FIELD P(1) THRU P(10):
   ppppppB ppppppB ppppppB ppppppB ppppppB
```

where

eeeeee is the error type: REJECT, FAULT, ACCESS, INFORMATION

xxxxxxB is the error field, in octal

yyyyyyB is the identification field, in octal

q is the error-reporting message from the device

ppppppB is the 5-word parameter area, in octal, for the error

If an undefined error bit is encountered in one of the status words, the message

```
FATAL INTERNAL ERROR
```

replaces the <mnemonic error code> in the message. Refer to the DVM/DVN Driver Reference Manual for a description of the error reporting scheme and a complete summary of all mnemonic error codes.

Chapter 9
On-Line Driver Replacement Utilities

Introduction

The driver replacement utilities DRREL and DRRPL allow you to selectively relocate and install device drivers on-line, thus eliminating the need to regenerate the system each time a driver is changed. DRREL relocates the drivers and DRRPL performs the installation. The actual driver installation (DRRPL DR command) can be issued only by the System Manager, or in a non-session environment. Further restrictions pertaining to driver installation are defined in the DRRPL description. Note that a driver will be recognized only if it conforms to all module and entry-point naming conventions defined in the RTE Operating Systems Driver Writing Manual (92200-93005).

A full driver partition or all of the system driver area can be installed at one time, either in memory or permanently on disc. Table entries are automatically constructed for the replacement drivers and are installed in the Equipment Table (EQT), driver-mapping table and interrupt table.

Finding Space for Drivers

Space for replacement drivers can be obtained in one of three ways: overlaying an existing driver, assigning a driver partition to unused physical pages, or reserving space via a dummy driver at system generation.

Overlaying an Existing Driver

In overlaying an existing driver, you must be sure that other drivers are not affected in the process. In systems generated with base page linking, you can determine the precise bounds of each driver from the system generation listing. However, if the system is generated with current page linking, the links are placed before and after drivers and can be shared among drivers. In this case, you should replace all drivers in the partition or system driver area, or replace only the last driver in the area.

Using Available Pages

By assigning a driver to an unused physical page in the system, you are essentially creating a temporary driver partition. Free physical pages can be identified using the DRREL FR command.

Creating a Dummy Driver

Space for a new driver can be reserved by creating a dummy driver when the system is generated. As an example, the following code could be used to create a two-page driver partition and to reserve base page links:

```
        NAM DVD00,0
        ENT ID00,CD00
        ORB
        REP 40              reserve 40 base page links
        DEC 0
        ORR
ID00    NOP
CD00    NOP
        BSS 2048            reserve driver partition
        END
```

At system generation, the response for the EQT prompt to reserve the partition and to reserve a 20-word EQT extension would be (this example assumes EQT table entry 14)

```
EQT 14?
27,DVD00,X=20
```

If you want to place the driver in the system driver area, the response would be

```
EQT 14?
27,DVD00,S
```

Base Page Links

You should reserve base page only if the replacement driver contains an ORB, if the driver is to fit into a limited logical address space, or if the replacement driver is large.

Base page links for Table Area 1, Table Area 2, SSGA, partitions and system driver areas are allocated starting from 1644B down. In the generation listing, the BP pointer always gives the address of the next link to be allocated, as

```
BP LINKAGE  01611

DP  03  <<PAGE 00030>>

DVD00(0099) 06000  11777
   *ID00    06000
   *CD00    06001

BP LINKAGE  01541
```

Using this example, you would respond to the DRREL prompts as: high base page address, 01611; low base page address (-1), 01541; high logical address, 11777; and low logical address, 06000. (Refer to the DRREL discussion below for the actual prompt format and sequence.)

Loading the Driver Replacement Utilities

Both DRREL and DRRPL require extra memory for building interim tables during the driver relocation/replacement processes. The loading sequence thus requires sizing the programs to allow two pages for table generation. To load the utilities on-line, use the load sequence

```
:RU,LOADR
 /LOADR:  RE %DRREL  or
 /LOADR:  RE %DRRPL
 /LOADR:  EN
:SYSZ,DRREL   or   SYSZ calls system command SZ from FMGR
:SYSZ,DRRPL
 aaaaa  ppp
```

The program size is returned with aaaaa = address of last word +1, and ppp = minimum partition size. Now re-size the program as

```
:SYSZ,DRREL,<ppp+2>  or
:SYSZ,DRRPL,<ppp+2>
```

The driver replacement utilities use the LOADR library, so it is not necessary to specify a library search. Be aware than the driver replacement utilities may not be loaded as extended background.

Driver Relocation Utility (DRREL)

The driver relocation utility DRREL allows you to selectively relocate one or more driver routines, either to a driver partition or to the system driver area, with or without base page links. DRREL can be run interactively from your terminal (the default state), or from a command file. If the input source is a command file, the parameters must be in the same sequence as the interactive prompts.

Invoke DRREL as

```
[RU,]DRREL[,input source]
```

When run interactively, DRREL first prompts for the configuration parameters in the sequence listed below, followed by the relocation command prompt. When your response to a prompt is an octal value, the number must be given with a B suffix, as nnnnB.

```
BASE PAGE LINKS (Y,N)?
```

If you have generated a dummy driver to reserve base page links, respond Y and DRREL will allocate the links. (Refer to the section Finding Space for Drivers for instructions on creating a dummy driver module during system generation.)

If an area has not been reserved, respond N and DRREL will attempt to load the driver or drivers using all current page links. (This is the default state.) Note, however, that large drivers or drivers that must fit into a tight logical address may be required to use base page even if the use of base page links is not specified. DRREL always scans base page to see if it can reuse already existing links.

If your response is YES, DRREL prompts with

```
HIGH BP ADDRESS?
    and
LOW BP ADDRESS - 1?
```

These addresses may be obtained from the system generation listing, as described in the section Finding Space for Drivers. (Note that the low base page address must be specified as address minus 1.) Since the default is to current page linking (unless changed using the relocation command BP), base page links are used only if there is not enough space on the current page. DRREL still scans base page for existing links, but only outside of the allocation area.

If the driver is replaced more than once, the base page links allocated to it may be overwritten, since they are not used by other code.

DRREL then displays the logical address boundaries of the driver partition and the system driver area, as

```
DRIVER PARTITION     n PAGES      11111   hhhhh
          SYSTEM DRIVER AREA      11111   hhhhh
```

where 11111 and hhhhh are the low and high logical address of the partition and system driver area. DRREL then prompts for the low and high logical boundaries of the address space in which the driver code and links can be placed.

```
LOW LOGICAL ADDRESS?
      and
HIGH ADDRESS?
```

When more than one driver is to be relocated, the range specified must include sufficient space for all drivers. If the relocated drivers exceed the boundaries, an error message is issued and DRREL exits.

If the system was generated with all base page links, the responses can be taken directly from the generation listing. However, if the system was generated with current page links, it is safest to relocate all of the drivers in the partition or System Driver Area.

Finally, DRREL issues the prompt

```
OUTPUT FILE?
```

Enter the namr of a disk file to which the memory-image code will be sent. The file will be created as a type 6 file, and may be placed on any disc LU. This file is used by DRRPL to install the driver.

DRREL now prompts with

```
/DRREL:
```

and any of the following commands may be issued. The BPAGE or CPAGE commands, if used, must be given first; all other commands are order-independent. Only the first two characters are required to specify a command.

?? list all available commands with a brief description. The listing format is shown in Figure 9-1.

BPAGE	all linkage is to be through base page. (If no base page was given, links cannot be allocated.)
CPAGE	linkage is to be through current page as much as possible. If base page area is available, it will be used only if room cannot be found on the current page. This is the default state.
LL,namr	specifies the LU of the list device, or the list file namr. The list file may not already exist, unless defined as a list file (' in the first character position of the file name). Non-interactive LUs will be locked. The default is to your terminal.
ECHO	echo all prompts, commands, and comment lines to the list device.
RE,namr	load all modules into the given disc file. The module name, load addresses, entry points, and base page linkages are printed on the list device.
LO	list the current relocation address.
LO,+	set the current relocation address to the beginning of the next page.
LO,addr	set the current relocation address to the given address. The requested address must be greater than the current relocation address and must be below the high address boundary.
SEARCH	search the system library for undefined externals. The module names, load addresses, entry points, and base page linkages are printed on list device.
DISPLAY	display all undefined externals on the list device.
END EXIT /E	end command input. DRREL will search the system library, if required to satisfy any undefined externals and will finish loading the driver. The END, EXIT or /E command may not be given unless a driver is relocated. The /A command must be used to abort without relocating a driver.

/A abort utility. The /A response can be given to any DRREL prompt, and must be used to exit DRREL if a driver is not relocated.

* comment line. DRREL ignores everything on a line beginning with the comment character. If ECHO is specified, all comment lines are echoed to the list device.

At the end of the successful driver relocation, DRREL lists the number of drivers relocated, the beginning and ending logical addresses, and the base page used.

```
??              LIST COMMANDS
BP              ALL LINKS ON BASE PAGE
CP              CURRENT PAGE LINKING (DEFAULT)
RE,<namr>       RELOCATE FILE
LO              LIST CURRENT LOAD ADDRESS
LO,+            SET LOAD ADDRESS TO NEXT PAGE
LO,<addr>       SET LOAD ADDRESS
SE              SEARCH SYSTEM LIBRARY
DI              DISPLAY UNDEFINED EXTERNALS
EC              ECHO INPUT TO LIST DEVICE
LL,<namr>       CHANGE LIST DEVICE
EN,EX,/E        END LOAD
*               COMMENT LINE
/A              ABORT LOAD
```

Figure 9-1. DRREL Command Summary List Format

Example

In the following example two drivers, DVA12 and DVR23, are relocated without using base page linkage, and the memory-image code is placed in output file DP3. The addresses at which each driver initiation and continuation sections will be located in the driver partition are displayed following the RE commands for the drivers. After the user-entered end command, DRREL issues a message defining the number of drivers relocated, the logical address range for the drivers, the output file name and time the file was created, and the utility exit line.

```
  BASE PAGE LINKS (Y,N)? NO

    DRIVER PARTITION      2 PAGES    4000  7777
            SYSTEM DRIVER AREA      20000 23672

  LOW LOGICAL ADDRESS? 4000B

  HIGH ADDRESS? 7777B

  OUTPUT FILE? DP3

  /DRREL: RE,%DVA12

  DVA12                4000  4700   92001-16020  780511  REV 1826
    IA12                 4000
    CA12                 4215

  /DRREL: RE,%DVR23

  DVR23                4701  5614  92202-16001 REV.1913 - 790202
    I.23                 4701
    C.23                 5560

  /DRREL: DI

    NO UNDEFS

  /DRREL: END

  ******     2  DRIVERS   ******
  LOGICAL ADDRESS   4000  5614
     NO BASE PAGE
  /DRREL:  FILE DP3   READY AT 8:26 AM TUE., 14 JULY, 1981
  /DRREL:  END
```

Error Messages

The following error messages can be issued by DRREL. Fatal errors that will cause DRREL to abort are flagged with (F) following the error definition. Note that all errors are fatal when the utility is run non-interactively.

D-IL PRM INPUT ERROR

An invalid command or response was given to a prompt.

D-IL ADD OUT OF RANGE

A bad address was given. Logical addresses given must be within the driver partition or the System Driver Area. Base page addresses must be between the base page fence and the System Communication Area. An address given with the LO command must be greater than the current relocation address and within the high address bound.

D-IL CMD ILLEGAL COMMAND

The BP, CP, and LL commands may not be given after the first RE command. The END command may not be given if a driver has not been relocated.

D-IL ALC ILLEGAL ALLOCATION

A module attempted to allocate common, save, pure code, or EMA area. DRREL does not allow any of these allocations. (F)

D-RF EMA or D-ML EMA

Illegal EMA reference. Drivers may not access EMA. (F)

D-IL REL

Assembler or compiler produced an illegal relocatable module. Recompile and try again. (F)

D-RE SEQ RECORD OUT OF SEQUENCE

May be caused by a corrupt relocatable file. Recompile and try again. (F)

D-DU ENT DUPLICATE ENT

Duplicate entry points were encountered. Rename or remove one of the duplicate ents. (F)

D-CM BLK

Common Block error. Drivers may not allocate common. (F)

D-OV SYM SYMBOL TABLE OVERFLOW

DRREL does not have enough room to do relocation. Use SZ operator command to expand the size of DRREL, use the SE command, or reload DRREL as a large background program. (F)

D-OV FIX FIXUP TABLE OVERFLOW

DRREL does not have enough room to do relocation. Use SZ operator command to expand the size of DRREL, use the SE command, or reload DRREL as a large background program. (F)

D-OV MEM MEMORY OVERFLOW

The relocated code exceeds the logical address boundaries. Find more room for the drivers. (F)

D-IL REC ILLEGAL RECORD

A bad relocatable record was encountered. The file given may not have been a relocatable file, or the file has been corrupted. Give correct file name or try recompiling. (F)

D-CK SUM CHECKSUM ERROR

A bad relocatable file was given. Specify the correct file or recompile. (F)

D-OV BSE BASE PAGE OVERFLOW

Not enough base page links were available to the driver. This may be caused by an ORB encountered when no base page is available, or the space available for links on the current page and base page is not sufficient to relocate the driver. Add base page links or expand the logical address space, which increases current page links available. If the module causing the error is a page or more in length, try starting relocation of the module on a page boundary (do LO,+ if necessary). Setting the current location slightly greater using the LO command may help. (F)

D-UN EXT UNDEFINED EXTERNALS

An END command was attempted, and undefined externals were not satisfied by the system library search. Supply the missing entry points.

D-SY LEN WARNING: SYMBOL TRUNCATED TO 5 CHARACTERS

An entry point or external symbol is greater than five characters in length. DRREL truncates these symbols before storing them in its output file (the input file for DRRPL).

Driver Replacement Utility (DRRPL)

DRRPL controls the installation of replacement drivers relocated by DRREL. The actual installation can only be done by the System Manager, or in non-session mode from the system console. Since DRREL searches for system entry points and may reuse existing base page links, drivers must be installed in the same system on which they were relocated. The driver installation command DR is rejected if DRRPL is loaded in extended background.

DRRPL can be run interactively from your terminal (the default state) or from a command file. If the input source is a command file, the installation parameters must be in the same sequence as the DR command prompts. Note that the DR command itself need not be included in the file.

Invoke DRRPL as

```
[RU,]DRRPL[,input source]
```

If run interactively, DRRPL issues the prompt

```
/DRRPL:
```

and any of the following commands may be issued. In particular, you should issue the DD/MD and FR commands for a listing of the current driver configuration and the number of free pages, since this information is used during the driver installation phase.

?? list all available commands with a brief description. The list format is shown in Figure 9-2.

LL,namr is the LU of the list device, or list file namr. The list file must not already exist, unless it is defined as a list file (' in the first character position of the namr). Non-interactive LUs are locked. When the DR command is issued, all prompts and command inputs are echoed to the list device. The default is your terminal.

/A utility abort. The /A may be issued in response to any DRRPL prompt.

`*` comment line. DRRPL ignores everything on a line beginning with the comment character. Comment lines that are part of the input given with the DR command are echoed to the list device.

`DI[,beginning select code[,ending select code]]`
`MI[,beginning select code[,ending select code]]`

list the Interrupt Table/trap cells from the system on disc (DI) or in memory (MI). The list format is shown in Figure 9-3.

`DE[,beginning EQT#[,ending EQT#]]`
`ME[,beginning EQT#[,ending EQT#]]`

list the Equipment Table entries from the system on disc (DE) or in memory (ME). Only static information is listed. Note: the EQT extension size listed by this command is taken from the system on disc because this value is wiped out in some memory-resident EQT entries. The listing format is shown in Figure 9-4.

`MD`
`DD` list the driver configuration in memory (MD) or on disc (DD). All EQTs listed are grouped according to the driver partition or the System Driver area they reference. The listing format is shown in Figure 9-5.

`FR` list the physical pages that are free in the system. Free pages may be obtained by declaring the pages as bad using the reconfigurator (not within SAM extension). Note that pages in high memory are most frequently free.

`DR,namr` install the new drivers. Namr can be an LU or file that contains the output of DRREL. The sequence of prompts described in the section Replacement Driver Specification begins when this command is entered. Note that this command is valid only if:

1. You are the System Manager (capability >60, or
2. The command is entered at the system console in a non-session environment, AND
3. DRRPL is invoked as RU,DRRPL:IH, to inhibit renaming of the program.

`END`
`EXIT`
`/E` exit DRRPL. Any of these commands may be used to terminate the utility. The /E command also is used to indicate the end of command prompt groups.

```
??            LIST COMMANDS
MD            LIST DRIVER CONFIGURATION (MEMORY)
DD            LIST DRIVER CONFIGURATION (DISK)
ME,<#>,<#>    LIST EQTS (MEMORY)
DE,<#>,<#>    LIST EQTS (DISK)
MI,<#>,<#>    LIST INTERRUPT TABLE, TRAP CELLS (MEMORY)
DI,<#>,<#>    LIST INTERRUPT TABLE, TRAP CELLS (DISK)
FR            LIST FREE PAGES
DR,<namr>     ENTER DRIVER REPLACE MODE
LL,<namr>     CHANGE LIST DEVICE
EX,EN,/E      EXIT
/A            ABORT
*             COMMENT LINE
```

Figure 9-2. DRRPL Command Summary List Format

```
/DRRPL:  DI,10,25

 S.C.          INTERRUPT TABLE          TRAP CELL  (DISC)
 12                     0               JSB 1553,I
 13                     0               JSB 1544,I
 14                     0               JSB 1544,I
 15               EQT  10               JSB 1544,I
 16               EQT   1               JSB 1544,I
 17               EQT   2               JSB 1544,I
 20                PRMPT                JSB 1544,I
 21               EQT   8               JSB 1544,I
 22               EQT   8               JSB 1544,I
 23               EQT   6               JSB 1544,I
 24                PRMPT                JSB 1544,I
 25                PRMPT                JSB 1544,I

Notes:   Select code S.C.:   octal value
         Interrupt Table:    decimal EQT entry, or
                             program name, or
                             octal value
         Trap Cell:          JSB,I, or
                             JMP,I to octal address, or
                             octal value
```

Figure 9-3. DRRPL DI/MI Command Listing Format

```
/DRRPL:  DE,10,25

              EQT   (DISC)

EQT   10 = SC 15 TYPE 65       T=      3  X=    7 IN= 10307 CC= 11162
EQT   11 = SC 40 TYPE 65       T=      3  X=    7 IN= 10307 CC= 11162
EQT   12 = SC 41 TYPE 65       T=      3  X=    7 IN= 10307 CC= 11162
EQT   13 = SC 42 TYPE 66       T=      0  X=   12 IN=  6150 CC=  6161
EQT   14 = SC 42 TYPE 66       T=      0  X=    0 IN=  6150 CC=  6161
EQT   15 = SC 43 TYPE 66       T=      0  X=   12 IN=  6150 CC=  6161
EQT   16 = SC 43 TYPE 66       T=      0  X=    0 IN=  6150 CC=  6161
EQT   17 = SC 44 TYPE 66       T=      0  X=   12 IN=  6150 CC=  6161
EQT   18 = SC 44 TYPE 66       T=      0  X=    0 IN=  6150 CC=  6161
EQT   19 = SC 45 TYPE 67 D     T=      0  X=    0 IN=  6126 CC=  6072
EQT   20 = SC 46 TYPE 37       T=  32767  X=   81 IN=  6122 CC=  6706
EQT   21 = SC 24 TYPE 05   B   T=  12000  X=   13 IN=  6102 CC=  6200
EQT   22 = SC 25 TYPE 05   B   T=  12000  X=   13 IN=  6102 CC=  6200
EQT   23 = SC 26 TYPE 05   B   T=  12000  X=   13 IN=  6102 CC=  6200
EQT   24 = SC 27 TYPE 05   B   T=  12000  X=   13 IN=  6102 CC=  6200
EQT   25 = SC 30 TYPE 05   B   T=  12000  X=   13 IN=  6102 CC=  6200
```

Notes: EQT - entry number, decimal;
SC - select code, octal;
Type - equipment type code, octal;
D - DMA is used;
B - automatic output buffering is used;
S - system driver - does not do own mapping (not shown);
M - system driver - does own mapping (not shown);
T - timeout interval, decimal;
X - EQT extension length in words, decimal;
IN - driver initiation section address, octal;
CC - driver continuation/completion section address, octal.

Figure 9-4. DRRPL ME/DE Command Listing Format

```
/DRRPL:  MD

      DRIVER CONFIGURATION  (MEMORY)
 DRIVER PARTITION     2 PAGES     6000  11777
          SYSTEM DRIVER AREA    26000  27774
 DRIVER PARTITION PAGES     3 -    4     42 -  59

 DRIVER PARTITION   1   <<PAGE     3>>
EQT  1 = SC 16 TYPE 32  D    T=  0  X=  0  IN=   7333  CC=  6145
EQT 10 = SC 15 TYPE 65       T=  3  X=  7  IN=  10307  CC= 11162
EQT 11 = SC 40 TYPE 65       T=  3  X=  7  IN=  10307  CC= 11162
EQT 12 = SC 41 TYPE 65       T=  3  X=  7  IN=  10307  CC= 11162
  .              .            .                .           .
  .              .            .                .           .
  .              .            .                .           .
 DRIVER PARTITION  10   <<PAGE    58>>
EQT  7 = SC 37 TYPE 12    B  T=  0  X=  5  IN=   6000  CC=  6737
EQT  8 = SC 21 TYPE 23  D B  T=  0  X=  0  IN=   7611  CC= 10734

 SYSTEM DRIVER AREA     <<PAGE    11>>
EQT  9 = SC 11 TYPE 50     S T=  0  X=  0  IN=  26000  CC= 26173
EQT 49 = SC 71 TYPE 43     M T=  0  X= 18  IN=  26660  CC= 26663
EQT 50 = SC 72 TYPE 43     M T=  0  X= 18  IN=  26660  CC= 26663
EQT 51 = SC 73 TYPE 43     M T=  0  X= 18  IN=  26660  CC= 26663

  Note:  Addresses are given in octal;
         Driver partition page numbers are given in decimal.
```

Figure 9-5. DRRPL MD/DD Command Listing Format

Replacement Driver Specification

The installation process for the drivers and the associated I/O tables does not begin until the entire sequence of prompts has been completed. The prompt sequence is summarized in Table 9-1, and described in detail in the following paragraphs. Where your response is an octal number, the number must be entered with the suffix B, as nnnnB; decimal numbers do not require a suffix.

Table 9-1. Driver Replacement Prompt Sequence

```
MEMORY OR PERMANENT REPLACE (ME,PE)?

PHYSICAL PAGE OR DRIVER PART.(PG,DP)? (Driver Partition only)
    DP:  DRIVER PARTITION NO.?
    PG:  DP STARTING PHYSICAL PAGE?

EQT NUMBER?
EQ TYPE CODE?                          (Mult. equip. types only)
DMA (Y,N)?
AUTOMATIC OUTPUT BUFFERING (Y,N)?
SELECT CODE?
TIMEOUT INTERVAL?
OWN MAPPING (Y,N)?                     (System Driver Area only)
EQT NUMBER (/E TO END)?

SELECT CODE (/E TO END)?
    ENTRY TYPE (EQT,PRG)               (Driver Partition only)
    ENTRY TYPE (EQT,PRG,ENT,ABS)       (System Driver Area only)
    ENTRY TYPE (ENT,ABS)?              (Select Code 4 only)
        EQT:   EQT NUMBER?
        PRG:   NAME?
        ENT:   NAME?
        ABS:   VALUE?

MEMORY REPLACE READY (Y,N)?

PERMANENT REPLACE READY (Y,N)?
```

NOTE

The display formats cited in this section are examples only. The displays you see when running the driver replacement utility are those related to your specific system configuration and the DRREL output file you have created.

Issuing the DR,namr command causes DRRPL to enter the driver replacement phase of the utility. The DRREL output file is read into memory and DRRPL displays the number of drivers relocated, the beginning and ending logical addresses, and the beginning and ending base page addresses, as defined by the DRREL file. DRRPL then issues the prompt

```
MEMORY OR PERMANENT REPLACE (ME,PE)?
```

You may install the drivers on the system either in memory or permanently on the disc. Your response to this prompt determines the format of the "REPLACE READY?" prompt that is issued to initiate the installation process.

If the driver has been relocated into the driver partitions, the following sequence of prompts is issued:

```
PHYSICAL PAGE OR DRIVER PART.  (PG,DP)?
```

A driver can be installed either in physical pages allocated for drivers at system generation, or in pages unused by the system. If your response is DP to install the driver in the allocated pages, DRRPL issues the prompt

```
DRIVER PARTITION NO.?
```

Enter the driver partition number. The partition numbers can be found in the generation listing, or from the driver configuration listing output of the DD or MD commands if these were issued before you entered the replacement phase. It is advisable to use one of these commands to get the current configuration since DRRPL numbers driver partitions by finding EQTs that reference them. If, for example, all of the EQT entries originally pointing to a particular driver partition are pointed at other partitions, the driver partition is temporarily lost to the utility and can be recovered only by responding with PG to specify the starting page. However, by using the DP response, you can prevent accidental overlay of other drivers. Be sure that the partition contains sufficient pages to hold all of the drivers being installed, or an error will be returned.

If your response is PG to install the driver or drivers in unused or allocated pages, DRRPL prompts with

```
DP STARTING PHYSICAL PAGE?
```

The page specified must be either a page allocated for partition drivers at system generation, or a page unused by the system. The FR command can be used to list the free pages in the system. Be sure that there are sufficient contiguous pages to hold the replacement drivers, or an error will be returned. Free pages are most often found in high physical memory, or pages may be reserved for this purpose by using the reconfigurator to declare bad pages (not within SAM extension). Be aware that free pages may not be used for permanent replacement.

After the driver location is specified, DRRPL displays all of the EQTs that are currently referencing that area, together with the first driver listed in the input file (the DRREL relocation output):

```
 DRIVER PARTITION  3  <<PAGE 35>>
  EQT  3 = SC 23 TYPE 23  D B  T=  9999  X=    0  IN=  4701  CC=  5560
  EQT  7 = SC 12 TYPE 12    B  T=   200  X=    0  IN=  4000  CC=  4215

   DVA12
   IA12   4000
   CA12   4215
```

Be aware that, on replacing a driver, DRRPL does not update the system library to show the addresses of its entry points. DRRPL then prompts with

```
    EQT NUMBER?
```

Enter the EQT number to be modified to reference the listed driver. Select the EQT number from the display list. At least one EQT must be changed for every driver in the input file. (Note: Any EQT using the replaced driver must be redefined here. The information required is the same as the EQT entry in the system generation.)

If the listed driver has multiple initiation/continuation entry-point pairs, DRRPL prompts with

```
    EQ TYPE CODE?
```

Enter the equipment type of the entry points to be referenced by the EQT.The following prompts require a Y or N response.

```
    DMA (Y,N)?

    AUTOMATIC OUTPUT BUFFERING (Y,N)?
```

DRRPL then requests the select code referenced by the EQT and the timeout interval with

```
    SELECT CODE?

    TIMEOUT INTERVAL?
```

If the driver is to be installed in the system driver area, DRRPL prompts with

```
OWN MAPPING (Y,N)?
```

and then issues the prompt

```
EQT NUMBER (/E TO END)?
```

to allow you to specify the modification sequence for any additional EQT entries that must reference the given driver. If you have made an error in any of the preceding entries, you can repeat the full sequence for the EQT at this time by entering the EQT number again.

If more than one driver was relocated by DRREL, the next driver and entry points are displayed. The EQT modification prompt sequence is repeated for all relocated drivers.

The information given during system generation for any EQT redefined above must be redefined in this section (i.e., respecify what you did for the interrupt table entry in the generation).

After the EQTs have been modified for all the drivers in the input file (/E response to the EQT NUMBER? prompt), DRRPL displays the default Interrupt Table changes, as

```
INTERRUPT TABLE CHANGES:
S.C.      INTERRUPT TABLE    TRAP CELL  (MEMORY)
 12          EQT   7         JSB 1601,I
 23          EQT   3         JSB 1601,I
```

The Interrupt Table entry corresponding to the select code for each modified EQT will, by default, contain the address of that EQT. The trap cells associated with the Interrupt Table entries will contain a JSB LINK,I where LINK contains the address of $CIC. If the system uses OS microcode, a call to the equivalent microcode routine is placed in the trap cell. DRRPL then issues the prompt

```
SELECT CODE (/E TO END)?
```

to allow you to modify the interrupt table/trap cells. If you enter a select code, DRRPL responds with one of the following prompts:

```
ENTRY TYPE (EQT,PRG)?              (Partition drivers)

ENTRY TYPE (EQT,PRG,ENT,ABS)?      (System Driver Area)

ENTRY TYPE (ENT,ABS)?              (Select code 4)
```

The action taken by DRRPL for each of the entry type responses is

Entry Type	DRRPL Prompt	Action
EQT	EQT NUMBER?	The address of the given EQT is placed in the interrupt table, and a JSB LINK,I is placed in the trap cell. If the system uses OS microcode, a call to the equivalent microcode routine is placed in the trap cell. Be aware that only modified EQT entries can be specified in response to this prompt.
PRG	NAME	The negated ID segment address is placed in the interrupt table. A JSB LINK,I or OS microcode call is placed in the trap cell.
ENT	NAME	The entry point name must be contained in one of the replacement drivers. Zero is placed in the interrupt table and a JSB,LINK,I is placed in the trap cell. In this case, the LINK must pre-exist or be explicitly placed on base page. This can be accomplished in the driver by the following sequence of instructions: ORB DEF ENT ORR DRRPL will search the allocated base page links and the existing base page links for a link to the given ENT.
ABS	VALUE	Zero is placed in the interrupt table and the given value is placed in the trap cell.

DRRPL continues to prompt for the SELECT CODE modification sequence until you enter /E to end. If you make an error in any of the select code modifications, the process can be repeated for the code in error.

Driver Replacement

After all system modifications have been made (/E response to the SELECT CODE? prompt), DRRPL displays a summary of the changes to be made, as

```
 SUMMARY OF SYSTEM CHANGES:
 ************************************************************
 DRIVER PARTITION    n     <<PAGE     nn>>
 LOGICAL ADDRESS     nnnn    nnnn
    NO BASE PAGE
       (or)
    BASE PAGE

 EQT CHANGES:
   (changes)

 INTERUPT TABLE CHANGES:
   (changes)
```

If the driver is to be installed in memory, DRRPL issues the warning and prompt

```
    THIS CAN CRASH YOUR SYSTEM!  THE SYSTEM SHOULD BE
    INACTIVE.  EQTS TO BE REPLACED MUST NOT HAVE
    REQUESTS PENDING.

    MEMORY REPLACE READY (Y,N)?
```

If your response is Y, DRRPL turns off the interrupt system and checks to be sure that requests are not pending on any EQT to be replaced. If requests are pending, DRRPL turns on the interrupt system, issues the error message

```
    EQTS BUSY--NO REPLACEMENT DONE
```

and reprompts for the OK to replace.

If the existing driver cannot process the queued requests, you can OFF the programs making the requests or you can use the DN command to down the LUs that reference the driver, and temporarily redirect these LUs to the bit bucket (System LU command).

When all requests are cleared, DRRPL installs the new drivers and associated I/O tables and exits.

If the driver is to be installed permanently on disc, DRRPL issues the warning and prompt

```
THIS CAN DESTROY YOUR SYSTEM!  MAKE SURE THE SYSTEM IS
BACKED UP!  DOUBLE CHECK YOUR CHANGES TO THE SYSTEM
BEFORE PROCEEDING.  TURN OFF DISK WRITE PROTECT.

PERMANENT REPLACE READY (Y,N)?
```

Since DRRPL checks only to see that the write-protect switch is off, you should make a practice of backing up the system before turning off the write protect.

Driver Replacement Example

In the following example two drivers relocated by DRREL (DRRPL input file DP3) are installed in memory, in the driver partition. The example is limited to the driver replacement phase; refer to the related illustrations for the configuration command output formats.

```
/DRRPL:  DR,DP3

     2  DRIVERS
LOGICAL ADDRESS    4000  5614
   NO BASE PAGE

MEMORY OR PERMANENT REPLACE (ME,PE)? ME

PHYSICAL PAGE OR DRIVER PART. (PG,DP)? DP

DRIVER PARTITION NO.?  3

DRIVER PARTITION    3   <<PAGE    35>>
 EQT    3 = SC 23 TYPE 23  D B   T=  9999  X=    0 IN=  4701 CC=  5560
 EQT    7 = SC 12 TYPE 12    B   T=   200  X=    0 IN=  4000 CC=  4215

WARNING:  MAY BE OVERLAYING DRIVERS.  TYPE '/A' TO ABORT

DVA12
IA12    4000
CA12    4215

EQT NUMBER?  7

DMA (Y,N)? NO

AUTOMATIC OUTPUT BUFFERING (Y,N)?  YES

SELECT CODE? 12B
```

```
TIME OUT INTERVAL? 200

EQT NUMBER (/E TO END)?  /E

DVR23
I.23    4701
C.23    5560

EQT NUMBER?  3

DMA (Y,N)? Y

AUTOMATIC OUTPUT BUFFERING (Y,N)?  Y

SELECT CODE? 23B

TIME OUT INTERVAL? 9999

EQT NUMBER (/E TO END)?  /E

INTERRUPT TABLE CHANGES:
 S.C.   INTERRUPT TABLE   TRAP CELL  (MEMORY)
  12         EQT   7      JSB 1601,I
  23         EQT   3      JSB 1601,I

INTERRUPT TABLE MODIFY:

SELECT CODE (/E TO END)? 24B

ENTRY TYPE (EQT,PRG)? EQT

EQT NUMBER?  3

SELECT CODE (/E TO END)? /E
```

```
SUMMARY OF SYSTEM CHANGES:
**************************************************************
DRIVER PARTITION    3    <<PAGE    35>>
LOGICAL ADDRESS    4000  5614
   NO BASE PAGE

EQT CHANGES:
 EQT    7 = SC 12 TYPE 12     B    T=    200   X=    0 IN=   4000 CC=   4215
 EQT    3 = SC 23 TYPE 23   D B    T=   9999   X=    0 IN=   4701 CC=   5560

INTERRUPT TABLE CHANGES:
 S.C.    INTERRUPT TABLE    TRAP CELL  (MEMORY)
  12          EQT   7       JSB 1601,I
  23          EQT   3       JSB 1601,I
  24          EQT   3       JSB 1601,I
**************************************************************

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

        WARNING!    WARNING!    WARNING!    WARNING!

      THIS CAN CRASH YOUR SYSTEM!  THE SYSTEM SHOULD BE
      INACTIVE.  EQTS TO BE REPLACED MUST NOT HAVE
      REQUESTS PENDING.

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

MEMORY REPLACE READY (Y,N)?  Y
```

Error Messages

The following error messages can be issued by DRRPL. Fatal errors that will cause DRRPL to abort are flagged with (F) following the error definition. Note that all errors are fatal when the utility is running non-interactively.

DRRP 001 INPUT ERROR

An invalid command or response was given to a prompt

DRRP 002 OUT OF RANGE

An EQT number, driver partition number, physical page number, or select code was given which does not exist. An invalid timeout interval was given; the range is 0 through 32767 (decimal). Check the system generation listing for the correct response.

DRRP 003 NOT FOUND

An entry point or program was not found. This may be caused by a bad equipment type code, or an entry point not in one of the replacement drivers.

DRRP 004 BAD SELECT CODE

A select code may not be between 5 and 7, may not be the select code of the privileged interrupt card, may not be less than the privileged interrupt card for the partition driver, and may not be 4 for partition drivers.

DRRP 005 NOT AVAILABLE

A physical page requested as part of a driver partition is being used by the system. Pages may be reserved by declaring bad pages with the reconfigurator.

DRRP 006 BAD EQT NUMBER

An EQT number was given that was used with a previous driver. An Interrupt Table entry can only be changed to reference one of the modified EQTs. Enter a valid EQT number.

DRRP 007 PARTITION TOO SMALL

The replacement drivers are too big to fit in the requested driver partition without overlaying part of another. Override this check by specifying the driver partition starting physical page. This error may also occur if a requested driver partition will overlay pages used by the system.

DRRP 008 LINK NOT FOUND

No link to the given entry point was found on base page in either the existing links or in the replacement links. You must provide a base page link to the entry point.

DRRP 011 INVALID FILE

An incorrect file name was given, the file is corrupt, or it was created by the wrong version of DRREL. Give the correct file name or recreate the file with DRREL. (F)

DRRP 012 FILE CREATED ON ANOTHER SYSTEM

The input file given was created by DRREL running on a different system. Drivers must be relocated on the same system in which they will be installed. Re-run DRREL on this system.

DRRP 013 TABLE OVERFLOW

DRRPL uses the space between the end of the program and the end of its logical address space to store the input file and to build new I/O tables to be installed in the system. Use the SZ command to increase the space for DRRPL, or reload DRRPL as a large background program. (F)

DRRP 014 EQTS BUSY--NO REPLACEMENT DONE

DRRPL will not replace a driver in memory until all pending requests on EQTs to be replaced are cleared. If the existing driver cannot process the requests, you can OFF the programs making the requests and down those LUs that reference the EQT using the system DN command, and temporarily redirect the LUs to the bit bucket with the system LU command. If this does not correct the problem, you may have to reboot the system.

DRRP 015 CORRUPT SYSTEM

There is no link to $CIC on base page, or the replacement driver links will overlay this link. Check your generation listing and specify different base page addresses for the driver. (F)

DRRP 016 NOT SYSTEM MANAGER

An attempt was made to use the DR command running under an account other than the system manager, or running non-session from a terminal other than the system console. Log on correctly and call DRPL as RU,DRRPL:IH.

DRRP 017 NOT 'DRRPL'

An attempt was made to use the DR command with a copy of the program. Run the program with the command RU,DRRPL:IH or reload the program with the LOADR command OP,DC.

DRRP 018 SYSTEM DISK WRITE ERROR

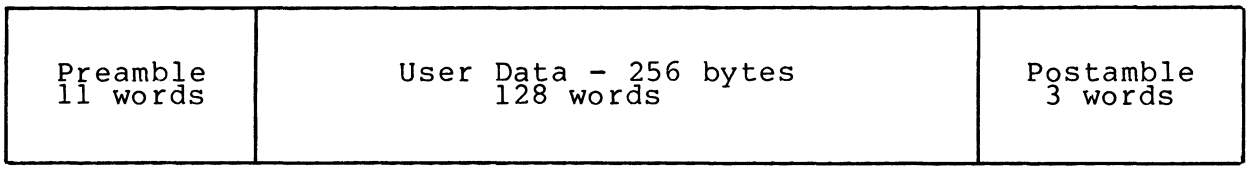
An error occurred on an attempt to do a permanent replacement. This error is usually caused because the disk is write protected. Turn off the write protect and try again.

DRRP 019 PROGRAM CANNOT BE EXTENDED BACKGROUND

An attempt was made to use the DR command running DRRPL as an extended background program. Reload DRRPL so that it is not extended background.

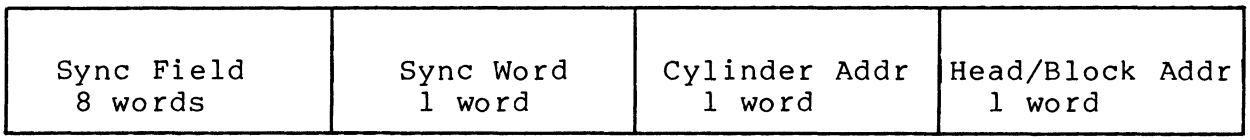
Appendix A
Initializing, Formatting, and Sparing Discs

All data is written on ICD/MAC discs in blocks of 256 bytes (128 words). The term block will be used here, since "sector" is defined in the RTE environment to mean 128 bytes, or 64 words. Each block has three components: the preamble, the data area and the postamble, as



The postamble consists of the Cyclic Redundancy Check (CRC) word and one word reserved for future use.

The 11-word preamble consists of four fields, as



The synchronization field provides the timing signals that indicate the start of the block.

The synchronization word is a fixed value for the disc drive type (7905, 7906, 7910, 9895, etc.).

The cylinder address is contained in bits 15-0 of the address word.

The head/block address word contains the block address in bits 7-0, the head address in bits 12-8, and the block status in bits 15-13.

The preamble is more complex for the hard disc than for the flexible disc. (For flexible discs supplied by HP, only the D bit is significant.) The status portion of the preamble includes three bits:

S bit: (bit 15). Indicates that the track is a spare for a defective track. Only hard discs supplied by HP permit sparing.

P bit: (bit 14). Indicates a protected track. A protected track is a read-only track unless the FORMAT switch is activated on the disc drive.

D bit: (bit 13). Indicates that the track is defective and should be spared or not used at all.

In most cases, the address associated with the block will be the address of the block itself; i.e., it will point to itself. The exceptions are:

Hard Disc: If the D bit is set (track defective), the address points to the track used as a spare.

If the S bit is set (the track is a spare), the address points to the defective track for which it is a spare.

Flexible Disc: If the D bit is set (track defective), the address is not relevant (may be anything).

The RTE utility programs will declare an entire track bad if one bad block is found on the track. This method of management results in less head movement (hence faster access) when encountering spared blocks (tracks).

For the hard disc, once a track has been spared, the disc controller finds that the track is spared whenever a seek is made to that defective track. The disc drive will automatically access the spare track instead. This process is transparent to the software.

For the flexible disc, although the technique is not called sparing, the effect is virtually the same. Once the D bit has been set for the defective tracks, the utility gives the drive a command to skip bad tracks. For example, the outermost track is normally track zero. But if the D bit has been set for this track, then when commanded to access track zero, this track (and any following bad tracks) will be skipped. The first good track becomes track zero.

If a bad track develops in using the disc, the hard disc sparing capability has the advantage that a simple recovery process is possible by running the SPare function. This is not possible with the flexible disc.

Initializing and Sparing a Hard Disc

Disc initialization uses the track map table in RTE-6/VM to determine the starting cylinder, head number, number of tracks and number of spare tracks for each disc LU. Tracks in the spare pool are used only as replacements for the defective tracks - they are otherwise unused.

The initialization procedure is as follows:

1. Clean up tracks in the spare pool. The full block (including the preamble and postamble) is read and the status bits examined. If the D bit is set (defective) go to next track. A track from the spare pool will not, itself, be spared. If the D bit is not set, rewrite the full track with zeroes, including the S, P, and D bits. Verify by reading back the track. If found bad during verify, set the D bit.

2. Read each track in the mapped portion of the LU. If the D bit is set or if the read is unsuccessful, set the flag indicating sparing is needed.

3. If sparing is needed:

   a. Prepare the preamble of the defective track: Set the D bit and set the address of the spare track in the preamble of the defective track.

   b. Prepare the preamble of the spare track. Set the S bit and set the address of defective track in the preamble of the spare track.

4. Write initialize the track. If sparing is required, use the preambles set in step 3. If sparing is not required (good track found in step 2), the address will be that of the good track; it will point to itself. If this is a spare track, the address will be that of the defective track (a backward pointer). If this is a defective track, then the address will be that of the spare track (a forward pointer).

The same procedure is used to spare an individual track (SP) except that, where possible, the data buffer is copied from the defective track on a block-by-block basis. In recovering user data, an offset read is performed to pick up data that could not normally be read with the head aligned to the center of the track.

5. Verify the whole track on a block-by-block basis. If successful, move to the next track and repeat the process. If unsuccessful, get the next spare track and repeat the sparing procedure.

Formatting a Flexible Disc

Formatting is the most basic form of initialization. Normal reads and writes cannot be made to a disc until it has been formatted. The initialization process for the hard disc assumes that the disc has been previously formatted at least once. The formatting process writes the full block as described above, tagging the block with an address. Initially this address points to itself, but may be changed if the block is defective or a spare.

A hard disc is always pre-formatted before shipment from HP, to ensure the interchangeability of disc packs. Therefore the capability to format a hard disc is not included in the FORMT and FORMC utilities. A flexible disc, however, must be formatted before it can be used initially.

Interleaving (Fill Number)

When formatting a flexible disc, you can define the order in which consecutive logical blocks are accessed on the disc by specifying a "fill number" in the FOrmat function runstring. This value defines the number of physical blocks to be skipped between logical blocks. If the fill number is zero, then the N+1th block is accessed following the Nth block. Otherwise, the number of blocks specified by the fill number are skipped. Once the disc is formatted, the fill number is transparent to the user software.

The fill number may be chosen based upon a knowledge of how the user software is to access the disc and a knowledge of the rotational speed of the disc. For example, suppose that the software takes T milliseconds to read and process one block and it picks up sequential blocks. If the time to rotate once around the disc is divided by the number of blocks per track, then the time to move over a single block is found. The fill number is then computed so that the number times the time to scan one block is slightly greater than the processing time. Thus, the next block will rotate into the proper position for access just when it is needed.

As an example, assume

360 RPM (rotations/minute) = 167 milliseconds/rotation
30 blocks/track = 5.6 milliseconds/block

If the processing time is less than 0.9 ms, then a fill number = 1 can be specified, since processing of the first block will be completed before the next block becomes available. If processing time is between 5.6 ms and 11.2 ms, a fill number = 2 should be specified. If the number were 0 or 1, the disc would have to rotate one full revolution before the next block could be picked up (167 ms versus less than 5.6 ms).

Most user programs accessing files will define a Data Control Block (DCB) of 144 words, of which 128 words (1 block) will be used for disc transfers. Therefore, a minimum fill number can be computed in terms of the amount of time to complete one EXEC request to write a block and return to the user program. Or the value can be determined experimentally by trying several fill numbers. For sequential access to the files with a DCB of 144 words, an approximate number to use would be in the range 4 to 6.

Of course, the higher the fill number, the longer the formatting process will take. The performance graph in Figure A-1 plots the formatting time against the fill value.

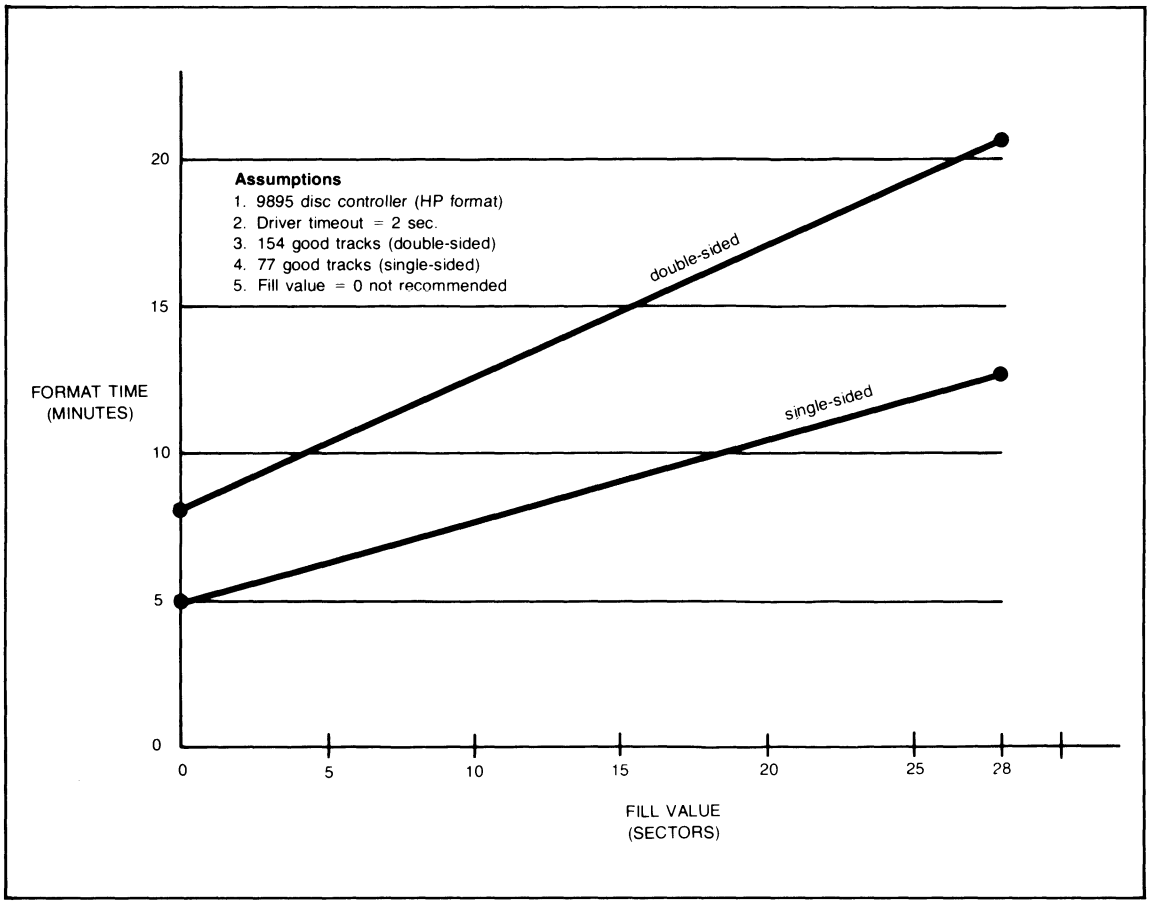


Figure A-1. Formatting Time vs. Fill Value

Index

!BCK01, 6-30
!BCK02, 6-30
!BCK03, 6-31
!option, FC, 4-12
[] optional parameters, 1-2
{}, namr list delimiters, FC, 4-8
"" keyword identifier, GENIX, 2-27
& text continuation identifier, GENIX, 2-27
, command delimiter, 1-2
/ interactive prompt, 2-21
7900 disc subchannel definition, 6-42, 6-47, 6-50

A

Abort Group command, FC, 4-18
active program status summary, 2-10
AG command, FC, 4-18
AL option, WHZAT, 2-5, 2-6
ALLOC instruction translation, 3-3
appending files, MERGE, 2-39
ASCII CRN, abbreviated form, FC, 4-11
ASCII headers structure, 6-56
Auxiliary disc overlaying, 6-9

B

B option, FC, 4-12
background programs summary, 2-11
base page links, 9-2
BF, backspace tape, 6-53
Brief status-display format option, FC, 4-12

C

C option, FC, 4-12
calling FORMC, 8-22
calling FORMT, 8-2
calling PCOPY interactively, 6-27
calling PRSTR interactively, 6-19
calling PSAVE interactively, 6-12
calling RE interactively, 6-43
calling READT, 5-6
calling SA interactively, 6-51
calling WRITT, 5-1
cartridge allocation, FC, 4-25

cartridge backup utility programs, 5-1
Cartridge List commands, FC, 4-22
cartridge lock option, FC, 4-14
cartridge open option, FC, 4-14
cartridge save, 5-1
CAUTION,
   FO, format command, 8-27
   FORMT, 8-2
   IN, initialize an LU, 8-8
   PCOPY, 6-27
   PRSTR, 6-22
   RE, reformat a disc, 8-14, 8-15, 8-16
   SP, spare a track, 8-11
   tape positioning, 6-5
   VE, verify a disc, 8-17
   WRITT, 5-2
CF, name comment file, FC, 4-6
CL command, FC, 4-22
CLAL Command, FC, 4-22
Clear destination disc option, FC, 4-12
CLOAD utility,, 2-34
   calling, 2-34
   examples, 2-36
   session maximum, spool files, 2-31
CMD utility,, 2-20
   calling, 2-21
   error messages, 2-25
   examples, 2-23
CO command examples, 4-16
CO command, disc-to-disc copy, LIF, 7-5
command out-of-sequence errors, 4-47
command set summary, 4-2
command summary function, 4-4
command syntax, 1-2
command syntax, parameter errors, 4-44
comment file record length, FC, 4-6
compile and load utility CLOAD, 2-34
compile utility, COMPL, 2-31
COMPL utility,, 2-31
   calling, 2-31
   non-spooling, 2-32
   session maximum, spool files, 2-31
concatenating files MERGE, 2-37
configuration commands, FC, 4-5
control characters, GENIX, 2-27
control statement override, 2-32
COpy command options, 4-11
COPY command, FC, 4-7
copy status messages, FC, 4-6
create keyset, 2-53
creating a dummy driver, 9-2
CS80 cartridge tape handling, 6-4

CS80 disc maintenance utility FORMC, 8-20
CS80 disc subchannel definition, 6-42, 6-47, 6-50
CS80 pushbutton format, 6-9
CTD define LU subchannel, 6-4
CTD tape maintenance utility FORMC, 8-20
CTD tape records, 6-57
CTD,
   block, 8-21
   disc cache, 8-21, 8-32

D

D option, FC, 4-12
data block, A-1
DATA instruction translation, 3-4
data structures, 6-55
data verification, on-line physical backup, 6-6
DBL instruction translation, 3-3
DD, list driver configuration command, 9-12
DE command, FC, 4-15
DE, list EQT entries command, 9-12
deadlock situations, 2-5
DEBUG instruction translation, 3-4
default command, FC, 4-15
destination disc handling, 4-25
DI, list Interrupt Table command, 9-12
Directory List command, FC, 4-19
disc backup utilities, compatibility, 6-1
disc data I/O errors, 4-37
disc data verification, 6-7
disc definition record, 6-56
disc drives, compatibility, 6-2
disc formatting utilities, 8-1
disc formatting, FORMC, 8-30
disc read verification, FC, 4-15
disc unit protect switch, 6-44
disc write verification, FC, 4-15
disc-caching, 6-4
disc-to-disc copy (CO), 6-46
disc-to-tape copy verify errors, FC, 4-43
discs, preformatted, 8-1
DL - directory list command, LIF, 7-6
DL command, FC, 4-19
drive protect switch, 6-21
driver installation, 9-11
driver partition number, 9-17
driver relocation utility (DRREL), 9-4
driver replacement, 9-21
driver replacement utility (DRRPL),, 9-1, 9-11
   loading, 9-3
driver timeout, 8-24

drivers,
   creating dummy, 9-2
   finding space for, 9-1
   overlaying existing, 9-1
DRREL command summary list format, 9-7
DRREL driver relocation utility,, 9-4
   error messages, 9-9
   sizing, 9-3
DRRPL driver replacement utility,, 9-11
   error messages, 9-25
   sizing, 9-3
DRT (Device Reference Table), 8-5
Duplicate file replace option, FC, 4-12

E

E option, FC, 4-13
EC, echo command, FC, 4-7
echo command default, FC, 4-7
EG command, FC, 4-18
Eliminate extents option, FC, 4-13
EMA instruction translation, 3-4
End Group command, FC, 4-18
END instruction translation, 3-4
ENT instruction translation, 3-3, 3-4
error checking LIF,
   level, 7-11
   override, 7-11
error messages, 2-20, 2-25, 2-51, 5-4, 5-11, 6-61, 7-13, 8-18, 8-34
    9-25
errors affecting single file, 4-38
errors requiring operator action/response, 4-32
errors that cause rejection of current source tape volume, 4-43
errors that terminate current command, FC, 4-47
errors that terminate FC, 4-50
EXT instruction translation, 3-3
extended background programs summary, 2-11

F

F option, FC, 4-12
father-son relationship, 2-5
FC commands, 4-2
FC, running, 4-1
FC, file copy utility, 4-1
FF, forward space tape, 6-53
File Copy utility, FC, 4-1
file descriptor legality checking, LIF, 7-1
file manipulation utilities, MERGE, SCOM, 2-37
file ranges, FC, 4-9

file descriptor legality checking, LIF, 7-1
file manipulation utilities, MERGE, SCOM, 2-37
file ranges, FC, 4-9
files,
   append, 2-39
   concatenate, 2-37
   keyed text, 2-26
   transfer, 2-27, 2-37
   unit save tape, 6-5
fill number, A-4
flexible disc,
   formatting, sparing, A-2, A-4
   verification, 8-17
FMGR error Help function, FC, 4-5
FO, format a flexible disc, 8-4
   using, 8-5
FO, format command, 8-27
format a flexible disc (FO), 8-4
formatting, A-1
FORMC utility, 8-20
   options, 8-21
   runstring, 8-22
FORMT utility, 8-1
   calling, 8-2
   error messages, 8-18
   loading, 8-2
FORTRAN, statement translation, 3-2, 3-5
Full status-display format option, FC, 4-12
function keys, display, 2-58
Function-key Manipulation Utilities, 2-52

G

GEN instruction translation, 3-4
GENIX utility, 2-26
   calling, 2-26
   text format, 2-28
globals definition, FC, 4-28
GR command, FC, 4-18

H

hard disc, verification, 8-17
HE - Help, LIF, 7-4
help function, 6-43
help function, FC, 4-4
HELP utilities,
   calling, 2-17
   error messages, 2-20
   examples, 2-18

I

I option, FC, 4-13
ICD disc subchannel definition, 6-36
Ignore data errors option, FC, 4-13
ignore specified lines, 2-41
IN, initialize an LU, 8-7
information messages and warnings, FC, 4-33
initialize an LU (IN), 8-7
initializing and sparing a hard disc, A-3
input command file, PRSTR, 6-17
input file format GENIX, 2-27
interleaving (fill number), A-4
interleaving, 8-4, 8-29
IO, display I/O information, 6-30

J

JOBFIL, 2-32

K

K option, FC, 4-13
Keep tape on-line option, FC, 4-13
KEY NOT FOUND message, 2-18
KEY NOT UNIQUE message, 2-18
key, GENIX, 2-28
KEYS, 2-52
   calling, 2-53
keyset,
   create, 2-53
   list, 2-55
   modify, 2-56
KYDMP utility, 2-57
   calling, 2-57

L

L option, FC, 4-14
large background programs summary, 2-11
LC command, FC, 4-22
LGTAT, 2-14
   abbreviated output, 2-14
   complete output, 2-15, 2-16
LH command, FC, 4-22

LIF utility,, 7-1
   calling interactively, 7-4
   commands executed, 7-4
   copy command, 7-5
   default values, 7-7
   errors, 7-12
   exit command, 7-12
   files, 7-2, 7-3
   HE command, 7-4
   initialization command, 7-7
   list command, 7-8
   logical list command, 7-8
   mount cartridge command, 7-8
   naming conventions, 7-3
   pack cartridge command, 7-9
   purge command, 7-9
   rename command, 7-9
   set severity level command, 7-11
   store command, 7-10
   transfer command, 7-11
List Comment file command, FC, 4-22
list device, LL, 4-5
list directory: DL, 4-19
List Header file command, FC, 4-22
LIST keyset, 2-55
listing commands, FC, 4-19
LL, list device, FC, 4-5
loading FORMT, 8-2
loading the off-line utilities, 6-30
loading the on-line utilities, 6-10
LOADR command file #FC6, FC, 4-26
LOADR default options, 2-34
lock cartridge, 6-9
Lock cartridge option, FC, 4-14
lock disc, 6-9
LOD instruction translation, 3-4
loss of unidentified files on copy from tape, FC, 4-42
LU 2 overlaying, 6-9
LU 3 overlaying, 6-9
LU save, 6-6
   saves one disc LU, 6-11
LU saves, 6-56
LUs and EQTs status summary, 2-7

M

MAC disc subchannel definition, 6-42, 6-47
MACRO/1000 OLDRE restrictions, 3-2, 3-4
master security code, FC, 4-10
MD, driver configuration listing, 9-12
ME, EQT listing, 9-12

memory-based operating system loading, 6-33
memory-resident programs summary, 2-11
MERGE utility,, 2-37
message continuation, GENIX, 2-27
messages logged by FC, 4-32
MI, Interrupt Table listing, 9-12
modify keyset, 2-56
multiple saves, 5-3
multiple-volume tape sets, 6-6

N

NAM instruction translation, 3-3
name comment file, CF, 4-6
new record identifier field, 3-1
non-fatal tape read errors, FC, 4-37
non-session environment, READT, 5-7
non-spooling option, GENIX, 2-32
non-translatable record, OLDRE, 3-1
null cartridge field, FC, 4-10
null security code, FC, 4-10

O

O option, FC, 4-14
off-line physical backup utility, 6-29
  loading from cartridge tape, 6-31, 6-37
  loading from magnetic tape, 6-32, 6-38
  loading with MTL ROM, 6-33
  loading without MTL ROM, 6-32
  PBO I/O reconfigurator, 6-34
  system console operations, 6-38
OLDRE,
  error conditions, 3-2
  namr format, 3-2
  non-translatable record, 3-1
  processing speed languages, 3-2
  running, 3-2
on-line physical backup utilities, 6-9
on-line utilities, loading, 6-10
Open cartridge option, FC, 4-14
operating environment, RTE-6/VM, 1-1
optional parameters, 1-2
overlaying LU 2, LU 3, 6-9
overlaying an existing driver, 9-1

P

P option, FC, 4-14
PA option, WHZAT, 2-8, 2-9
partition status summary, 2-8
Pascal statement translation, OLDRE, 3-2, 3-5
PBU I/O reconfigurator, 6-34
PCOPY,, 6-26
   calling interactively, 6-27
   example, 6-28
   options, 6-27
performance considerations, FC, 4-26
physical backup utilities loading off-line, 6-30
physical backup utilities,, 6-2
   unit save tape files, 6-5
physical backup utility, off-line, 6-29
physical tracks reserved for spares, 8-21
PL option, WHZAT, 2-10, 2-12, 2-13
postamble, data block, A-1
preamble, data block, A-1
program page requirements FORMT, 8-2
program segment summary, 2-11
program states, 2-2
program status summary, 2-5
program suspension messages, 2-4
protect switch disc drive, 6-44
PRSTR,, 6-17
   calling interactively, 6-19
   error returns, father program, 6-22
   example, 6-23
   options, 6-19
PSAVE,, 6-11
   calling interactively, 6-12
   examples, 6-14
   options, 6-12
Purge source-file option, FC, 4-14
pushbutton save/restore data verification, 6-7

R

RE, off-line restore, calling interactively, 6-43
RE, reformat a disc, FORMT, 8-14
read verification, 6-7
READT,
   calling, 5-6
   error messages, 5-11
   non-session environment, 5-7
   restore cartridge, 5-6
real-time programs summary, 2-11

reconfiguration process generation process, 6-34
recover Unused space option, FC, 4-15
referencing devices off-line, 6-30
reformat a disc (RE), 8-14
replace Duplicate files option, FC, 4-12
replacement driver specification, 9-15
restore disc cartridge, READT, 5-6
restore tape file (RE), 6-40
RPL instruction translation, 3-4
running FC, 4-1
running OLDRE, 3-2
runstring mode, FORMC, 8-23
RW, rewind tape, physical backup utility, 6-53

S

S option, FC, 4-14
SA, calling interactively, 6-51
save definition record, 6-56
save disc to tape (SA), 6-49
SC, scratch area definition, FC, 4-7
scheduling SCOM programmatically, 2-41
SCOM, file compare utility,, 2-39
   examples, 2-43
   options, 2-40
   runstring, 2-39
   scheduling programmatically, 2-41
   status interrogation, 2-43
scratch area definition: SC, 4-7
sector, A-1
sector/track value differences, READT, 5-8
select code, 9-19
setting tape device off-line, FC, 4-25
Single-volume copy option, FC, 4-14
Skip option, FC, 4-24
SM option, WHZAT, 2-5, 2-7
soft key command set dump, 2-57
Soft key programming, 2-52
softkey command set, modifying an existing, 2-56
source, destination parameter considerations, FC, 4-10
SP, spare command, FORMT, 8-10, 8-31
sparing, A-1, A-2
status interrogation, SCOM, 2-43
status utility (WHZAT), 2-1
subchannel definition,, 6-29
   forced, 6-41
SV - severity level definition, LIF, 7-11
system BReak command, response to, 8-23
system disc overlaying, 6-9

T

T option, FC, 4-14
tape data verification, 6-6
tape device lock,unlock, FC, 4-25
tape format, FC, 4-23
tape formatting, 8-28
tape handling, FC, 4-23
tape positioning, 6-4
tape protection, FC, 4-25
tape read verification, FC, 4-15
tape write verification, FC, 4-15
tape verify, 5-9
Tape-length-required display option, FC, 4-14
text files, 2-26
TI, title command, FC, 4-6
title command, TI, 4-6
title, maximum length, FC, 4-6
TR command, FC, 4-23
track assignment summary, 2-15
track assignment table log utility, 2-14
track map redefinition, 6-24
transfer to input LU (TR), 6-30
translation results, 3-3

U

U option, FC, 4-15
unit save, 6-44
unit save tape files, 6-5
Unused space recovery option, FC, 4-15
using FMGR globals, FC, 4-28
using globals in transfer files, FC, 4-30

V

V option, FC, 4-15
VE command, FORMT, 8-24
VE, verify a disc, 8-17
verification of restore, 6-6
verification of saves, 6-6
Verify data option, FC, 4-15
volume, 8-21

W

wait state messages, program, 2-5
WHZAT, 2-1
   AL/SM options, 2-5
   PA option, 2-8
   PL option, 2-10
wildcard characters, FC, 4-8
write tape utility (WRITT), 5-1
write verification, 6-6
WRITT utility,
   calling, 5-1
   examples, 5-3

X

XDBL instruction translation, 3-3
XEND instruction translation, 3-4
XENT instruction translation, 3-3
XEXT instruction translation, 3-3
XNAM instruction translation, 3-3

READER COMMENT SHEET

RTE-6/VM
UTILITY PROGRAMS
Reference Manual

92084-90007 **December 1981**

Update No. _____
(If Applicable)

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

FROM:

Name _____

Company _____

Address _____

_____

Phone No. _____ **Ext.** _____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 141 CUPERTINO, CA.

— POSTAGE WILL BE PAID BY —

Hewlett-Packard Company
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014
ATTN: Technical Marketing Dept.

FOLD FOLD

MANUAL PART NO. 92084-90007
Printed in U.S.A. December 1981

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014